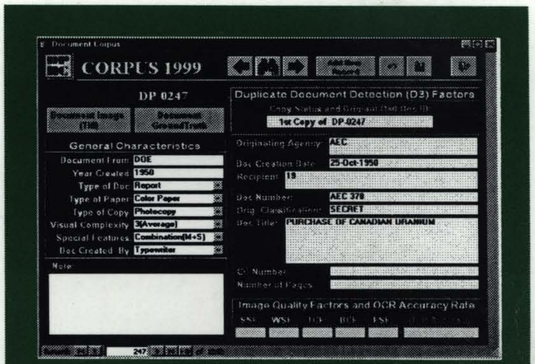
Proceedings

1999 Symposium on Document Image Understanding Technology



Document Corpus

CORPUS 1999

DP 0247

General Characteristics

Document From	DOE
Year Created	1950
Type of Doc	Report
Type of Paper	Color Paper
Type of Copy	Photocopy
Visual Complexity	3(Average)
Special Features	Combination(M+S)
Doc Created By	Typewriter

Note

Duplicate Document Detection (D3) Factors

1st Copy of DP-0247

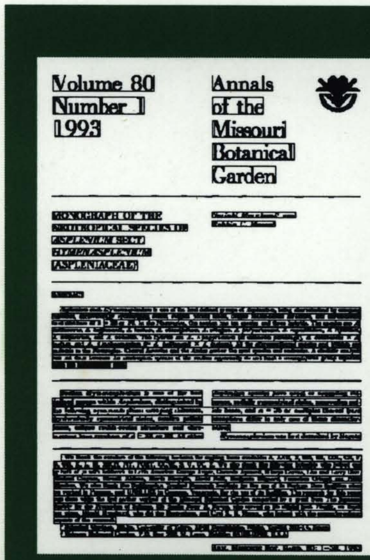
Originating Agency	AEC
Doc Creation Date	25-Oct-1950
Recipient	19
Doc Number	AEC 378
Orig. Classification	SECRET
Doc Title	PURCHASE OF CANADIAN URANIUM
C/ Number	
Number of Pages	

Image Quality Factors and OCR Accuracy Rate

SSF WSF TCF BCF FSF

247

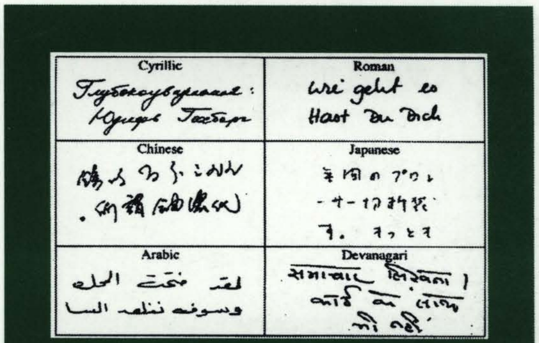
Declassification



Volume 80
Number 1
1993

Annals
of the
Missouri
Botanical
Garden

Document Image Processing



Multi-lingual Processing

April 14-16, 1999
Historic Inns of Annapolis
Annapolis, Maryland

Proceedings

SDIUT99

The 1999 Symposium on Document Image Understanding Technology

Historic Inns of Annapolis
Annapolis, Maryland
April 14-16, 1999

Sponsored by: The United States Department of Defense
Organized by: The Laboratory for Language And Media Processing
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

A limited number of additional copies of these proceedings are available for $50 from:
University of Maryland
Institute for Advanced Computer Studies
College Park, MD 20742
Phone (301) 405-6444
Fax: (301) 314-9115
Email: sdiut99@documents.cfar.umd.edu

Table of Contents

KEYNOTE SPEAKERS

SESSION 1 DOCUMENT IMAGE PROCESSING

SESSION 2 DECLASSIFICATION

SESSION 3 DUPLICATE DETECTION

SESSION 4 DOCUMENT IMAGE ASSESSMENT AND ENHANCEMENT

SESSION 5 INFORMATION EXTRACTION

SESSION 6 APPLICATIONS AND SYSTEMS

SESSION 7 DOCUMENT IMAGE PROCESSING

SESSION 8 ABSTRACTS

SESSION 9 GOVERNMENT SUPPORT

ADDITIONAL SUBMISSIONS

A Message from the Organizers

It is my distinct pleasure to welcome you to the 3rd bi-annual Symposium on Document Image Understanding Technologies. The Symposium was established in 1995 in response to the growing need to nurture ties between government, industrial and academic organizations providing support to the document understanding community. It is our hope that this Symposium will continue to contribute to the interaction between these organizations and the community at large.

This year we are proud to bring you three exciting and informative keynote presentations. The first is from William Dowling of the United States Postal Service who will provide a vision of the Postal Services Integrated Processing Facility. The second talk is by researchers Keith Knox from Xerox, and Robert Johnston and Roger Easton from the Rochester Institute of Technology, who will describe their work on the "Image Restoration of the Dead Sea Scrolls". Finally, Gary Strong from the National Science Foundation will speak on "Universals in Document and Image Understanding."

The technical and overview presentations are broken up into nine single-track sessions, on topics covering everything from declassification to multi-lingual processing to text processing in multimedia environments. As in previous years, we will also have an abstract session and demo session where researchers can interact with one another in an informal environment. This year's Proceedings contain over 40 contributions ranging in detail from technical descriptions of systems to high-level philosophical discussions of problems facing the field. Our hope is that it will serve as a benchmark for evaluating where we have come from and where we are headed in document understanding research.

Last but not least, I would like to thank Ms. Denise Best, for her countless hours of work in preparation for this Symposium. Without question, her dedication is reflected in the quality of the Proceedings, the readiness of the facilities and the overall professionalism with which this meeting has been run. We thank her for her support in making this a successful Symposium.

Thank you for your participation, and we hope you enjoy your stay in Annapolis.

David Doermann
University of Maryland

Keynote Speakers

USPS Integrated Processing Facility - Vision

William J. Dowling
United States Postal Service

Abstract

The right technology at the right price -- in time to make a difference. The Integrated Processing Facility (IPF) is a concept for a fully automated mail processing environment that integrates current and evolving technology, now under development by USPS Engineering. The goals of the IPF project are to: Upgrade and link existing automated mail processing equipment in the Processing and Distribution Centers (P&DCs); Provide tools that generate and track savings; Facilitate the introduction of value-added services; and Create a safer, more effective environment for employees.

Biographical Sketch

William J. Dowling was named Vice President of Engineering on August 21, 1992. Previously he had served as the Assistant Postmaster General for Engineering and Technical Support and before that he held the post of Regional Director of Operations Support for the Postal Service's Northeast Region.

As the Vice President for Engineering, Dowling oversees all engineering and development efforts focused on internal processes. He directs all engineering and acquisition support functions, including the design and development of new automation, material handling systems and vechicles. Dowling reports to the Chief Operating Officer and Executive Vice President, Clarence E. Lewis, Jr.

Image Restoration of the Dead Sea Scrolls

Keith T. Knox
Xerox Corporation
800 Philips Road, M/S/ 128-27E
Webster, NY 14850
kknox@crt.xerox.com

Robert H. Johnston and Roger L. Easton
Rochester Institute of Technology
Carlson Center for Imaging Science
54 Lomb Memorial Drive
Rochester, NY 14623-5604
easton@cis.rit.edu, rhjfad@ritvax.isc.rit.edu

Abstract

A team of scientists at the Xerox Digital Imaging Technology Center and the Chester F. Carlson Center for Imaging Science at the Rochester Institute of Technology has been collaborating for four years on a joint project to apply imaging technology to the study of ancient documents. The goals of the team are to enhance and clarify many kinds of degraded writings, with a particular emphasis on the Dead Sea Scrolls. Two methods have been used in this study. A custom digital camera from Eastman Kodak, sensitive from the ultraviolet to the near infrared, has been used to reveal characters in otherwise unreadable manuscripts. Additionally, special purpose image processing software has been developed at Xerox and RIT to enhance text images from photographs of degraded documents. These techniques have been applied successfully to several manuscripts dating from 1000 to 2000 years old. Through this joint effort, the team has been able to reveal writing that has not been seen, in some cases, for over two thousand years.

Biographical Sketches

Keith T. Knox received his B.S. (1970) in electrical engineering and Ph.D (1975) in Optics from the University of Rochester. He joined Xerox Corp. in 1974 and is a Principal Scientist in the Digital Imaging Technology Center, Rochester, N.Y. Knox has been involved in image processing research for over 25 years, focusing on enhancement and restoration. In his doctoral research, he developed an algorithm to restore turbulence-degraded, astronomical photographs. At Xerox, he has devised algorithms for digital halftoning, for correcting scanned input images and for digitally restoring illegible text in ancient documents. He is a Fellow of IS&T and OSA.

Robert H. Johnston is a published scholar, archaeoiogist, researcher, digital imager, consultant and academic administrator. Participated in over 60 archaeological excavations in the Middle East and the Far East. Expertise in the application of digital and scientific analysis to the study of ancient archaeological material. Special interest in the digital restoration of ancient degraded textual discoveries. Extensive research in the digital enhancement of Dead Sea Scroll fragments, Papyrus documents, (from Petra, jordan), and cuneiform tablets. Experience in digital capture, manipulation and enhancement techniques along with analytical applications using infrared, Ultraviolet, x-ray, neutron activation and the scanning electron microscope. Additional expertise in computer applications to the broad field of forensic science.

Roger Easton attended Haverford College, the University of Maryland, and the University of Arizona, where he received the Ph.D. in Optical Sciences. He is a faculty member in the Chester F. Carlson Center for Imaging Science of the Rochester Institute of Technology, where he teaches courses in imaging mathematics, optics, and digital image processing. He received the Professor Raymond C. Bellman Award from the Society for Imaging Science and Technology in 1997 for undergraduate teaching of imaging. His research interests include applications of digital holography and development of new imaging techniques to enhance readability of documents, particularly ancient writings.

Image Restoration of the Dead Sea Scrolls

Keith T. Knox
Digital Technology Center
Xerox Corporation
Webster, NY 14580

Roger L. Easton, Jr. Robert H. Johnston
Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology
Rochester, NY 14623

Abstract

Three scientists from the Xerox Digital Imaging Technology Center and the Chester F. Carlson Center for Imaging Science at the Rochester Institute of Technology have been working for four years to apply modern digital imaging technologies to the study of ancient documents. The goals of the team are twofold: to enhance the legibility of the documents, thereby aiding scholars in their studies, and to develop techniques and technologies that can be applied to modern documents in commercial applications.

1 Introduction

The Dead Sea Scrolls comprise over 800 manuscripts that include some of the earliest known religious accounts of both Biblical and non-Biblical texts. The scrolls were discovered in several caves near the Dead Sea over a period of several years beginning in 1947. A few of the scrolls were stored in jars and are in good-to-excellent condition. Most lay unprotected on the floor of the caves and were damaged, often severely, by exposure to moisture and animal waste. Many of the scrolls are now physically fragmented, decomposing, or otherwise unreadable. The importance of the scrolls to historians and religious scholars is such that ardent efforts have been made to preserve the scrolls and prevent further damage.

Until quite recently, the imaging technologies used to document and enhance the scrolls have been based on traditional photography. Only within the last few years have the methods and technologies of digital imaging have been applied to the problem of improving the legibility of characters in these manuscripts. Recently, our group applied the techniques of digital image processing to photographs of one of the Dead Sea Scrolls to enhance characters in degraded regions. We also used a wideband digital CCD camera to obtain images of some scroll fragments in the ultraviolet, visible, and infrared regions of the spectrum.

The choice of enhancement method depends upon the particular artifact or document being studied. In cases where inks have faded, images obtained under ultraviolet-light illumination may reveal characters that are either difficult to distinguish or totally unreadable when viewed by eye. If the parchment has darkened due to the effects of age or exposure, infrared imaging may reveal otherwise unreadable characters. Both techniques require that the original scrolls be available to be imaged, and the images may be further enhanced using digital image techniques.

If the actual scrolls are not available for direct examination, photographs may be processed in a digital computer to enhance subtle color differences between ink and parchment and may reveal new characters. In the work reported here on the Dead Sea Scrolls, we have used images of actual scrolls, obtained under infrared illumination, and have digitized and processed color photographs taken by others. In both cases, additional characters have been revealed.

2 Enhancing Photographs

The Temple scroll [1] contains 67 columns of text written on parchment (animal skin). A photograph of a piece of one column of the Temple Scroll is shown on the left side of Figure 1. The photographs of this scroll were taken by Bruce and Kenneth Zuckerman in collaboration with James Charlesworth, editor of the Dead Sea Scrolls Project of the Princeton Theological Seminary.

The manuscript describes an idealized Temple in Jerusalem, including the rituals that should be practiced there. Much of the scroll is in good condition while some sections of the parchment are very dark and others have many pieces missing. The original scroll is

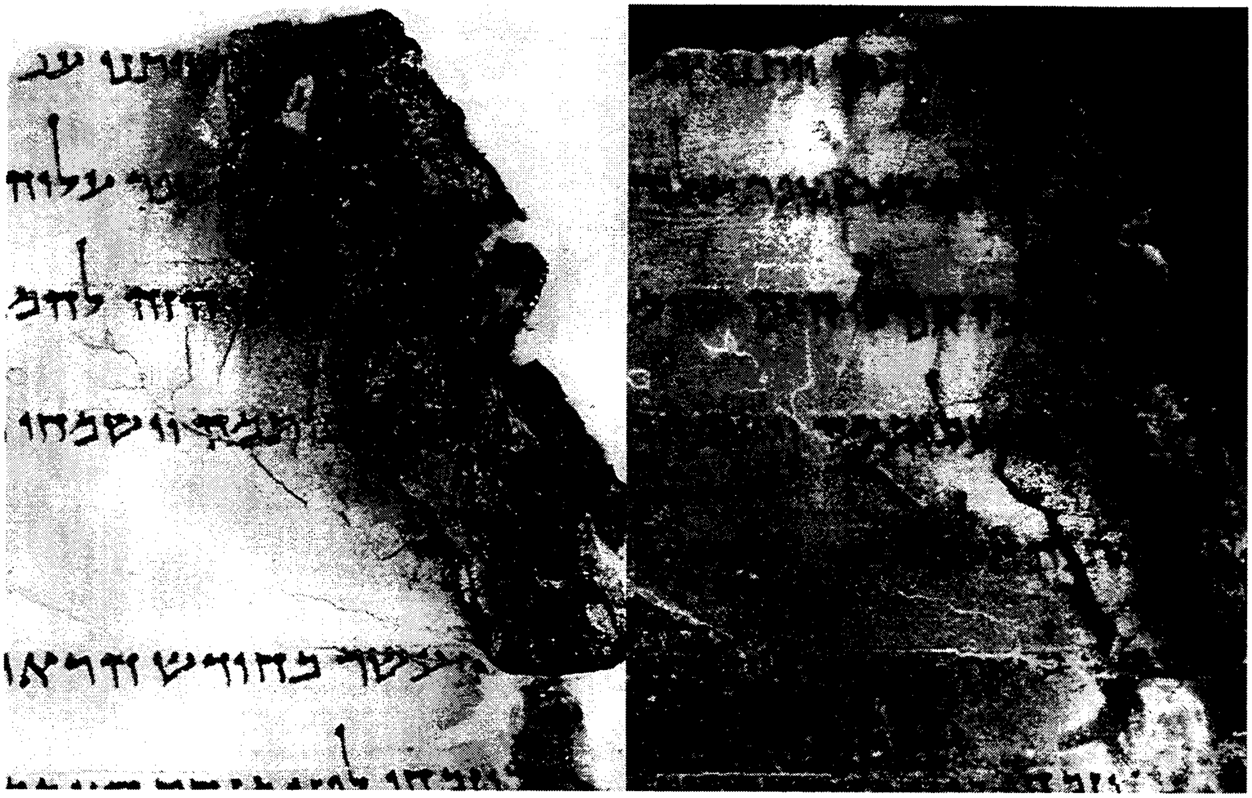


Figure 1: Temple Scroll. A photograph of the top of column 17 is shown on the left. The red-green component of the original color photograph is shown on the right. In the red-green component, five new lines of characters are visible as four overwritten lines and a brand new line in the gap. From these pictures, scholars have discovered 18 new characters that have filled in a missing piece of the description in the neighboring column.

preserved in Jerusalem and is available to scholars by request from the Israeli Antiquities Authority. Our work was done completely from photographs of the scroll taken in 1992.

The color photographs of the Temple Scroll were digitized and processed to enhance the small color differences between characters and parchment within the darkened regions. The most successful enhancement was obtained by transforming the RGB image data to the Xerox "YES" color space [2]. The "E" channel (red - green) shows remarkable new character information, as can be seen in the image on the right side of Figure 1. The first four lines are now seen to contain two sets of characters, including some previously unseen that are superimposed on the visible characters. In addition, a new line of characters has appeared in the blank space following the fourth line.

We consulted with Professor Charlesworth to establish the meaning and significance of these results. He determined that the newly discovered characters had been transferred from the neighboring column to the right from a section of parchment that no longer exists. We confirmed this conclusion by making some measurements on the digital images. Photographs of the neighboring columns were precisely aligned by matching features in the overlapping regions. After careful measurements of the distances between degraded areas, we were able to confirm Professor Charlesworth's hypothesis of the source of these characters.

We have subsequently determined that the newly discovered characters are on a piece of the scroll that tore off the neighboring column and stuck to the back side of the column shown in Figure 1. The enhancement of the small color differences made these characters visible from the photograph of the front of the scroll. Professor Charlesworth has updated the reconstructions of this missing region that were made previously by scholars [3] who studied this scroll.

3 Imaging Scroll Fragments

If actual scroll fragments are available, additional information may be obtained by imaging over a range of wavelengths. Infrared photography was used on the Dead Sea Scrolls when they were first discovered. More recently, scientists have been using digital imaging and infrared light [4] to enhance the visibility of degraded characters.

Through the kindness of Father John Peter Meno, the Archdiocesan General Secretary at St. Mark's Syrian Orthodox Cathedral in Teaneck NJ, we had the

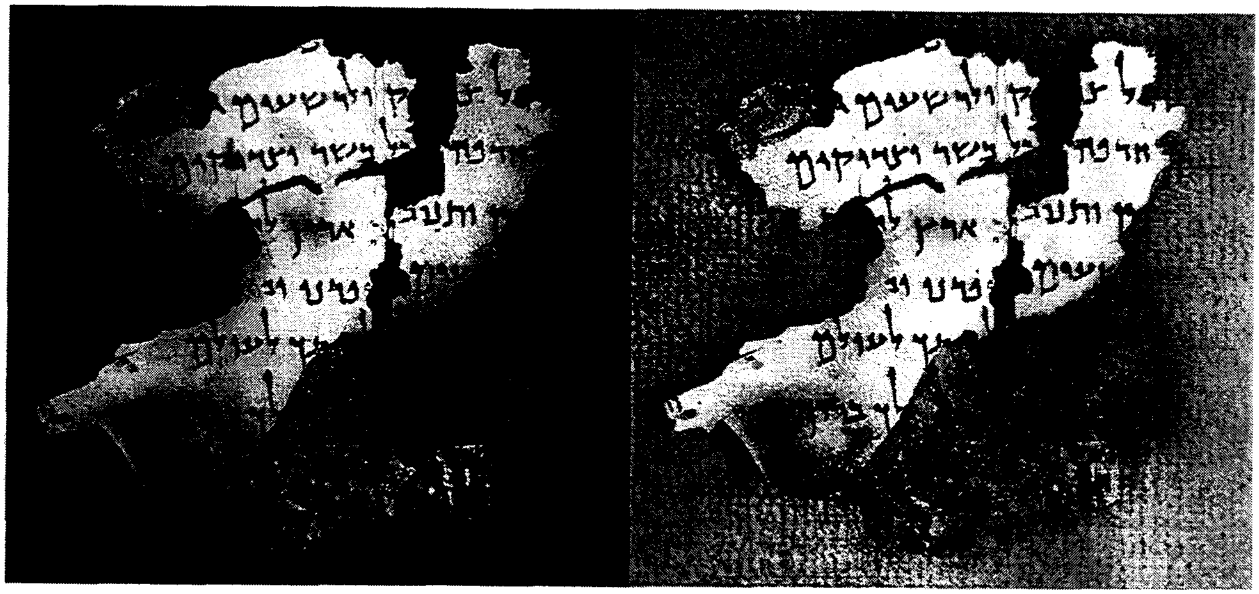


Figure 2: Liturgical Scroll. On the left is a fragment of a liturgical scroll, seen in visible light. The lower right portion of the fragment is very dark and only a few traces of characters are visible. When this fragment is viewed in infrared light, several characters appear and are easily read. The new readings from this and a similar fragment and were recently combined to reconstruct a hymn that was previously unknown to the Dead Sea Scrolls scholars.

opportunity to image several scroll fragments in June of 1997. These fragments originally were acquired in 1947 by Mar Athanasius Yeshue Samuel, who was the archimandrite of the Syrian Orthodox monastery of Saint Mark in Jerusalem. He purchased the fragments from the Bedouin shepherds who originally discovered them in caves near Qumran.

Regions of these fragments have darkened significantly due to chemical or other damage. The text in these areas is very difficult or impossible to read by eye. Images of these fragments were created in the several available bands of wavelengths with our DCS-200 Kodak digital camera and a set of glass bandpass filters under tungsten-light illumination. The contrast of the characters in the darkened sections of the parchment is significantly enhanced.

The benefit of infrared imagery is evident from Figure 2, which shows the visible and infrared images of a fragment of a liturgical scroll. Additional text is clearly visible in the infrared image. Upon subsequent analysis of these images by scholars, it was revealed that this text constitutes a previously unknown hymn. The acquisition of these images and the subsequent analysis was featured in the BBC-TV production "Traders of the Dead Sea Scrolls", which was shown on "The Learning Channel" on January 12, 1999. More details of our work on the Temple Scroll can be found in an article [5] in ***Optics & Photonics News***.

Acknowledgements

The authors wish to thank Dr. James Charlesworth and Mr. Henry Reitz of the Princeton Theological Seminary for their assistance in this work. This project was supported in part by the Center for Electronic Imaging Systems and the New York State Science & Technology Foundation.

References

[1] Y. Yadin, "The Temple Scroll: The longest and most recently discovered Dead Sea Scroll," *Biblical Archaeology Review,* Sept./Oct. (1984).

[2] Xerox Systems Institute, *Color Encoding Standard* (Xerox Corp., Palo Alto, Calif. 1989).

[3] E. Qimron, *The Temple Scroll—A Critical Edition with Extensive Reconstructions*, (Ben-Gurion Univ. of the Negev Press and the Israel Exploration Society, Beer Sheva, Jerusalem, Israel, 1996).

[4] G. H. Bearman and S.I. Spiro, "Archaeological applications of advanced imaging techniques," *Biblical Archaeologist.* **57:1** 56–66 (1996).

[5] K.T. Knox, R. H. Johnston and R. L. Easton, Jr., "Imaging the Dead Sea Scrolls," *Optics and Photonics news,* August. (1997), pp. 30-34.

Universals in Document and Image Understanding

Gary W. Strong
National Science Foundation
4201 Wilson Blvd., Suite 1115
Arlington, VA 22230
gstrong@nsf.gov

Abstract

A vision of the future of information technology research is identified and suggested for use in document image understanding. The landscape of document and image understanding is rich in the current variety of research thrusts being pursued. Neighboring research thrusts share approaches that contribute to advances in each. It is natural to wonder whether or not a merging of thrusts will be a profitable research strategy. For example, analysis of document structures can contribute to fruitful approaches to text summarization, to identification of figures and tables, and to translation of figures or tables into text. Would it be useful to develop a more universal (with respect to modality of display) description of entities, events, figures, tables, and image objects to enable additional cross-thrust processing strategies? In studies of web-based information services of the future, such an "intermedia" has been offered as one way to represent information independently of modality and language.

Biographical Sketch

Gary Strong is Deputy Director of the Information and Intelligent Systems Division of the National Science Foundation. Part of his duties are to co-ordinate efforts and represent NSF in information technology and critical infrastructure protection interagency working groups. He also manages the Human Computer Interaction Program of the Foundation, having developed several interagency programs, among them Human Language Resources and Speech, Text, Image, and Multimedia, Advanced Technology. Gary received his doctorate from the University of Michigan jointly in Computer and Communication Sciences and Anthropology. In anthropology, his thesis on Japanese culture won the 1981 American Association for the Advancement of Science Arthur F. Bentley Socio-Psychological Prize. Prior to that he received a Masters Degree from Columbia and a Bachelor's Degree from Michigan, both in Electrical Engineering. He was co-inventor on two patents at Bell Telephone Laboratories where he worked from 1967 to 1974.

Gary's work for NSF includes contributing to the Human-Centered Systems Program Component Area of the Computing, Information, and Communications R&D Subcommittee of the National Science and Technology Council. A major piece of this effort is the interagency research thrust on Universal Access, a working group on which he co-chairs. A recent study he commissioned from the National Research Council entitled *More then Screen Deep* has been widely influential in this developing field.

Document Image Processing

A System for Address Extraction from Facsimile Images

Craig R. Nohl **Jan I. Ben** **Dar-Shyang Lee**[1] **Christopher J. C. Burges**
Lucent Technologies Bell Labs
101 Crawfords Corner Road
Holmdel, NJ 07733 USA
nohl@lucent.com
ben@lucent.com
burges@lucent.com

Abstract

We describe a prototype system capable of extracting machine print addresses from fax images of English language business letters and fax cover sheets. The system automatically orients incoming page images, locates and parses machine printed addresses, and classifies each address as one of {sender, recipient, other}. We present results of preliminary performance tests, and discuss potential improvements.

1 Introduction

Document image understanding systems are useful to the extent that they can automatically extract *actionable* information from document images. In recent years, systems have been designed for progressively broader applications, including mailpiece routing, amount reading from financial documents, data entry from forms, invoice processing, and routing of business letters [1–7]. Lii and Srihari [8] consider the extraction of name and address information from fax cover sheets where field identifiers are present.

Practical systems were first developed for OCR-ready and other tightly structured documents, where the geometric and logical layout do not vary appreciably among documents of a given class. For these documents, the locations and semantics of document fields may be treated as known in advance.

More recently attention has been directed toward extraction of specific information from less structured documents, where different documents belonging to the same class may adhere only approximately to rules for geometrical and logical layout.

U.S. personal bank checks are a good example of a relatively tightly structured document class. While personal checks are not OCR-ready, and differ widely in appearance, they do conform to size constraints, and the locations of specific fields are specified with respect to a reference point on the physical document.

A *semi-structured* document class has members that exhibit statistical regularities in contents and layout, rather than uniformity. U.S. *commercial* bank checks are a good example. With the exception of the MICR (magnetic ink) line at the bottom, the layout of the document is up to the printer. Field locations do not conform to well-defined rules. However, some common practices are observed in the layout. Signatures are usually on the right-hand side of the check, toward the bottom. The date is most often in the upper half of the check, and frequently in the right half. The courtesy amount usually appears above the signature, and tends to be on the right hand side. The payee name (and sometimes the payee address) is usually in one of several general locations.

In this paper, we consider the problem of locating and determining the roles of addresses in images of two other semi-structured document classes: business letters and fax cover sheets.

Addresses provide important information about the documents containing them: they identify persons or organizations that have a variety of roles with respect to a document: author, transmitter, recipient, contact, etc. Addresses can thus be important for routing, indexing, and retrieving documents in a spectrum of real-world applications. This is true in particular of documents typically transmitted by facsimile.

Business letters commonly contain the same set of components: organization name and logo, sender address, recipient address, date, salutation, body, closing ("Sincerely"), and signature block containing a signature plus machine printed text identifying the signer. Beyond this there are no hard rules for where the components appear or how they are formatted. Sender address information may appear at the top of bottom of a page; on the left or right side; in block, multi-column, or single line format. Sender

[1] Current address: Ricoh Silicon Valley, Inc., 2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025 dsl@rsv.ricoh.com

address information may appear in more than one place. Recipient addresses may also appear in several locations, including locations typical of sender addresses. Business letters also frequently contain other address information that is associated with neither sender nor recipient.

Fax cover sheets exhibit an even wider variety of contents and layouts. Some contain no more than a handwritten name and fax number. Others look like letterhead from the sender's organization, perhaps with fields added to contain recipient address information. Still others consist of a stick-on form (with fields for sender and recipient information) applied to another document page. Many contain a mixture of handwritten and machine printed information; others are machine-generated on desktop computers, and are entirely machine printed. Some are cover sheets recycled from a fax transmission in the opposite direction ("FROM" overwritten with "TO" and vice versa). Sender and recipient address information may be formatted in distinct blocks or not. They may occupy side-by-side columns, or instead fill groups of adjacent rows in a two-column tabular format.

We describe a system designed to extract and classify machine print addresses from fax images. Our approach, which uses both lexical and geometric layout cues to identify addresses, promises good results even when image quality limits OCR accuracy.

We have focused initially on extracting U.S. addresses for senders and recipients from English language business letters and fax cover sheets. More specifically, we attempted to extract all addresses that contain at least a P.O. Box or street address plus city and state, and to classify all such addresses as one of {SENDER, RECIPIENT, OTHER}. For fax cover sheets we attempt more generally to extract any address information associated with a "To:" or "From:" identifier (or equivalent).

Address extraction is potentially useful in a number of applications. In a multi-media messaging system, where users' mailboxes may contain voice messages, email, and fax messages, address extraction supports a more friendly user interface for those who have to deal with large numbers of messages. For a desktop mailbox interface, the sender name can be included in message summaries, and messages can be listed in order priority according to sender. When accessing messages via voice interface (e.g., from a cell phone), such features are even more important to the user's efficiency. It is also possible to alert on the receipt of messages from particular senders, or to forward copies to others who are able to respond more quickly.

In the following sections, we give an overview of our system, discuss keyword spotting and the structure of address grammars, and report results of preliminary experiments.

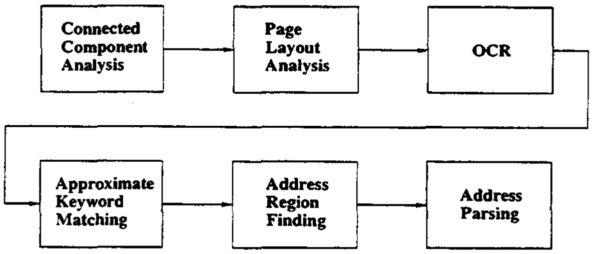


Figure 1: System Processing Overview

2 System Overview

Figure 1 shows the overall processing flow of our system.

Processing on standard or fine resolution fax images begins with connected component analysis, followed by a page layout analysis using algorithms that do not require prior knowledge of text line orientation [9]. For multi-lingual image streams, script identification and page orientation detection are performed using the methods described in [10]. OCR on lines of machine printed text is performed using the system described in [11].

OCR output is then examined in the context of the page layout to localize and parse potential addresses. Addresses are then classified according to their type.

3 Address Extraction

Addresses are extracted using a three step procedure:

- OCR output is processed to spot keywords commonly appearing in addresses, such as street suffixes ("Street", "Avenue", "Circle") and state names, and patterns typical of ZIP Codes and telephone numbers.
- Using the keywords found, one or more regions are defined such that each is very likely to contain all the words of an address.
- Words within each region are parsed according to an address grammar. Parsing assigns an address "part of speech" (e.g., CITY_WORD, or JUNK for non address words) to each word in the region, thereby locating one or more addresses. Parsing returns a score reflecting the degree of confidence in the identification.

The subsections below contain a more detailed description.

3.1 Approximate Keyword Spotting

We use keyword spotting to limit computation by limiting the portions of a document page subjected to parsing.

Since fax images are usually of relatively low resolution (200x100 dpi or 200x200 dpi) and may be imaged from poor quality hard copies, it is important that keyword spotting be tolerant of OCR errors; therefore, we used the fast approximate matching algorithm [12]. Our implementation permits matching on any number of patterns with integer insertion, deletion, and substitution costs and maximum edit cost specified separately for each pattern. It is also possible to specify that certain characters in a pattern be matched exactly. The maximum edit distances are tuned to optimize word-spotting performance for the combination of OCR engine and typical image quality.

Patterns are associated with lexical tokens, such that one or more tokens are associated with a word of OCR output through matches.

Figure 2 shows a portion of the keyword matching specification used for U.S. address extraction.

Table 1: Tag types for keyword matching

Tag	*Matches*
TELNTAG	Telephone number
ZIPTAG	ZIP Code
DIRTAG	Directional ("North")
SUFXTAG	Street suffix ("Avenue")
UNITTAG	Building part identifier ("Apt")
POBOXTAG	Post Office Box identifier
TELTAG	Telephone identifier ("Tel","Fax")
CITYTAG	City name
STATETAG	State name or abbreviation
EMAILTAG	Email identifier ("email")
EMAILADTAG	Email address
FROMTAG	Sender identifier ("From")
TOTAG	Recipient identifier ("To")
COMPANYTAG	Firm identifier ("Company")
GCAPWD	General capitalized word
GDIGSTR	General digit string

Table 1 shows a listing of tag types used for keyword matching. Note that in addition to tags intended to match the common components of addresses, there are also tags for *identifiers* – words that communicate the purpose of adjacent information in a pre-printed address ("Fax", "email"), or the purpose of fields on a fax cover sheet ("To", "From", "Company"). Finally, there are two general-purpose tags for capitalized words and strings of digits. These help in the subsequent parsing of address

```
Tag TELNTAG 1.0
 "(\d\d\d) \d\d\d[ \p]\d\d\d\d" \
  1  2  2  2  0   \
 "\d\d\d[ \p]\d\d\d[ \p]\d\d\d\d" \
  1  2  2  2 0

Tag  ZIPTAG  1.0
 "\d\d\d\d\d\p\d\d\d\d" 1  2  2  2  0   \
 "[\d01][\d01][\d01][\d01][\d01][ \p]" \
  1  2  2  0  0

Tag SUFXTAG 1.0
 "Street"   1  1  1  1  0   \
 "S[tT]"   1  1  1  0  0  \
 "Road"    1  1  1  0  0   \
 "R[dD]"   1  1  1  0  0  \
 "Avenue"   1  1  1  1  0   \
 "Ave"   1  1  1  0  0  \
 "Drive"   1  1  1  0  0  \
 "D[rR]"   1  1  1  0  0

Tag UNITTAG 1.0
 "APT"   1  1  1  0  0  \
 "SUITE"   1  1  1  1  0  \
 "ROOM"   1  1  1  0  0

Tag TELTAG  1.0
 "Fax"  1  1  1  0  0   \
 "Tel"  1  1  1  0  0   \
 "Telephone"  1  1  1  2  0  \
```

Figure 2: Part of the keyword matching specification for U.S. addresses. Fields within each pattern specification are: pattern, substitution cost, insertion cost, deletion cost, maximum edit distance, ignore case flag. The sequence \d matches any digit; the sequence \p matches punctuation.

forms.

3.2 Address Region Finding

The purpose of address region finding is to reduce overall computation by applying full-fledged address parsing only to those regions likely to contain an address.

In our system, address region finding and address extraction are currently limited to the domain of a single text block, as found by the page layout analysis algorithm.

In an early implementation, an address region was deduced from the locations and lexical tags for keyword matches within a text block, based on the structure of the grammar ultimately used for parsing the address. The grammar was analyzed to determine for each type of lexical tag the maximum number of address words that could precede and follow a word with that tag. Thus, each keyword match defined an address region ("window"). The union of such windows (for all keyword matches in the block) was the address region.

Subsequently, we have implemented grammar-based address region finding. This approach has two advantages: first, it can potentially avoid generating address regions where only isolated keyword matches are present (e.g., isolated city names in the text of a letter); second, it is capable of using geometrical layout cues to find candidate regions even when there are no keyword matches.

In the grammar-based approach, the results of page layout analysis and text interpretation plus keyword spotting are used to generate a stream of lexical tokens that are parsed by the grammar.

- *Geometrical layout* tokens include delimiters for text blocks and text lines, plus markers to indicate the alignment of the current line with the previous line. Geometrical layout tokens are shown in Table 2.
- *Word* tokens are either the specific lexical tokens generated by keyword matching, or a generic ("other") word token.

Table 2: Geometrical layout tokens

Token	*Meaning*
EB	block delimiter
EL	line delimiter
LA	line left-aligned with preceding line
GL	matches any geometrical layout token

The grammar is a stochastic regular grammar for the occurrence of one or more addresses in a text block, formatted either in common block formats (as in address blocks in business letters) or in *in-line* format (*"... next week. Please send any additional information to Craig Nohl, Lucent Technologies Bell Labs, PO Box 3030, Holmdel NJ 07733"*).

A grammar is defined as a 4-tuple $G = \{\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S}\}$, where $\mathcal{N}$ is a set of *nonterminal* symbols, $\mathcal{T}$ is a set of *terminal symbols*, $\mathcal{P}$ is a set of *production rules*, and $\mathcal{S} \in \mathcal{N}$ is a special nonterminal called the *start* symbol. Conventionally, elements of $\mathcal{N}$ are written as upper case Roman letters (A,B,C,...) and elements of $\mathcal{T}$ are written as lower case Roman letters (a,b,c,...). *Strings* of symbols from $\mathcal{V} = \mathcal{N} \cup \mathcal{T}$ are written as lower case Greek letters $(\alpha, \beta, \gamma, ...)$. Production rules are of the form $\alpha \rightarrow \beta$.

The set of all *strings* of finite length over an alphabet $\mathcal{V}$ is denoted $\mathcal{V}^*$. $\mathcal{V}^*$ includes the null string λ. When $\mathcal{P}$ contains the production rule $\alpha \rightarrow \beta$, the concatenation of strings $\gamma_1 \alpha \gamma_2$ *derives* the string $\gamma_1 \beta \gamma_2$ in the grammar G, written $\gamma_1 \alpha \gamma_2 \Rightarrow \gamma_1 \beta \gamma_2$. When $\sigma_1 \Rightarrow \sigma_2 \Rightarrow \cdots \Rightarrow \sigma_n$, the string σ_1 is said to *ultimately derive* the string σ_2. The set of all *sentences*, or finite-length strings of terminals ultimately derived from $\mathcal{S}$ in G, is called the *language* accepted by G, $L(G)$.

Each derivation of a sentence from the start symbol is a *parse* of the sentence.

A *regular* grammar can be expressed such that all production rules are of the form $A \rightarrow aB$ or $A \rightarrow a$.

A *stochastic* grammar assigns a probability to each production rule, and thus induces a probability measure on derivations, parses, and sentences. In this paper, we speak instead of the *cost* of a derivation or parse. Costs are to be thought of as the negative logarithms of probabilities, $C = -\log P$; however, the use of this language is intended to suggest that we may not have an estimate of a true probability. The cost of a parse is the sum of the costs of its individual derivations; a parse with a smaller cost (higher probability) is *better* than a parse with a larger cost.

3.3 Structure of the Address Grammar

The address grammar is used to determine the best (lowest cost) parse of the lexical token stream. This parse results in the assignment of each text word to an address component type (i.e., an address "part of speech"), or to the non- address component type `JUNK`. When the parse is complete, special nonterminals `BEGIN` and `END` can be associated with positions in the lexical token stream. The intervening words constitute a candidate address region. If the cost of the best parse is below a threshold, the address region is accepted for further analysis.

We found it convenient to specify the grammar hierarchically, as the composition of three components:

- *Full Address to Line Grammar.* The structure of a full address, expressed as allowed sequences of address line types.
- *Line to Address Component Grammar.* The structure of each line type, expressed as allowed sequences of address components.
- *Address Component to Word Grammar.* The structure of each address component, expressed as allowed sequences of Word lexical tokens.

This corresponds to partitioning of the grammar's production rules into the three sets above, and mapping its nonterminals into two sets $\mathcal{L}$ (corresponding to address line types) and $\mathcal{C}$ (corresponding to address component types). The Full Address to Line Grammar has production rules $\mathcal{S} \rightarrow \alpha$, where $\alpha \in \mathcal{L}^*$. The Line to Address Component Grammar has production rules $\alpha \rightarrow \beta$, where $\alpha \in \mathcal{L}^*$ and $\beta \in \mathcal{C}^*$. Finally, the Address Component to Word Grammar has production rules $\beta \rightarrow \gamma$, where $\beta \in \mathcal{C}^*$ and $\gamma \in \mathcal{T}^*$ is a string over the terminals.

It is convenient to summarize our grammar in an equivalent, but more compact, notation. We use an extended version of the familiar notation for regular expressions. The expression $A : \mathcal{R}$, where A is a single nonterminal and $\mathcal{R}$ is a regular expression constructed according to Table 3, is equivalent to the set of production rules $\{A \rightarrow \alpha \mid \alpha \textit{ matches } \mathcal{R}\}$. Note that terms in the regular expression $\mathcal{R}$ contain cost attributes; the cost for a production rule is the sum of the costs associated with the matching terms in $\mathcal{R}$.

Table 3: Notation for regular expression operations

Symbol	*Operation*
//	delimit regular expression
{}	enclose multi-character token
|	OR
?	zero or one occurrences of previous expression
*	zero or more occurrences of previous expression
()	grouping of operations
<>	cost for preceeding expression

A simple example of a Full Address to Line grammar is shown in Figure 3. Note that four types of tokens are accepted at this level of the grammar: geometrical layout tokens, line type tokens (summarized in Table 4), the JUNK token (intended to match non-address text), and the auxiliary tokens BEGIN and END, which delimit the address region.

```
{START}:
  / (({JUNK}<.05>)|{GL})*

  {BEGIN}
  ({EL}|{LA})?
  (({BOXLN}<3>)?
   ({STLN}<.1>({EL}{LA})? ({SDLN}<1.0>)?)
   )
  {EL}? {LA}?
  {CSZLN}
  {EL}? {LA}?
  ({TELN}<0.7> {EL}? {LA}?)*
  {END}

  (({JUNK}<.1>)|{GL})* /
```

Figure 3: A simple Full Address to Line grammar. Additive costs associated with tokens or compound expressions are shown in angle brackets (<>).

Table 4: Address line types

Symbol	*Line Type*
TOLN	Cover sheet TO line
FROMLN	Cover sheet FROM line
COMPLN	Company name
BOXLN	PO Box line
STLN	Street line
SDLN	Street secondary line
CSZLN	City/state/ZIP line
TELN	Telephone/fax line
EMAILN	Email line

A sample Line to Address Component grammar is presented in Figure 4.

The main purpose of the Address Component to Word grammar is to associate keyword matching tags and unmatched words with address components, which generally may consist of one or more words. A sample Address Component to Word grammar is shown in Figure 5. Costs in this grammar correspond to the probabilities that a particular address component will contain a given number of words, or that keyword match will actually occur where appropriate. One new token is present in this grammar: the null token Eps representing null input or output. This occurs because no input token is required to generate the BEGIN and END tokens that are accepted by the top level grammar to delimit an address region.

```
{CSZLN}:/ ({CITY}{STATE}{ZIP}) /
{BOXLN}:/ ({POBOX}{BOXNUM}) /
{STLN}:/ ({STNUM}{STNAME}{SUFX}<0.4>) |
         ({STNUM}{PRFX}{STNAME}{SUFX}<2.2>) |
         ({STNUM}{STNAME}{SUFX}{UNITNUM}<2.2>)
         ({STNUM}{STNAME}{SUFX}{PRFX}<2.2>) /

{TELN}:/ ({TEL}{TELNUM}<0.9>) |
         ({TELNUM}<.5>) /

{SDLN}:/ ({UNIT}{UNITNUM}) /
{EMAILN}:/ ({EMAIL}{EMAILAD}) /
{JUNK}:/ {JUNK} /
{GL}:/ {GL} /
{BEGIN}:/ {BEGIN} /
{END}:/ {END} /
```

Figure 4: A simple Line to Address Component grammar, expressing line types in terms of sequences of address components.

```
{STNUM}:/ ({GDIGSTR} |{w}<1> ) /
{STNAME}:/  ({GCAPWD} |
 ({GCAPWD}{GCAPWD})<1.1> | {w}<1.5> |
 ({w}{w})<2> | ({w}{w}{w})<4> ) /
{SUFX}:/ ({SUFXTAG} | {w}<1>) /
{PRFX}:/ ({PRFXTAG} | {w}<1>) /
{CITY}:/ ({CITYTAG} | {GCAPWD}<1> |
 ({GCAPWD}{GCAPWD})<1.5> | {w}<2> |
 ({w}{w})<3>) /
{STATE}:/ ({STATETAG} | {GCAPWD}<3> |
 {w}<4> | ({w}{w})<5>) /
{ZIP}:/ {ZIPTAG} | {GDIGSTR}<1> |
 {w}<2.5> /
{TEL}:/ {TELTAG} | {w}<5> /
{TELNUM}:/ ({TELNUMTAG} |
 ({GDIGSTR}{GDIGSTR})<5> |
 {w}<8> | ({w}{w})<8>) /
{GL}:/ {GL} /
{BEGIN}:/ {Eps} /
{END}:/ {Eps} /
```

Figure 5: Part of a simple Address Component to Word grammar, expressing forms of allowed address components in terms of keyword matches and unmatched words.

3.4 Grammar Training

To date, most of the cost parameters in our grammar have been hand-tuned. However, it is possible in principle to train them from ground truthed data by standard techniques, as follows.

For the training data set, ground truthing assigns a part of speech for each word, and yields a token stream consisting of these plus geometric layout tokens capturing text block structure, line breaks, and alignment of succesive text lines. In addition, the **BEGIN** and **END** tokens are inserted to delimit the actual address. The token stream for all truthed address regions are parsed by the top two layers of the grammar (Full Address to Line and Line to Address Component), and the uses of each production rule are counted. For each set of production rules sharing the same nonterminal on the left-hand side, the counts are used to estimate the probability of each possible production given the nonterminal. The negative logarithms of these counts are the costs of the productions.

This leaves the problem of determining the set of production rules. We did this by trial and error on a training database. We have not considered techniques for automated grammar inference.

3.5 Address Parsing

Once address regions have been found, they are parsed more carefully with substantially the same grammar used for address region location, but with one important change to the parsing procedure. In address region location, the costs for associating a particular interpreted text word with an address component type depended only on the set of matching keywords. No effort was made to assess the *degree* of matching. We attempt to remedy this deficiency by modeling the probability of correct matches for each address component type.

For each of the address component types {**POBOX, DIR, SUFX, UNIT, CITY, STATE, TEL**}, word interpretations are compared with a list words that may appear in an address component of that type. A proximity $d \in [0,1]$ by a fast approximate string matching algorithm, with matching cost equal to $-\log(d+\epsilon)$. Proximity equals 1 for a perfect match, and the small positive number ϵ corresponds to the probability of a match when the proximity is zero.

Address components that are normally numbers, e.g., **STNUM, UNITNUM, BOXNUM, ZIP, TELNUM**, receive a matching cost $C = -\log P_{char\ class} - \log P_{string\ length}$. Here $P_{char\ class}$ models the probability of the observed sequence of character classes {ALPHA,DIGIT,OTHER} in the OCR output given that the actual word is a string of digits; alphabetical characters and punctuation easily confused with digits are treated as special cases. $P_{string\ length}$ models

x coord of block ULH corner (x_{ULH})
y coord of block ULH corner (y_{ULH})
$N_{blocks\ to\ left} - N_{blocks\ to\ right}$ using x0
$N_{blocks\ above} - N_{blocks\ below}$ using y0
Text block area
Text block aspect ratio
$(y_{LRH} - y_{ULH})/(x_{LRH} - x_{ULH})$
Number of words in text block
Number of lines in text block
Number of left-aligned lines in text block
Fraction of block's words in address region
Number of TEL tags
Number of EMAIL tags
Number of TO tags
Number of FROM tags

Figure 6: Features used in address type classification.

the prior probability of occurrence of a digit string of specified length for an address component of the specified type.

Street names (STNAME) are currently modeled as capitalized alphabetical strings, using an approach similar to that described above for numbers.

The parsing is of a new token stream, consisting of a sequence of word interpretations and geometric layout (GL) symbols in reading order, as produced by the OCR subsystem. The Address Component to Word grammar of Figure 5 is in effect replaced by a version without the lexical tags from keyword matching: thus the "production rules" of Figure 5 contain only generic word tokens ({w}) on their right hand sides. As generic word tokens are matched with word interpretations from the input token stream, matching scores are computed dynamically using the algorithms described above.

The overall parsing algorithm computes the lowest-cost parse for the full grammar. As before, in the course of parsing an address component type (or JUNK) is assigned to each word in the address region. The identification of an address is accepted or rejected according to the cost of the parse.

4 Address Type Classification

Addresses found by the procedures described above are classified as SENDER, RECIPIENT, or OTHER addresses. Our approach to classification uses only the geometrical layout features of address regions, plus limited lexical information from the address region.

Features used for address type classification are shown in Figure 6.

Initially, we manually constructed and tuned a set of rules for address type classification, expressed in terms of the feature values for all the address regions found on a page. The following gives the flavor of the rules employed. Address regions were first analyzed to identify those that could confidently be classified as SENDER or RECIPIENT. For example, addresses that were a single line at the bottom of a page, or appeared in the upper right hand corner of the page were classified as SENDER. When not all address regions could be classified in this way, further rules were invoked that made use of features from more than one address region. For example, when exactly two good regions are found:

```
IF (both SENDER) leave as is;
ELSE IF (one is SENDER) mark other RECIPIENT;
ELSE IF (2nd region is in BOTTOM_HALF
         AND not last block AND has >1 line)
  {
  type[2nd reg]=OTHER;
  IF (1st region is RIGHT_HALF
      OR has TEL or EMAIL)
    type[1st]=SENDER;
  ELSE type[1st]=RECIPIENT;
  }
ELSE
  {
  region with greater x0-y0 is SENDER;
  other region is RECIPIENT;
  }
```

This approach yielded moderate accuracy in discrimination, but (unsurprisingly) was difficult to tune or extend.

Subsequently, we have tested machine learning approaches to discriminating address types. Our experiments so far have addressed only the classification of individual address regions, without regard to the presence or features of other address regions found on the same page. Nonetheless, overall accuracy was somewhat better than obtained using the rule-based approach.

The best-performing classifier was a set of three linear Support Vector Machines (SVMs)[13]. Each SVM exercises a linear decision surface $\mathbf{w} \cdot \mathbf{x} + b = 0$ on feature vectors $\mathbf{x}$. For each SVM, positive values of $\mathbf{w} \cdot \mathbf{x} + b$ correspond to classification as one of the classes {SENDER, RECIPIENT, OTHER}, while negative values correspond to the other two. To classify an address region, its feature vector is submitted to each of the three SVMs; the class of the SVM having the most positive score is chosen as the address type classification. Feature vector components were normalized by linearly rescaling so that for each the training set covers the range $[-1, 1]$.

5 Implementation

Our system was implemented in C and C++ on UNIX. So far, little attention has been devoted to real time performance for address parsing operations. On a Pentium II 450 MHz processor, a typical fax business letter page image is processed in less than 15 seconds, including page orientation, OCR, and processing for address extraction.

6 Experimental Results

6.1 Performance Metrics and Truthing

Ultimately, it may be of value to locate and parse entire addresses, including person names, firm names, and other organization-related information such as titles, department names, and the like. In our initial experiments, however, we concentrated on locating and parsing addresses containing a street address or post office box plus city, state, and ZIP Code. Where they were also present, we attempted to parse telephone numbers (including fax numbers) and email addresses. Addresses could comprise one or more text blocks in geometrical layout, or could occur *in-line*, e.g., as part of a textual paragraph; however, our system currently finds only addresses that lie within a single text block as determined during page layout analysis.

For fax cover sheets, we attempted in addition to extract machine printed contents of any address field contained in the same block as a `TO` or `FROM` keyword. By "block" we mean any collection of text lines that appeared to a human scorer to be part of the same logical unit. In practice, this set quite an exacting standard against which to measure system performance, since for some cover sheets the discrimination between sender and recipient address information was not immediate for human scorers.

On the other hand, we attempt to *reject* isolated occurrences of address components that don't truly constitute addresses, such as city names or telephone numbers occurring as parts of sentences.

How does one measure success in locating and parsing addresses? Clearly this depends on the application. We chose the following set of metrics:

- *Fraction of addresses found.* An address is found if any address component is returned by the system, whether or not correctly parsed.

- *Fraction of city, state, and ZIP Code components found (CSZ found).* CSZ is found if all the city, state, and postal code words are returned by the system as part of the address, whether or not correctly parsed.

- *Fraction of CSZ components correctly parsed (CSZ OK).* All CSZ words are assigned the correct part of speech during parsing.

- *Fraction of street address components found (Street found).* Street found occurs when all words of the street address component, including street number, street name, street suffix ("Avenue"), directional ("North"), and street secondary address ("Suite 200") are returned by the system as part of the address, whether or not correctly parsed.

- *Fraction of street address components correctly parsed (Street OK).* Street OK occurs when all words of the street address component are assigned the correct part of speech during parsing.

Analogous metrics are defined for the other address components (PO Box, telephone number, fax number, etc.).

6.2 Business Letters

We tested on a database of 159 standard resolution FAX images of *first pages* of U.S. business letters, from the UNLV image database [14]. These images were not used in the development of the BLADE algorithms. Results are shown in Table 5.

Table 5: Performance on UNLV Standard Resolution Fax Business Letters Database. Entries show percent correct for various tasks, separately for recipient and sender addresses. See text for explanation of row headings.

	RECIPIENT	*SENDER*
Total addresses	86	150
Address found	88%	50%
CSZ present	100%	99%
CSZ found	84%	45%
CSZ OK	83%	43%
Street present	98%	85%
Street found	69%	30%
Street OK	65%	27%

Of this set, 94% of images contain at least one address. A majority of images 79% contained exactly one sender address; 14% contained no sender address, and 8% contained more than one sender address. 54% of images contained a recipient address. 46% of images contained both a sender and a recipient address.

For recipient addresses, 88% were found; 82% had city, state, and ZIP Code correctly parsed ("CSZ OK"), and 65% had the street address correctly parsed ("Street OK") as well.

For sender addresses, 50% were found; 42% had city, state, and ZIP Code correctly parsed, and 27% had the street address correctly parsed as well.

Contributing factors to the poorer accuracy performance for sender addresses were:

- Sender addresses often appear in smaller fonts and/or italics; output from our OCR system was often of poor quality for these addresses, and they were never spotted.

- Sender addresses show a wider variation in address grammar, especially with respect to locations of line breaks.

- Sender addresses sometimes appear in a multi-column format with columns closely spaced: when such an arrangement is incorrectly segmented as a single text block during page layout analysis, the order of address components is corrupted.

Performance on fine resolution images would be expected to be somewhat better.

6.3 Fax Cover Sheets

The system for address extraction from fax cover sheets was trained and tested on a proprietary database of 177 fax cover sheet images, 138 at standard (200x100 dpi) resolution and 39 at fine (200x200 dpi) resolution. Because of the difficulty of obtaining a representative sample of fax cover sheet images, and because of the significant variability in layout among the cover sheets we collected, we decided to train our system on a subset of the test set (the first 50 images). Thus our test results may be biased toward overstating accuracy.

The training images were used in two ways:

1. To expand the portion of the grammar specific to cover sheets. When our initial grammar failed to parse some cover sheet address components, the product rules were expanded. Costs associated with the grammar were *not* trained using this image set.

2. To improve the performance of text block segmentation during page layout analysis. Initially, page layout analysis tended to split address regions (sender or recipient) across multiple text blocks, because of the relatively wide spacing between lines on fax cover sheets. We adjusted our segmentation algorithm to favor larger vertical gaps between text blocks.

We found it considerably more difficult to extract addresses from fax cover sheets than from business letters. Results are summarized in Table 6.

For the cover sheets database, 14/177 (8%) of images contained neither a sender nor a recipient address. Only 54/177 (31%) of images contained a recipient address, reflecting a large fraction of recipient address fields containing handwriting. 161/177 (91%) images contained at least one sender address region; 42/177 (24%) contained more than one.

Table 6: Performance on 177 fax cover sheets. Entries show percent correct for various tasks, separately for recipient and sender addresses. See text for explanation of row headings. CSZ and street component results are almost never present in recipient addresses.

	RECIPIENT	*SENDER*
Total addresses	54	205
Address found	24%	38%
CSZ present	6%	59%
CSZ found	6%	30%
CSZ OK	4%	22%
Street present	4%	58%
Street found	4%	29%
Street OK	2%	19%
Fax present	93%	72%
Fax found	19%	32%
Tel present	26%	71%
Tel found	9%	28%
Firm present	30%	NA
Firm found	9%	NA

52/177 (29%) of images contained both sender and recipient addresses.

Even though we started with a moderately large number of cover sheets, fewer than one third contained detectable [machine print] recipient address information. Recipient addresses were normally identified by the presence of a TO identifier; virtually none contained street or CSZ address components. The most frequently found recipient address component was a fax number; however, we still found this only about 20% of the time. Firm names were parsed with the aid of keyword matching on firm identifiers (e.g., "Company"); firm names were found about a third of the time when present.

The main reason for this poor performance was the difficulty in associating the recipient identifier (TO) with a machine printed address component that was normally in a different field. OCR errors were also a contributing factor.

Roughly 60% of sender addresses appeared in the same forms familiar from business letters – typically containing street or PO Box and CSZ components. Many occurred in "letterhead" versions of preprinted fax cover sheets. Addresses containing a CSZ component were found at rates similar to those in business letters, about 50%.

Sender address components also appeared as the contents of one or more cover sheet fields, typically consisting of a person and/or firm name, telephone number, and fax number. These were found at a lower rate, roughly 20%, due to the same difficulties as for recipient addresses.

6.4 Address Type Classification

Our address type classification subsystem was trained and tested only for fax cover sheet images.

Due to the small sample available for training, we evaluated performance by the "leave out one" procedure, where a classifier is trained on all but one sample and tested on the remaining sample (left out of training). Results are averaged over all choices of the partitioning between training and test samples. In this way, the classifier is never tested on a sample on which it has been trained, yet all samples are used for both training and testing. This approach was feasible only because classifiers could be trained rapidly on the relatively small training set.

The training/test set consisted of one feature vector (see Figure 6) for each address region yielding a valid parse, without regard to the parse score. Out of 177 fax cover sheet images, 70 yielded one or more such address regions, for a total of 97 address regions. The classifier achieved an accuracy of 79%. This compares with an accuracy of 77% for a nearest neighbor classifier. Table 7 shows the confusion matrix for the SVM classifier configuration.

Table 7: Confusion matrix for classification of address region type as {SENDER, RECIPIENT, OTHER} by three linear SVMs.

			Result	
TRUTH	Total	Sender	Recipient	Other
Sender	72	71	0	1
Recipient	16	10	6	0
Other	9	9	0	0

It is important to keep in mind that these results are obtained using only information about the address region being classified. Nonetheless, the overall accuracy was higher than we obtained via handcrafted rules that attempted to make use of information about *all* address regions found on a page. We would expect to see significantly improved accuracy from a hybrid approach that uses machine learning techniques in considering features from multiple address regions. Such an approach may require additional training data.

7 Conclusions

We have described a system for extracting machine printed address components from English language business letters and fax cover sheets, and have reported preliminary performance measurements.

For business letters, address extraction performance is primarily limited by OCR performance. Improvement can also be expected from expanding keyword lists, the address grammar, and the word lists used for dynamic matching of specific word types.

Performance on fax cover sheets was limited by the same factors, and others as well. Field-based address regions were often fragmented into multiple text blocks during the page layout analysis phase, often preventing recognition of an address. Techniques are needed to identify and group fields according to whether they contain sender or recipient address information.

In our preliminary tests, address type classification accuracy was mediocre. However, much potentially relevant information is not yet used. The *banner* entered at the top of each image by the sending fax machine (and often available in character form at the receiving fax machine) contains information that frequently correlates with the contents of sender address fields, and could improve type classification accuracy, though confusions are still possible. Similarly, when a cover sheet or business letter contains a signature block the person name, and company, if present, should match the sender information. A name in the salutation line of a letter usually matches the recipient name. Such features in combination with a machine learning approach that makes use of all address information found on a page should significantly improve classification performance.

References

[1] J. Schürmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler, and M. Oberländer, Document analysis – From pixels to contents. *Proceedings of the IEEE* **80** (1992) 1101–1119.

[2] T. Bayer, U. Bohnacker, and I. Renz, Information extraction from paper documents, in *Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P. S. P. Wang, editors (World Scientific, 1997) 653–677.

[3] S. Baumann, M. Ben Hadj Ali, A. Dengel, T. Jager, M. Malburg, A. Weigel, and C. Wenzel, Message extraction from printed documents: a complete solution, in *Proceedings of the International Conference on Document Analysis and Recognition* (IEEE Computer Society, 1997) 1055–1059.

[4] A. Dengel, R. Bleisinger, F. Fein, R. Hoch, and F. Hönes, Officemaid - a system for office mail analysis, interpretation, and delivery, in *Proceedings of the First International Workshop of Document Analysis Systems (DAS'94)*, Kaiserslautern, Germany, 1994, 253–275.

[5] C. Wenzel, Supporting information extraction from printed documents by lexico-semantic pat-

tern matching, in *Proceedings of the International Conference on Document Analysis and Recognition* (IEEE Computer Society, 1997) 732–735.

[6] H. U. Mogg-Schneider and C. Aufmuth, Information extraction from tax assessment forms, in *Document Analysis Systems II*, J. J. Hull and S. L. Taylor, editors (World Scientific, 1998) 209–222.

[7] M. Köppen, D. Waldöstl, and B. Nickolay, Information extraction from tax assessment forms, in *Document Analysis Systems II*, J. J. Hull and S. L. Taylor, editors (World Scientific, 1998) 223–241.

[8] J. Lii and S. N. Srihari, Location of name and address on fax cover pages, in *Proceedings of the International Conference on Document Analysis and Recognition* (IEEE Computer Society, 1995) 756–759.

[9] H.S. Baird, Global-to-local layout analysis, in *IAPR workshop on syntatic and structural pattern recognition* (1988) 1–16.

[10] D.-S. Lee, C. R. Nohl, and H. S. Baird, Language identification in complex, unoriented, and degraded document images, in *Proceedings of the IAPR Workshop on Document Analysis Systems*, J. J. Hull and S. L. Taylor, editors (World Scientific, 1998) 17–39.

[11] H.S. Baird, Anatomy of a versatile page reader, *Proceedings of the IEEE* **80** (1992) 1059–1065.

[12] S. Wu and U. Manber, Fast text searching allowing errors, *Communications of the ACM* **35** (October 1992) 83–91.

[13] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2**(2) (1998) 121–167.

[14] S. V. Rice, F. R. Jenkins, and T. A. Nartker, The fifth annual test of ocr accuracy, Technical Report TR-96-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1996.

A Unified Approach for Document Structure Analysis and its Application to Text-line Extraction

Jisheng Liang[†] **Ihsin T. Phillips**[‡] **Robert Haralick**[†]

[†] Department of Electrical Engineering
University of Washington Seattle, WA 98195

[‡] Department of Computer Science/Software Engineering
Seattle University, Seattle, WA 98122

Abstract

In this paper, we formulate the document segmentation as a partitioning problem. The goal of the problem is to find an optimal solution to partition the set of glyphs of a given document to a hierarchical tree structure where entities within the hierarchy have their physical properties and semantic labels. A unified approach is proposed for the partitioning problem. The Bayesian framework is used to assign and update the probabilities. An iterative, relaxation like method is used to find the partitioning solution that maximizes the joint probability.

We have implemented a text-line extraction algorithm using this framework. The algorithm was evaluated on the UW-III database of some 1600 scanned document image pages. For a total of 105,020 text lines, the text-line extraction algorithm identifies and segments 104,773 correctly, an accuracy of 99.76%. The detail of the algorithm is presented in this paper.

1 Introduction

Given a document image, the end result of a document segmentation algorithm, in general, produces a hierarchical structure that captures the physical structure and the logical meaning of an input document. The top of the hierarchical structure presents the entire page, and the bottom of the structure includes all glyphs on the document. Entities in the hierarchy are labeled and are associated with a set of attributes describing the nature of the entities. For example, the character set on a textual document would reside at the bottom of the hierarchy; each character would be labeled as a "glyph", and the attributes for the glyph may be the ASCII value, the font style, and the position of the character. The next level up may be words, then, text-lines, text-zones, text-blocks, and so on to the entire page.

Most known page segmentation algorithms [1]-[15] construct the document hierarchy from level to level, up and down within the hierarchy, until the hierarchical structures are built and the segmentation criteria are satisfied. Within this model, the page segmentation problem may be considered as a series of level-construction operations. That is, given a set of entities at a certain level of hierarchy, say source_level, the goal of the level-construction operation is to construct a set of entities for another level, say target_level.

In this paper, we propose a methodology for formulating and solving the document page segmentation problem. Our methodology uses the Bayesian framework. The methodology can be applied, uniformly, to any level-construction operation within the document hierarchy. To illustrate the usage of this methodology, a text-line extraction algorithm has been implemented and presented in this paper.

The remaining of this paper is organized as follows. In Section 2, we present the proposed methodology for the document segmentation problem and a general purpose algorithm derived from the methodology. In Section 3, we give, in detail, the text-line extraction algorithm which we implemented using the proposed methodology. In Section 4, we discuss how those probabilities used in the algorithm were computed. The paper summary is given in Section 6.

2 The Methodology

2.1 Document Structure Analysis Formulation

Let $\mathcal{A}$ be the set of entities at the source_level. Let Π be a partition of $\mathcal{A}$ and each element of the partition is an entity on target_level. Let L be a set of labels that can be assigned to elements of the partition. Function $f : \Pi \rightarrow L$ associates each element of Π with a label. $V : \wp(\mathcal{A}) \rightarrow \Lambda$ specifies measurement made on subset of $\mathcal{A}$, where Λ is the measurement space.

The problem can be formulated as follows: given initial set $\mathcal{A}$, find a partition Π of $\mathcal{A}$, and a labeling function $f : \Pi \to L$, that maximizes the probability

$$\begin{aligned} & P(V(\tau) : \tau \in \Pi, f, \Pi|\mathcal{A}) \\ = \;& P(V(\tau) : \tau \in \Pi|\mathcal{A}, \Pi, f)P(\Pi, f|\mathcal{A}) \\ = \;& P(V(\tau) : \tau \in \Pi|\mathcal{A}, \Pi, f) \\ & \times P(f|\Pi, \mathcal{A})P(\Pi|\mathcal{A}) \end{aligned} \tag{1}$$

By making the assumption of conditional independence, that when the label $f(\tau)$ is known then no knowledge of other labels will alter the probability of $V(\tau)$, we can decompose the probability 1 into

$$\begin{aligned} & P(V(\tau) : \tau \in \Pi, f, \Pi|\mathcal{A}) \\ = \;& \prod_{\tau \in \Pi} P(V(\tau)|f(\tau))P(f|\Pi, \mathcal{A})P(\Pi|\mathcal{A}) \end{aligned} \tag{2}$$

The possible labels in set L is dependent on the target_level and on the specific application. For example, $l \in L$ could be text content, functional content type, style attribute, and so for.

The above proposed formulation can be uniformly apply to the construction of the document hierarchy at any level, e.g., text-word, text-line, and text-block extractions, just to name a few. For example, as for text-line extraction, given a set of glyphs, the goal of the text-line extraction is to partition glyphs into a set of text-lines, each text-line having homogeneous properties, and the text-lines' properties within the same region being similar. The text-lines' properties include, deviation of glyphs from the baseline, direction of the baseline, text-line's height, and text-lines' width, and so for.

As for the text-block segmentation, for example, given a set of text lines, text-block segmentation groups text lines into a set of text-blocks, each block having homogeneous formatting attributes, e.g. homogeneous leading, justification, and the attributes between neighboring blocks being similar.

2.2 A General Purpose Algorithm for Document Entity Extraction

Given an initial set $\mathcal{A}$, we first construct the read order of the elements of $\mathcal{A}$. Let $A = (A_1, A_2, \cdots, A_M)$ be a linearly ordered set (chain in $\mathcal{A}$) of input entities. Let $\mathcal{G} = \{Y, N\}$ be the set of grouping labels. Let A^P denote a set of element pairs, such that $A^p \subset A \times A$ and $A^p = \{(A_i, A_j)|A_i, A_j \in A \text{ and } j = i+1\}$. Function $g : A^p \to \mathcal{G}$, associates each pair of adjacent elements of A with a grouping label, where $g(i) = g(A_i, A_{i+1})$. Then, the partition probability $P(\Pi|A)$ can be computed as follows,

$$\begin{aligned} & P(\Pi|A) = P(g|A) \\ = \;& P(g(1), \cdots, g(N-1)|A_1, \cdots, A_N) \\ = \;& P(g(1)|A_1, A_2) \times \cdots P(g(N-1)|A_{N-1}, A_N) \\ = \;& \prod_{i=1}^{N-1} P(g(i)|A_i, A_{i+1}) \end{aligned} \tag{3}$$

Therefore, the joint probability is further decomposed as

$$\begin{aligned} & P(V(\tau) : \tau \in \Pi, f, \Pi|A) \\ = \;& \prod_{\tau \in \Pi} P(V(\tau)|f(\tau)) \times P(f|\Pi, A) \\ & \times \prod_{i=1}^{N-1} P(g(i)|A_i, A_{i+1}) \end{aligned} \tag{4}$$

An iterative search method is developed to find the consistent partition and labeling that maximizes the joint probability of equation 4.

```
1. Determine initial partition
   Let $t = 0$, $\Pi^t = \{\{A_m\}\}_{m=1}^M$.
   (a) Compute $P_i^0(Y) = P(g(i) = Y|A_i, A_{i+1})$
       and $P_i^0(N) = P(g(i) = N|A_i, A_{i+1})$ where
       $1 \le i \le M-1$.
   (b) Let $R \subseteq A \times A$ and
       $R = \{(A_i, A_{i+1})|P_i^0(Y) > P_i^0(N)\}$. Update partition
       $\Pi^{t+1} = \{\tau|\tau = \{A_i, A_{i+1}, \cdots, A_j\}, \text{where } (A_k, A_{k+1}) \in R, k = i, \cdots, j-1\}$
2. Search for optimal partition adjustment
   Repeat
   • For $i = 1$ to $M - 1$ Do
     – If $A_i \in U$, $A_{i+1} \in W$, $U \neq W$ Then,
       (a) Let
           $T = U \bigcup W.$
           and
           $\hat{\Pi} = T \bigcup (\Pi^t - U - W)$
       (b) Find labeling $f$ by maximizing
           $P_{label} = \prod_{\tau \in \hat{\Pi}} P(V(\tau)|f(\tau))P(f|A, \hat{\Pi})$
       (c) $P_i^{\hat{t}}(Y) \propto P_i^0(Y) \times P_{label}$, and
           $P_i^{\hat{t}}(N) = P_i^{t-1}(N)$.
     – If $A_i \in W$ and $A_{i+1} \in W$, where $W = \{A_k, \cdots, A_i, A_{i+1}, \cdots, A_j\}$, Then
       (a) $S = \{A_k, \cdots, A_i\}$ and $T = \{A_{i+1}, \cdots, A_j\}$
           $\hat{\Pi} = (\Pi^t - W) \bigcup S \bigcup T$
       (b) Find labeling $f$ by maximizing
           $P_{label} = \prod_{\tau \in \hat{\Pi}} P(V(\tau)|f(\tau))P(f|A, \hat{\Pi})$
       (c) $P_i^{\hat{t}}(N) \propto P_i^0(N) \times P_{label}$, and
           $P_i^{\hat{t}}(Y) = P_i^{t-1}(Y)$
```

```
    End
  • Select $k$ such that,
        $k = \arg\max_i(\max\{\hat{P_i^t}(Y), \hat{P_i^t}(N)\})$
  • If $P_k^t\hat{(Y)} > P_k^t\hat{(N)}$, Then
      – $T = U \bigcup W$ where $A_k \in U, A_{k+1} \in W$
      – $\Pi^{t+1} = (\Pi^t - U - W) \bigcup T$
    Else, $W = \{A_i, \cdots, A_k, A_{k+1}, \cdots, A_j\}$,
      – Let $S = \{A_i, \cdots, A_k\}$ and $T = \{A_{k+1}, \cdots, A_j\}$
      – $\Pi^{t+1} = (\Pi^t - W) \bigcup S \bigcup T$
  • If $P(V, f, \Pi^{t+1}|A) \leq P(V, f, \Pi^t|A)$, end and return $\Pi^t$.
    Else, let $t = t + 1$ and continue.
```

Our method consists of two major components – off-line statistical training and on-line segmentation. Section 3 presents our on-line algorithm of text-line and zone segmentation. Our statistical training method is given in section 4.

3 Text-line Extraction Algorithm

Figure 1 gives an overview of the text-line segmentation algorithm. Without loss of generality, we assume that the reading direction of the text-lines on the input page is left-to-right. The text-line segmentation algorithm starts with the set of the connected-components bounding boxes of a given binary image of a textual document.

Algorithm:

1. Extract & Filter Glyphs:

   We apply the standard connected-component algorithm to obtain the glyph set, $C = \{c_1, c_2, \cdots, c_M\}$. Those components that are smaller than the $threshold_{small}$ or larger than the $threshold_{large}$ are removed from C.

2. Locate Glyph Pairs:

   For each $c_i \in C$, we search for its "nearest right mate", c_j, among those "visible" right neighbors of c_i. When a right mate is found, a link is established between the pair. The definitions for the nearest right mate and the visible right neighbors are given in section 3.1. Note that, a glyph at the right-most edge of a document would not have a right mate. At the end of this step a set of text-line segments are established, $T_{segment} = \{t_1, t_2, \cdots, t_{K_1}\}$. For each linked pair, c_i and c_j, we compute the grouping probability, $P(sameline(i,j)|c_i, c_j)$. This is the estimated probability that two components with their sizes and spatial relationships lie on the same text-line.

3. Group Text-lines:

   For each $t_k \in T_{segment}$ formed in step 2, we check each link $(c_i, c_j) \in t_k$ and estimate the linking probability $P(link(i,j))$ between c_i and c_j. If $P(link(i,j) = N) > P(link(i,j) = Y)$, we disconnect (c_i, c_j) link. That is, t_k becomes two subsegments.

   During the initial partition, $P(link(i,j)) = P(sameline(i,j))$, and this step yields our initial text-line set, $T_{initial} = \{t_1, t_2, \cdots, t_{K_2}\}$.

4. Detect Base-line & X-height:

   For each $t_k \in T_{initial}$, we apply a robust line fitting algorithm to the right-bottom corners of all glyphs in t_k to obtain the base-line and the direction of t_k. The computation of base-line and X-height are given in Section 3.2.

5. Detect Page Skew:

   The median of all the computed base-lines' direction for the entire set $T_{initial}$ is taken as the page skew angle, $angle_{skew}$. If the $angle_{skew} > threshold_{skew}$, we rotate the image by $-angle_{skew}$ using the technique given in [17], and the process repeats from step 1. Otherwise, proceed to the next step.

6. Compute Text-line Probability:

   For each text-line $t_k \in T_{initial}$, We compute its probability of having the homogeneous text-line properties,

   $$P(V(t_k)|textline(t_k)),$$

   where $V(t_k)$ is the measurement made on the text-line t_k.

   The observation that we make is the component distance deviation σ_{t_k} of t_k from its base-line. If $P(\sigma_{t_i}|textline(t_i)) > threshold_\sigma$ we accept t_i. Otherwise, we pick the weakest link (c_i, c_j) within t_k as the potential breaking place where we may sub-divide t_j into t_{j1} and t_{j2}.

7. Adjust Pairs linking Probability:

   To determine whether to sub-divide t_k, we compute t_{k1}'s and t_{k2}'s base-lines, and their component deviations, $\sigma_{t_{k1}}$ and $\sigma_{t_{k2}}$.

   We update the linking probability between c_i and c_j by combining their grouping probability with the text-line probability,

   $$\begin{aligned} &P(link(i,j) = Y) \\ &\quad \propto \;\; P(sameline(i,j) = Y|c_i, c_j) \\ &\qquad\quad \times P(\sigma_{t_k}|textline(t_k)), \end{aligned}$$

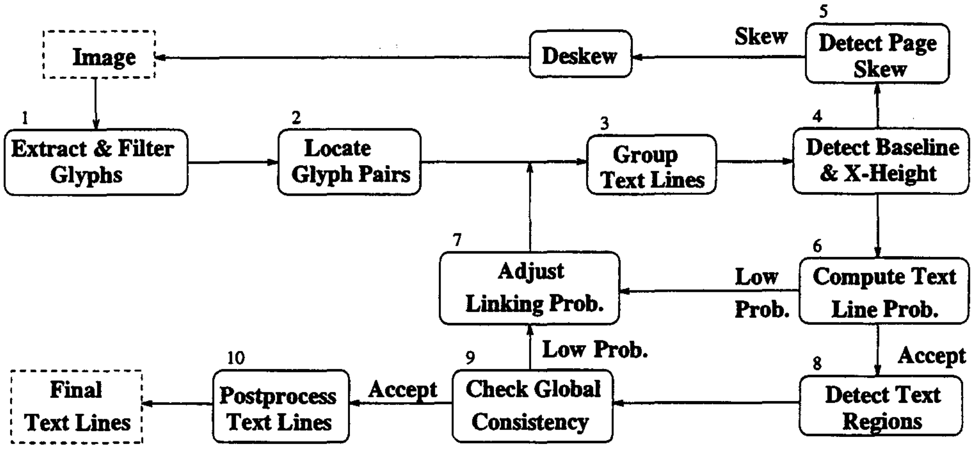


Figure 1: Illustrates the processing steps of the text-line segmentation algorithm.

and

$$\begin{aligned} & P(link(i,j) = N) \\ & \quad \propto \; P(sameline(i,j) = N|c_i, c_j) \\ & \qquad \times P(\sigma_{t_{k1}}|textline(t_{k1})) \\ & \qquad \times P(\sigma_{t_{k2}}|textline(t_{k2}), \end{aligned}$$

where

$$P(link(i,j) = Y) + P(link(i,j) = N) = 1.$$

If $P(link(i,j) = N) > P(link(i,j) = Y)$, the process repeats from step 3. Otherwise, proceed to the next step.

8. Detect Text Regions and Zones:

To detect text-regions with respect to all text-lines in $T_{interim}$, we do as follows. For each text-line $t_k \in T_{interim}$, we compute it's bounding box and the three bounding box edge positions: the left, the center, and the right.

Then, a horizontal projection profile is computed on all the text-line bounding boxes. Each text-line box constitutes one count on the profile. A horizontal cut is made where the gap within the profile satisfies our cutting criteria. The computation of the projection profile and the cutting criteria are given in detail in section 3.3.

The result of the last step is a sequence of horizontal text-regions, $R = \{R_1, R_2, \cdots, R_r\}$. In this step, each of the region, R_i, is to be further decomposed into a sequence of text-zones by cutting R_i vertically. The top and the bottom edges of R_i become the top and the bottom edges of the text-zones. Our text-zone detection finds the left and the right edges of text-zones within R_i.

Let $R_i = \{t_1, t_2, \cdots, t_p\}$ be a horizontal text-region, $R_i \in R$. To detect a text-zone within R_i, we compute the vertical projection profile on the left, the center, and the right positions of all text-lines $t_k \in R_i$.

Next, we locate the bin with the max count on the profile. If the max count comes from, say, the left position of the majority of the text-lines that contribute to the max count, we say, we have detected a left-edge of a text-zone, Z_n. Let $\{t_1, t_2, \cdots, t_m\}$ be the sequence of text-lines whose left positions fall within the bin which has the max count. The left-edge of Z_n is estimated as the median of the left edge position of all text-lines within Z_n. The right edge of Z_n is computed in a similar fashion. The top and the bottom edges of Z_n are the top and the bottom edges of R_i. Then, all the text-lines within Z_n are removed from further consideration, and this step is repeated until each text-line in R is assigned to one of the detected text-zones. A complete description of this step is given in section 3.3.

9. Check Global Consistency (Splits & Merges):

Let $Z = \{Z_1, Z_2, \cdots, Z_n\}$ be the set of the detected text-zones from the last step. Let $Z_i \in Z$ and $Z_i = \{t_1, t_2, \cdots, t_k\}$. We examine the probability, $P_{context}(\omega(t_j), \omega(Z_i))$, that t_j's attributes $\omega(t_j)$ being consistent with its neighboring text-lines within Z_i. (The computation of $P_{context}$ is given in section 3.4.)

If $P_{context}(t_j) < threshold_{context}$, we update the linking probability for each pair within t_j, and the process repeats from step 3. Step 8 and 9 are repeated until $P_{context}(t_j) > threshold_{context}$ is satisfied for all t_j. The complete description of the global consistent check, the split and the merge procedures are given in detail in section 3.4.

10. Postprocess Text Lines: Finally, all components which were initially put into the reserved set and those text-lines which were not included during the text-zone formation, or as the results of splitting, are now be individually examined to determine whether it could be included in any of the segmented text-lines.

Figure 2 and 3 illustrate the text line detection process. Figure 2(a) shows a set of connected component bounding boxes. The extracted initial text line segments by merging pairs of connected components are illustrated in Figure 2(b). We notice some text lines are split while some are merged across different columns. Figure 3(c) plots the extracted text regions by grouping the edges of text segments. Finally, the corrected text lines given the observations on text regions are shown in Figure 3(d).

A few cases that the algorithm failed are shown in Figure 4. A vertical merging error was shown in Figure 4(a). Figure 4(b) and (c) illustrate horizontal and vertical splitting errors due to the large spacing. A spurious error caused by warping is shown in Figure 4(d).

3.1 Mate Pairs and Grouping Probability

Let $C = \{c_1, c_2, \cdots, c_M\}$ be the set of glyphs, the connected-component set after the too small components are removed. Each glyph $c_i \in C$ is represented by a bounding box (x, y, w, h), where x, y is the coordinate of top-left corner, and w and h are the width and height of the bounding box respectively. The spatial relations between two adjacent boxes are shown in Figure 5.

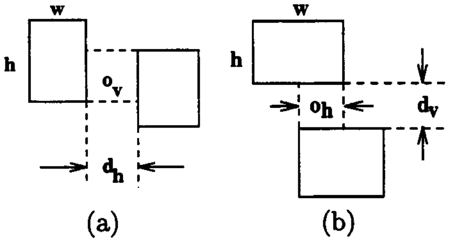


Figure 5: Illustrates the spatial relations between two bounding boxes that are (a) horizontally adjacent (b) vertically adjacent.

For a pair of bounding boxes a and b, the horizontal distance $d_h(a,b)$ and vertical distance $d_v(a,b)$ between them are defined as

$$d_h(a,b) = \begin{cases} x_b - x_a - w_a & \text{if } x_b > x_a + w_a \\ x_a - x_b - w_b & \text{if } x_a > x_b + w_b \\ 0 & \text{otherwise} \end{cases}$$

$$d_v(a,b) = \begin{cases} y_b - y_a - h_a & \text{if } y_b > y_a + h_a \\ y_a - y_b - h_b & \text{if } y_a > y_b + h_b \\ 0 & \text{otherwise} \end{cases}$$

The horizontal overlap $o_h(a,b)$ and vertical overlap $o_v(a,b)$ between a and b are defined as

$$o_h(a,b) = \begin{cases} x_a + w_a - x_b & \text{if } x_b > x_a,\ x_b < x_a + w_a \\ x_b + w_b - x_a & \text{if } x_a > x_b,\ x_a < x_b + w_b \\ 0 & \text{otherwise} \end{cases}$$

$$o_v(a,b) = \begin{cases} y_a + h_a - y_b & \text{if } y_b > y_a,\ y_b < y_a + h_a \\ y_b + h_b - y_a & \text{if } y_a > y_b,\ y_a < y_b + h_b \\ 0 & \text{otherwise} \end{cases}$$

Let $c_a = (x_a, y_a, w_a, h_a)$ and $c_b = (x_b, y_b, w_b, h_b)$ be two glyphs. We define c_b as a "visible" right neighbor of c_a if $c_b \neq c_a, x_b > x_a,$ and $o_v(a,b) > 0$. Let C_a be the set of right neighbors of c_a. The "nearest" right neighbor of c_a is defined as

$$\arg \min_{c_i \in C_a} (d_h(a,i) | c_i \neq c_a, x_i > x_a, o_v(a,i) > 0).$$

For each linked pair, c_a and c_b, we associate with their link with the probability, $P(sameline(a,b)|c_a, c_b)$, that indicate how probable they belong to the same text-line. Given the observations of their heights and widths, and the distance and the overlaps between the pair: h_a, w_a, h_b, w_b, $d(a,b)$, $o(a,b)$, we compute the probability that c_a and c_b belong to the same text-line as:

$$P(sameline(a,b)|h_a, w_a, h_b, w_b, d(a,b), o(a,b)).$$

3.2 Base-line, X-height, and Skew Angle

The baseline coordinate of a text-line is estimated using a robust estimator. The robust estimation means it is insensitive to small departures from the idealized assumptions for which the estimator is optimized.

We want to fit a straight line $y(x; a, b) = a + bx$ through a set of data points, which are the bottom-right corner of glyph boxes, since ascenders are used more often in English texts than descenders. The merit function to be minimized is

$$\sum_{i=1}^{N} |y_i - a - bx_i|.$$

The median c_M of a set of numbers c_i is also the value which minimizes the sum of the absolute deviations $\sum_i |c_i - c_M|$. It follows that, for fixed b, the value of a that minimizes the merit function is $a = median\{y_i - bx_i\}$, where $b = \sum_{i=1}^{N} sgn(y_i - a - bx_i)$. This equation can be solved by the bracketing and bisection method [16].

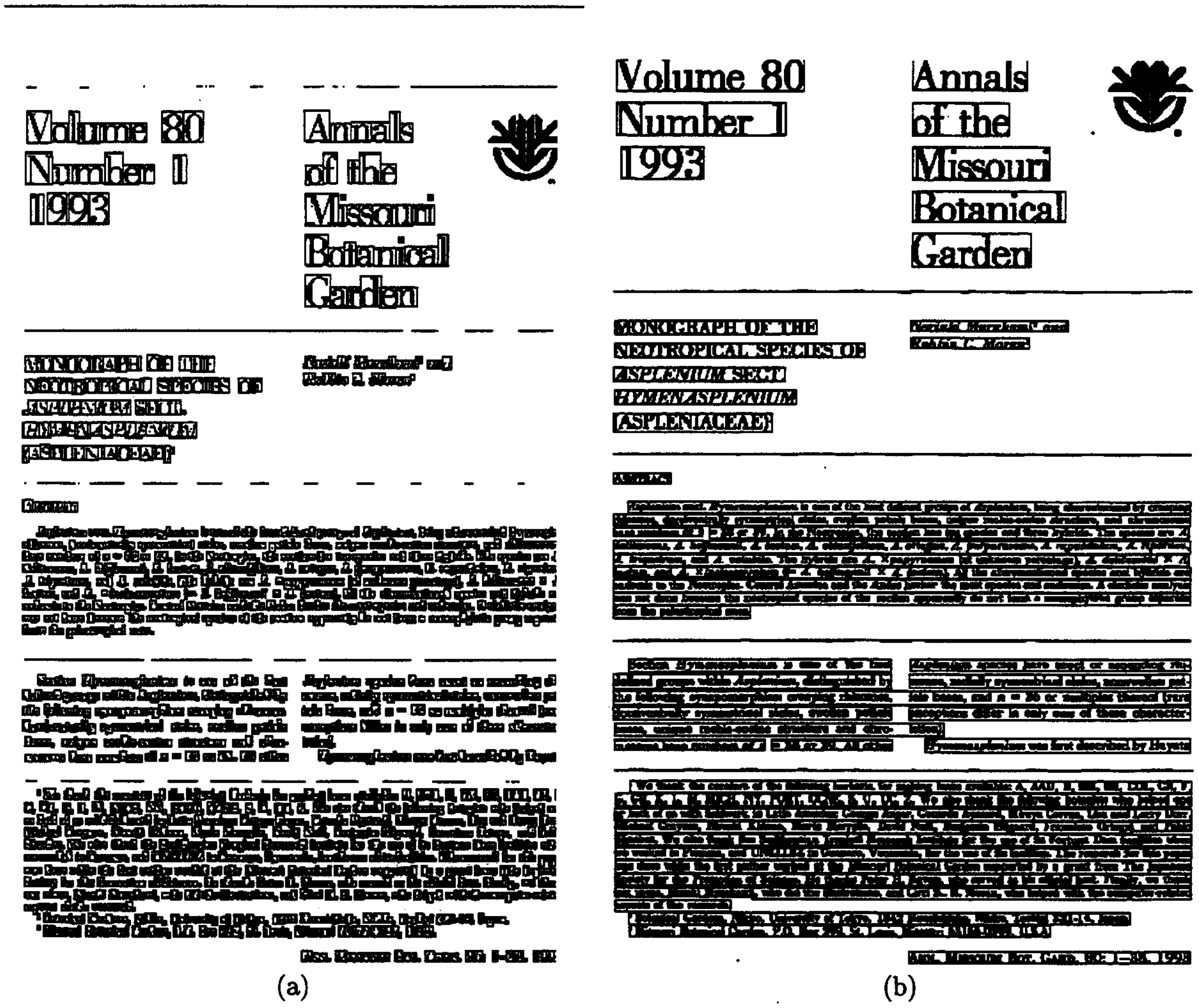


Figure 2: Illustrates a real document image overlaid with the extracted bounding boxes of (a) the connected components; and (b) the initial text line segments.

Given a set of baseline angles $\{\theta_1, \theta_2, \cdots, \theta_P\}$, the skew angle of page is estimated as

$$\theta_{page} = \text{median}\{\theta_1, \theta_2, \cdots, \theta_P\}.$$

If skew angle θ is larger than the threshold, $threshold_\theta$ the page will be rotated by $-\theta$.

For each given text-line t_i and the estimated baseline (a, b), we compute the absolute deviation of glyph from the estimated baseline

$$\sigma(t_i, a, b) = \sum_{i=1}^{N} |y_i - a - bx_i|.$$

The x-height of a text-line is estimated by taking the median of the distance from the top-left corner of each glyph box to the baseline

$$xh(t_i) = \text{median}\{d(x_i, y_i, a, b) | 1 \le i \le N\}.$$

Given the observations on text-line t_i, we can compute the likelihood that t_i has the property of a text-line

$$P(xh(t_i), \sigma(t_i, a, b)) | \text{textline}(t_i)).$$

3.3 Text-zone Formation

Horizontal Projection of the Text Line Boxes

Given a set of text-line bounding boxes $T = \{t_1, t_2, \cdots, t_M\}$, our goal is to group them into a sequence of horizontal text-regions $R = \{R_1, R_2, \cdots, R_N\}$. We do the following.

Let (x_i, y_i, w_i, h_i) represents the bounding box of the text-line $t_i \in T$. t_i is bounded by x_i and $x_i + w_i$.

Given an entity box (x, y, w, h), its horizontal projection (Figure 6) is defined as

$$\text{horz-profile}[j] = \text{horz-profile}[j] + 1, x \le j < x + w.$$

Vertical Projection of the Text Line Edges

The vertical projection of a set of entities is defined as

$$\text{vert-profile}[j] = \text{vert-profile}[j] + 1, y \le j < y + h.$$

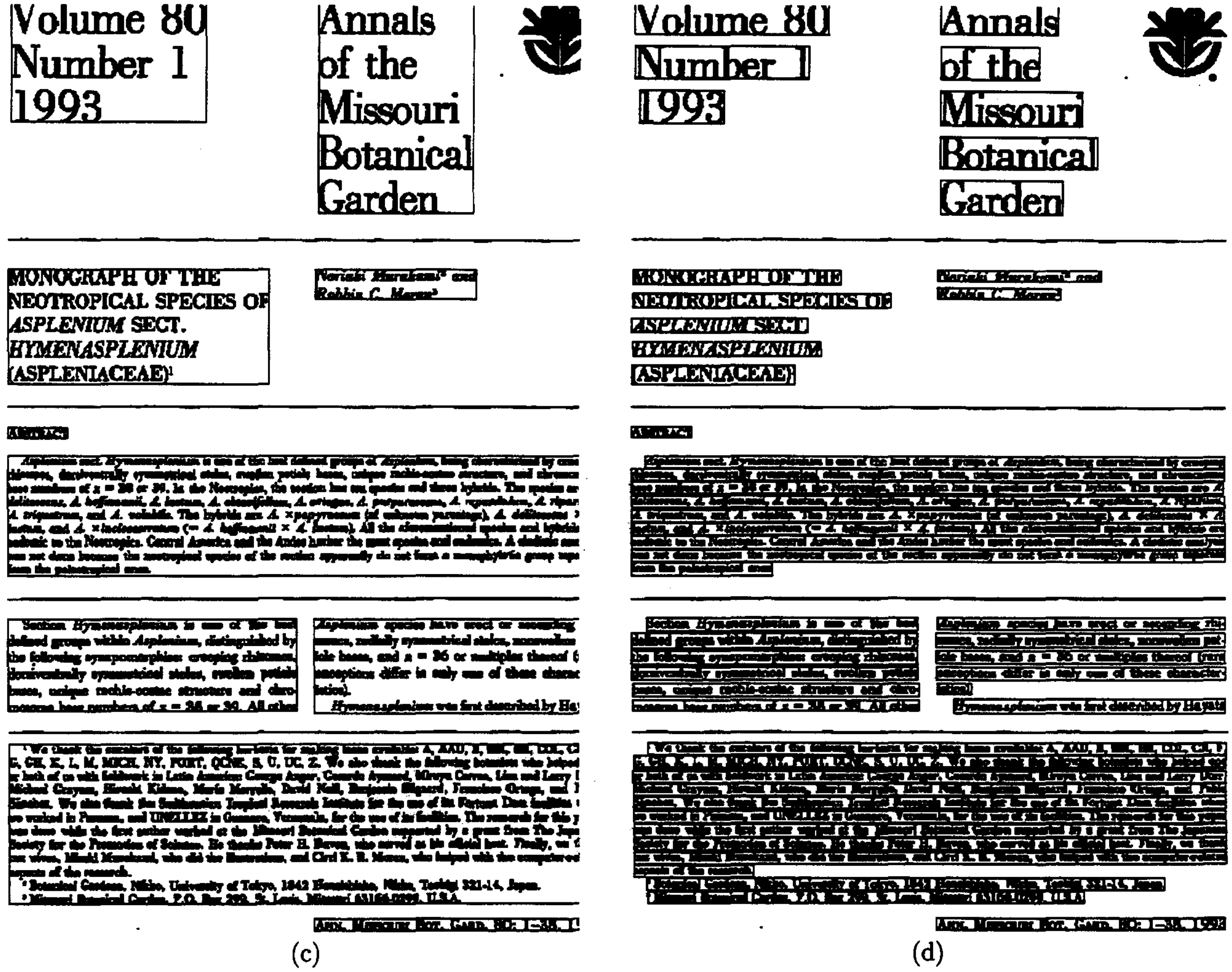


Figure 3: Illustrates a real document image overlaid with the extracted bounding boxes of (c) the text regions; and (d) the corrected text lines.

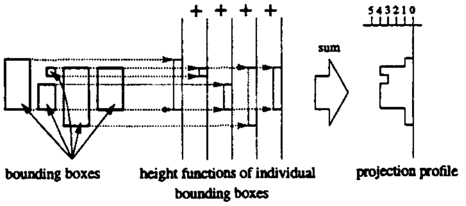


Figure 6: Illustrates the horizontal projection of bounding boxes.

Let (x_i, y_i, w_i, h_i) represents the bounding box of a text-line $t_i \in T$. We assign the left edge of t_i to be x_i, the right edge of t_i to be $x_i + w_i$, and the center of t_i to be $x_i + w_i/2$. The vertical edge projection on the three edges of the text-line bounding boxes of all $t_i \in T$ is defined as:

$$\begin{aligned} C_{left}[j] &= C_{left}[j] + 1, j = x \\ C_{center}[j] &= C_{center}[j] + 1, j = x + w/2 \\ C_{right}[j] &= C_{right}[j] + 1, j = x + w. \end{aligned}$$

Text-zone Detection Algorithm

1. Compute the horizontal projection profile of all text-line boxes.

2. Segment the page into a set of large regions, by making cut at the gaps of horizontal projection profile, where the width of gap is larger than a certain threshold. The threshold is determined by the median height of detected text-lines.

3. For each region

   (a) Compute the vertical projection count C of the left edges E_{left}, right edges E_{right}, and center edges E_{center} of text-line boxes.

   (b) Find a place which has the highest total count within its neighborhood of width w. $x = \arg_{i,j} \max(\sum_k C_{i,k}, i \in \{left, right, center\}, j - \frac{1}{2}w \leq k < j + \frac{1}{2}w)$, where w is determined by the dominant text-line height within the region.

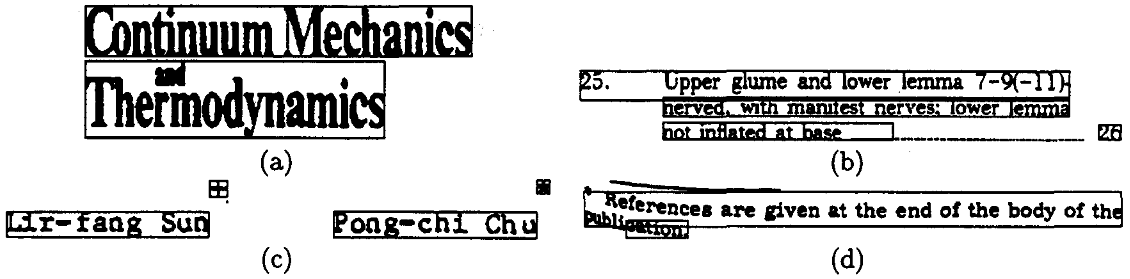

Continuum Mechanics
and
Thermodynamics
(a)
25. Upper glume and lower lemma 7-9(-11)-nerved, with manifest nerves; lower lemma not inflated at base
26
(b)
Lir-fang Sun
Pong-chi Chu
(c)
References are given at the end of the body of the publication.
(d)

Figure 4: Illustrates examples that the text detection algorithm failed.

(c) Determine the zone edge as the median of edges E_{ik}, within the neighborhood $j - \frac{1}{2}w \leq k < j + \frac{1}{2}w$.

(d) For each edge E_{ik}, finding its corresponding edge of the other side of the box $E_{jk}, j \neq i$

(e) Determine the other edges of this zone by taking the median of E_{jk}

(f) Remove the text-line boxes enclosed by the detect zone from T

(g) If $T = \emptyset$, an empty set, we are done, otherwise, repeat this step.

If the inter-zone spacing between two adjacent zones is very small, it may cause the majority of text-lines from those two zones to merge. On the other hand, a list-item structure usually has large gaps and this causes splitting errors. In order to detect these two cases, we compute the vertical projection profile of glyph enclosed by each zone.

If there is a zero-height valley in the profile, compute the probability that the region should be split into two zones

$$P(\text{twozone}(c)|w_{gap}, n, h_m, h_l, h_r, w_l, w_r),$$

where w_{gap} is the width of profile gap, n is the total number of text-lines within the current region c, h_m is the median of text-line height within c. h_l and w_l (h_r and w_r) are the height and width of the region on the left (right) side of gap. If the probability is larger than a certain threshold, split the region at the detected gap.

Given a pair of adjacent zones, the probability that they are part of the list-item structure is:

$$P(\text{list-item}(c_l, c_r)|w_{gap}, h_l, h_r, w_l, w_r, n_l, n_r),$$

where n_l and n_r are the number of text-line within the left and right zones respectively.

3.4 Text-line Splitting and Merging

Given the detected zones, we can determine if a text-line is horizontally merged or split, or vertically merged or split.

Given the observations on a text-line $t = (c_1, c_2, \cdots, c_m)$ and its neighbors $N(t)$ within the same zone Z, we compute the probability that t is vertically consistent, merged, or split:

$$P(\text{v-consistent}(t, N(t))|h(t), h_N(t), h_t(c), h_N(c)),$$

where $h(t)$ is the height of text-line t, $h_N(t)$ is the median of text-line height in zone $N(t)$, $h_t(c)$ is the median height of glyphs in t, and $h_N(c)$ is the median height of glyphs in $N(t)$. Then, we can update the linking probability between a pair of adjacent glyphs c_i and c_j:

$$\begin{aligned} P(link(i,j)) \propto\ & P(\text{sameline}(i,j)|c_i, c_j) \\ & \times P(\text{v-consistent}(t, N(t))), \end{aligned}$$

where $c_i \in t$, and $c_j \in Z$.

Given a pair of adjacent text-lines t_m and t_n within the same zone, we can update the linking probability between a pair of glyph $c_i \in t_m$ and $c_j \in t_n$:

$$\begin{aligned} P(link(i,j)) & \propto P(\text{sameline}(i,j)|c_i, c_j, \text{samezone}(i,j)) \\ & \propto P(c_i, c_j|\text{sameline}(i,j))P(\text{sameline}(i,j)) \\ & \quad \times P(\text{samezone}(i,j)|\text{sameline}(i,j)). \end{aligned}$$

Similarly, if a text-line is across two or more zones, we can update the linking probability for each pair of adjacent glyph that belong to different zones

$$\begin{aligned} P(link(i,j)) & \propto P(\text{sameline}(i,j)|c_i, c_j, \text{diffzone}(i,j)) \\ & \propto P(c_i, c_j|\text{sameline}(i,j))P(\text{sameline}(i,j)) \\ & \quad \times P(\text{diffzone}(i,j)|\text{sameline}(i,j)). \end{aligned}$$

4 Probability Estimation

Discrete lookup tables are used to represent the estimated joint and conditional probabilities used at each of the algorithm decision steps. We first quantize the value of each variable into a finite number of mutually exclusive states. If A is a variable with states $a_1, \cdots, a_n$, then $P(A)$ is a probability distribution over these states: $P(A) = (x_1, \cdots, x_n)$

where $x_i \geq 0$ and $\sum_{i=1}^{n} x_i = 1$. Here, x_i is the probability of A being in state a_i. If the variable B

has states $b_1, \cdots, b_m$, then $P(A|B)$ is an $n \times m$ table containing numbers $P(a_i|b_j)$. $P(A, B)$, the joint probability for the variables A and B, is also an $n \times m$ table. It consists of a probability for each configuration (a_i, b_j).

We conduct a series of experiments to empirically determine the probability distributions that we used to extract text lines. A tree structure quantization is used to partition the value of each variable into bins. At each node of the tree, we search through all possible threshold candidates on each variable, and select the one which gives minimum value of entropy. The total number of terminal nodes, which is equivalent to the total number of cells, is predetermined. Finally, the bins on each variable form the cells in the space. For each joint or conditional probability distribution, a cell count is computed from the the ground-truthed document images in the UW-III Document Image Database. Rather than entering the value of each variable for each individual in the sample, the cell count records, for each possible combination of values of the measured variables, how many members of the sample have exactly that combinations of values. A cell count is simply the number of units in the sample that have a given fixed set of values for the variables. The joint probability table can be computed directly from the cell count.

A few parameters, such as those thresholds used in the algorithms. Their values are estimated. A representative sample of a domain was used and a quantitative performance metric was defined. We tuned the parameter values of our algorithm and selected the set which produces the optimal performance on the input population. Assuming the criterion function is unimodal in the parameter value within a certain range, we used a golden section search method to find the optimal value within that range.

5 Experimental Results

We applied our text-line extraction algorithm to the total of 1600 images from the UW-III Document Image Database. The numbers and percentages of miss, false, correct, splitting, merging and spurious detections are shown in Table 1. Of the 105,020 ground truth text-lines, 99.76% of them are correctly detected, and 0.08% and 0.07% of lines are split or merged, respectively. Most of the missing errors are due to the rotated text.

6 Summary

In this paper, we formulate the document segmentation as a partitioning problem. The goal of the problem is to find an optimal solution to partition the set of glyphs on a given document to a hierarchical tree structure where entities within the hierarchy are associated with their physical properties and semantic labels. A unified approach is proposed. The Bayesian framework is used to assign and update the probabilities during the segmentation. An iterative, relaxation like method is used to find the partitioning solution that maximizes the joint probability.

A text-line extraction algorithm has been implemented to demonstrate the usage of this framework. This algorithm consists of two major components – off-line statistical training and on-line text-line extraction. The probabilities used within this algorithm are estimated from an extensive training set of various kinds of measurements of distances between the terminal and non-terminal entities with which the algorithm works. The off-line probabilities estimated in the training then drive all decisions in the on-line segmentation module. The on-line segmentation module first extracts and filters the set of connected components of the input image to obtain a set of glyphs. Each glyph is linked to its adjacent neighbor to form glyph pars. Associated with each link is the pair's linking probability. The entire text-line extraction process can be viewed as an iterative re-adjustment of the pairs' linking probabilities on the glyph set. The segmentation algorithm terminates when the decision can be made in favor for each link within the final set of text-line segments.

The algorithm was tested on the 1600 pages of technical documents within the UW-III database. A total of 105020 text lines within these pages, the algorithm exhibits a 99.8% accuracy rate. Currently, we are implementing a text-block extraction algorithm, also using the proposed framework. This new algorithm is currently at the testing phrase and the prelimary result looks promosing.

References

[1] S. Srihari and W. Zack, Document Image analysis, *Proceedings of the 8th International Conference on Pattern Recognition (ICPR'86)*, pp. 434-436, July 1986, Paris, France.

[2] N. Amamoto, S. Torigoe and Y. Hirogaki, Block Segmentation and Text Area Extraction of Vertically/Horizontally Written Documents, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 739-742, October 1993, Tsukuba, Japan.

[3] M. Okamoto and M. Takahashi, A Hybrid Page Segmentation Method, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 743-746, October 1993, Tsukuba, Japan.

Table 1: Performance of text-line extraction algorithms.

	Total	Correct	Splitting	Merging	Mis-False	Spurious
Ground Truth	105020	104773 (99.76%)	80 (0.08%)	78 (0.07%)	79 (0.08%)	10 (0.01%)
Detected	105019	104773 (99.77%)	172 (0.16%)	37 (0.04%)	25 (0.02%)	12 (0.01%)

[4] T. Saitoh, M. Tachikawa and T. Yamaai, Document Image Segmentation and Text Area Ordering, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 323-329, October 1993, Tsukuba, Japan.

[5] D.J. Ittner and H.S. Baird, Language-Free Layout analysis, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 336-340, October 1993, Tsukuba, Japan.

[6] Y. Hirayama, A Block Segmentation Method for Document Images with Complicated Column Structures, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 91-94, October 1993, Tsukuba, Japan.

[7] T. Pavlidis and J. Zhou, Page Segmentation and Classification, *CVGIP, Graphical Models and Image Processing*, Vol. 54, pp. 484-496, November 1992.

[8] L. OGorman, The Document Spectrum for Page Layout Analysis, *IEEE Transactions of Pattern Analysis and Machines Intelligence*, pp. 1162-1173, November 1993.

[9] G. Nagy and S. Seth, Hierarchical Representation of Optically Scanned Documents, *Proceedings of the 7th International Conference on Pattern Recognition (ICPR'84)*, pp. 347-349, July 1984, Montreal, Canada.

[10] H.S. Baird, Background Structure in Document Images, *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 1013-1030, October 1994.

[11] F. Jones and J. Litchter, Layout Extraction of Mixed Mode Documents, *Machines Vision and Applications*, Vol. 7, No. 4, pp. 237-246, 1994.

[12] S-Y. Wang and T. Yagasaki, Block Selection: A Method for Segmenting Page Image for Various Editing Styles, *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95)*, pp. 128-133, August 1995, Montreal, Canada.

[13] F. Esposito, D. Malerba and G. Semeraro, A Knowledge-Based Approach to the Layout Analysis, *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95)*, pp. 466-471, August 1995, Montreal, Canada.

[14] J. Ha, R.M. Haralick and I. Phillips, Document Page Decomposition by the Bounding-Box Projection, *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95)*, pp. 1119-1122, August 1995, Montreal, Canada.

[15] S. Chen, R.M. Haralick and I. Phillips, Extraction of Text Lines and Text Blocks on Document Images Based on Statistical Modeling, *International Journal of Imaging Systems and Technology*, Vol. 7, No. 4, pp. 343-356, Winter, 1996.

[16] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[17] S. Chen, R.M. Haralick and I. Phillips, Automatic Text Skew Estimation in Document Images, *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95)*, pp. 1153-1156, August 1995, Montreal, Canada.

[18] I. Phillips, *Users' Reference Manual*, CD-ROM, UW-III Document Image Database-III, 1995.

[19] I. Phillips, S. Chen and R. Haralick, CD-ROM Document Database Standard, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 478-483, October 1993, Tsukuba, Japan.

[20] I. Phillips, J. Ha and R. Haralick and D. Dori, The Implementation Methodology for the CD-ROM English Document Database, *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 484-487, October 1993, Tsukuba, Japan.

Model-Directed Document Image Analysis

Henry S. Baird

Xerox Palo Alto Research Center,
3333 Coyote Hill Road, Palo Alto, CA 94304 USA

E-mail: baird@parc.xerox.com

Abstract

If current OCR engineering trends continue, then, we believe, "general-purpose" — fully automatic and nonretargetable — systems will leave many potential users unsatisfied, and lucrative application niches unfilled, for years to come. However, for users who care enough to volunteer some manual effort — to help customize the system to their document(s) — significantly higher accuracy may be achievable, without delay. We discuss in detail two state-of-the-art document recognition systems — Lucent Technologies' Table Reader System (TRS) and Xerox's "document image decoding" (DID) research prototype — which yield high accuracy by reliance on explicitly stated models of properties of the target document, whether iconic (known typefaces and image degradations), geometric (restricted classes of layouts), or symbolic (linguistic and pragmatic contextual constraints). How great are the performance advantages that can be realized by sacrificing automation in these ways? To what extent can the necessary customizations be (semi-)automated? We outline recent and planned research at Xerox PARC motivated by these questions.

1 Performance of Current OCR Systems

The dominant type of present–day commercial OCR system, whether on the desktop or in service–bureau settings, is designed to operate fully automatically, refusing to accept guidance from the user. The majority of desk–top users welcome this since they are untrained and impatient with inconvenience. There is a similar reliance on more or less completely automatic operation in almost all of the highly specialized OCR application niches such as postal–code and financial–document processing, even though their costly equipment is tended by trained staff in controlled service–bureau settings. In this case, it is largely the daunting throughput requirements that dictate fully automatic operation.

Both of these user communities — the casual SOHO users and the sophisticated special-document users — tolerate surprisingly low performance. The latest competitive studies, at UNLV in 1996 [1], showed, for example, that desk-top OCR packages misrecognize 3–15% of characters — an intolerably high error rate, most users would agree — in over 40% of magazine pages: for other document categories, performance was far worse. The best current systems for reading hand–written courtesy amounts on checks [2] are tuned to reject 33–55% of the input in order to hold substitution errors below 1%. Similarly, the best handwritten postal-address readers fail to "finalize" 35% of the input [3].

All of these technologies are improvable, of course, and are improving: but slowly and at a high cost. The UNLV data suggested that the best desk–top OCR machines have been cutting character error rates by about 15–20% per year [1]. Every sanguine person hopes for sudden breakthroughs in performance – and individual researchers characteristically hope that these will result from isolated technical innovations – but the record of the last ten years does not encourage-such hopes.

Instead, the pattern I see is that overall performance of these increasingly complex systems does not dramatically improve as a result of any single *localized* improvement: say, a more accurate character classification algorithm, or a more refined linguistic model, or a more robust layout segmentor. On the contrary, as year after year the weakest components have been most improved, we are entering a regime where the origins of errors are more evenly distributed among the components of the system. The principal driver of improvements is large-scale empirical testing by ever-growing test data bases, followed by tedious manual analysis of failure cases. These training and testing databases are certainly large and growing larger, but it is not feasible to collect them systematically enough to guarantee coverage of the full cross-product of ranges of typefaces, type sizes, image degradations, layouts styles, scripts, languages, etc etc that occur in practice. Even worse from an engineering pont of view, it is becoming increasingly problematic to isolate one cause, or even the dominant cause, of specific failures. The cause is more and more often a subtle 'conspiracy' among the components of the system which is hard to understand. 'Fixing' one problem breaks another. So for multiple reasons it is often not clear even to the researchers and engineers most familiar with the internals of the machine where they should apply their next year's efforts to achieve the largest gain. Too often, all that can be found to work is a specific manual patch for that particular case. The systems are growing monotonically in the number of lines of code and the number of modules with specialized functions.

The perceptions I list above are not mine alone. At the IAPR DAS'98 Workshop in Nagano, Japan, I took the opportunity to ask three engineering managers of world-class OCR systems about their rate of progress, and the most serious obstacles to progress that they face. Most of them agreed with most of the points above.

I do not mean to paint the bleakest possible picture of the future. Ingenious researchers and engineers continue to solve hard problems. If systems complexity bogs us down, certainly Moore's law buoys us up.

But, overall I feel that OCR engineering current trends support these conclusions:

- the search for more strongly general-purpose, higher-performance document recognition systems will continue to absorb large engineering resources and continue to yield only incremental overall performance improvements;
- since no one system, in markets with many players, is able to sprint ahead of the others, competition on technical grounds will not slacken; and
- most players will have no choice but to continue incremental refinements within their idiosyncratic, slowly evolving, and increasingly complex system architectures.

This is bad news for the many users whose particular documents are poorly served by current machines. They may have to wait years for technology that performs adequately on their class of documents. Potentially lucrative application niches will remain unfilled.

2 The Case for Model-Directed Recognition

One way to summarize the state of the art of OCR systems is that we cannot now, and will not for many years, simultaneously achieve these three desirable properties:

- *high accuracy*, i.e. near-perfect character-by-character transcription;
- *versatility*: applicability to many types of documents, image qualities, etc; and
- *full automation*, requiring no assistance from the user.

How then can research help these many underserved users in the near term?

What if we relax one or more of these goals? If, for example, we attack problems that do not require high accuracy, can we achieve versatility and automation? Yes, clearly: one example is the use of OCR as a front end for word-token-based information retrieval. It has been amply documented that recall and precision are little affected by OCR error rates [4].

What if we sacrifice versatility? There are hosts of successful examples of this approach, from the adoption of the OCR-A font standard to special-forms readers.

What if we sacrifice automation, and so ask the user to intervene manually for each document (or document class)? This is the alternative research direction discussed in this paper. It is, of course, not new: in fact, it was already reasonably well articulated in May of 1992 by a few DIA researchers attending the first DARPA-funded Document Understanding Workshop, held at Xerox PARC. The "Model-Based OCR" panel of the workshop included Phil Chou, Andrew Gilles, Dan Huttenlocher, Tapas Kanungo, Gary Kopec, Prasana Mulgaonkar, Theo Pavlidis, Azriel Rosenfeld, Sargur Srihari, Steve Munt, Steve Dennis, and the present author. We were excited by the potential of a research program that somehow would exploit explicitly specified and often detailed models of the input document in hopes of achieving far higher performance (accuracy and speed) and versatility (range of documents handled) than any of the then–existing systems or their likely successors.

This panel concluded by recommending that DARPA encourage the development of:

(a) "a core technology in which all the assumptions about the writing system, language constraints, context, are explicit such that they can be replaced by new modules [...];"

(b) "alternative architectures and algorithms including promising novel approaches whose initial performance is inferior [...];"

(c) "uniform technology which is transportable across a variety of writing systems [...];" and

(d) "a core technology for developing and using explicit, quantitative, parameterized models of [image] distortion [...]".

It is remarkable to look back, six years later, and see with what tenacity a few of us — Gary and Phil at PARC; myself, David Ittner, and Tin Kam Ho at Bell Labs; and Tapas and Bob Haralick at Univ. Washington — struggled to realize these dreams. Gary and Phil seemed to me to be most committed to goals (a) and (b), while Tapas, Bob, and the Bell Labs folks focussed on (c) with a low-level but persistent pursuit of (d).

All four of these goals were felt to be dauntingly ambitious at the time. They were crafted in conscious contrast to the engineering — and research — methodologies dominant at that time. They are, in fact, continuing today. At considerable risk of oversimplification, and with no desire to understate the creativity, skill, and energy with which they have been pursued, I may characterize them as follows. The emphasis is on modularization of OCR systems into (typically) a pipeline of specialized components performing physical layout analysis and interpretation, isolated-character classification, hypothesize-and-test word segmentation, and contextual analysis. Each of these components is developed to a large degree in isolation from the others. With the exception of image classification and some aspects of contextual analysis, they are not trainable by example but must be substantially hand crafted and manually tuned for good results. They are rarely based on an explicit model of the class of documents to be read, so there is no escape from large-scale (but still unsystematic) empirical testing regimes which inevitably escalate to the limits of affordability. No matter how well the components perform in isolation, their integration is an unpredictable and often frustratingly unstable engineering exercise.

The end result of these dominant methodologies, for most leading OCR technology developers, has been a large and steadily growing software suite which is difficult to improve systematically and which therefore drains larger and larger engineering resources in return for chronically incremental performance improvements. As tempting as it must often be to restart from scratch and rearchitect more rationally, their large investment in code and the uncertainties of the OCR state of the art argue against radical course corrections. It was this morass of individually plausible but collectively *ad hoc* methods that the panel foresaw and were trying to circumvent.

What progress has been made towards these four "Model-Based OCR" goals, and what should be attempted next? The rest of this paper gives a partial answer to these questions: partial in that it emphasizes work in which the author has been, and remains, personally involved.

The next two sections describe two model-directed OCR systems which embody many of these principles. The first is a retargetable table-reader product developed by a team in Bell Laboratories (including the present author), first used on a large scale within AT&T, and now offered for sale by Lucent Technologies. The second is an experimental prototype within Xerox PARC, whose development was led by Gary Kopec and Phil Chou, and which has been successfully applied to a variety of uniquely challenging documents, especially in the context of the UC Berkeley Digital Library Initiative project. Although Phil has left Xerox and Gary died in December 1998, extensions and refinements of the DID system remain active topics of research at PARC by a team that includes the present author. We list a number of open research problems, engineering challenges, and opportunities for feasibility trials and joint work.

3 A Retargetable Table Reader

At least one model–directed, manually retargetable document image analysis system exists and is heavily used today. It is a system for reading machine–printed documents in known predefined tabular–data layout styles [5] (telephone bills, to be precise). In these tables, textual data are presented in 'record' lines made up of fixed–width fields. Tables often do not rely on line–art (ruled lines) to delimit fields, and in this way differ crucially from fixed forms. This table–reader system performs these steps: identifies multiple tables per page; identifies records within tables (ignoring non–record text); segments records into fields; and recognizes characters within fields, constrained by field–specific contextual knowledge.

Obstacles to good performance on these tables included small print, tight line–spacing, poor–quality text (such as photocopies), and line–art or background patterns that touch the text. Precise skew–correction and pitch–estimation, and high–performance OCR using neural nets proved crucial in overcoming these obstacles. However, the principal obstacle to building a system of this sort was the wide variability of layouts among the hundreds of table form types encountered. The variability would overwhelm any fixed, fully automatic system; if each distinct "form model" had to be manually specified, then the retargeting effort must be small and "deskilled." Therefore the most significant technical advances in this work appear to be algorithms for identifying and segmenting records with ***known layout***, together with the integration of these algorithms with an efficient graphical user interface (GUI) for *defining new layouts*.

Unlike most prior work on forms and table analysis, the system does not depend on guidance from line–art or fiducial marks. The operator describes a new layout model by annotating images of a sample page (noting the location of fields, and whether certain characters are required or optional, etc). This example is thus abstracted into "record–line template" which is matched (using simple convolution–based methods) to every text–line in the image, to distinguish record lines from non–record text and to splt each record line into fields. The model–specification GUI has been ergonomically designed to make efficient and intuitive use of exemplary images, so that the skill and manual effort required to retarget the system to new table layouts are held to a minimum. In fact, each tabular layout model can typically be specified in less than 15 minutes by a clerk with data–entry skills.

In short, the system succeeds because a user can quickly specify a layout model which can then be effectively and fully automatically applied to every page of tables of the same layout. The system has been applied in this way to more than 400 distinct tabular layouts. Over a period of three years the system read over fifty million records with high accuracy. Large scale tests have shown that the system fully automatically achieves 97% to 99.98% characters correct. The GUI also supports manual correction, which typically yields a semi-automatic accuracy of greater than 99.99%.

This performance is so much higher than any previously published on tables, and the range of table–types handled is so much greater than any previous commercial table–reader system, that it is tempting to assert that the key determinants of success were (a) restriction to known predefined layouts and (b) exploitation of field–specific context. That is, manual specification and automatic exploitation of detailed models.

Thus, this table reader system (now offered for sale by Lucent Technologies) is an example of a model–directed OCR system of the type we envisaged. It has successfully colonized a previously underserved application niche.

It is significant that this application niche is a service-bureau operation, where the operating staff (however non-technical their entry skills) can be trained and managed, and where engineers are available to back them up in the occasional difficult case. This is a far cry from desk-top casual-use OCR.

4 The Document Image Decoding Prototype

As early as 1990 Gary Kopec and Phil Chou of Xerox PARC were consciously adapting to

OCR the paradigms characteristic of the early days of signal processing research, especially the communications–theory framework [6]: applied to document images, this views any observed document image as a signal which has been synthesized through several distinct stages: the underlying message (e.g. the ASCII text) is first "encoded" as an ideal image by choices of typefaces and page layout, and this ideal image is, in turn, "degraded" by noise introduced during printing and scanning, yielding the observed image. Recognition is then viewed, in this framework, as an attempt to "decode" the observed signal by estimating the most probable transmitted message, among all messages implied by the models, that may have led to it. The models of encoding that Gary and Phil used usually involved probabilistic finite–state machines and rigid character template images. The typical model of degradation was probabilistic asymmetric bit-flip.

Gary and Phil's collaboration was, it seems to me, distinguished from the work of their peers most clearly by two principles:

- every stage of the system is explicitly modeled; and
- the system, as a whole, is simultaneously optimized by minimizing the expected "loss" between the message sent and the message decoded.

Everyone else in the DIA field — including myself — backed away from one or both of these principles, at times, in the face of theoretical difficulties or from a desire to exhibit a near-term practical success.

In the face of many technical difficulties Gary, Phil, and their collaborators managed to illustrate many strengths of this approach [7,8,9,10,11]. They showed that their family of encoding models - probabilistic regular grammars, sometimes attributed — was rich enough to capture not only plain text but textual markup, logical layout labeling, highly structured technical text and tables, and mathematical expressions — even music notation. By insisting that the system be optimized simultaneously as a whole, not a single component at a time, they obviated several artificial distinctions — notably between recognition and segmentation of characters — which trigger complexity, confusion, and errors in other systems. They showed that the optimal decoding (for a 0–1 loss function) could be approximately found by a segmental Viterbi search through the 2-D trellis implied by the composition of the synthesis models. The models were formally and practically separate from the recognition (search) engine, and as a result many ways were found to improve (e.g. speed up) the search engine independent of any model. They found ways to infer some aspects of the models — e.g. character bi–gram probabilities and character templates — automatically from ground–truthed training data (using maximum–likelihood estimation), thus reducing the effort to retarget the system to particular documents.

Perhaps most impressively, from the point of view of potential users of the system, they showed repeatedly that it could *drop the character error rate, by up to an order of magnitude* in many cases, compared to commercial OCR systems. There are well-understood technical reasons for this extraordinary advantage. Our decoding algorithm gives, by rigorous probabilistic search, the best possible result given the model and the scanned image: the result is exactly that data which is most likely to give rise to the printed and scanned image. Thus although our results can be improved using a better model — a more complex, more specific model that fits the document better — nevertheless whenever we use a specific model we do as well as possible consistent with it.

Further, by judicious use of attributed grammars in modeling the encoding stage, the logical structure of text — e.g. the functional parts of a dictionary entry — can be captured and preserved, as a beneficial side–effect of recognition. Few if any commercial OCR systems offer such a feature; the manual effort to add the structural tags to the plain ASCII that they produce is usually prohibitive.

As of a year ago, certain weaknesses were nevertheless still apparent. The asymmetric bit-flip model of degradation had proven brittle in practice; later extensions to "multi-level templates" allowed close approximation of arbitrary blur and additive noise, but not to other common degradations such as affine distortions. First attempts to incorporate language models richer than unigram character probabilities caused an explosion in time complexity. In spite of the fact that, given a modest amount of ground-truthed training data, character templates could be learned almost fully automatically, it was still the case that the manual effort and technical skill required to use the system was often excessive. Compounding this was the fact that the system was composed from routines in several languages (both C and LISP).

But perhaps the most serious deficiency of the system was its low speed: it often ran two orders of magnitude or more slower than competing commercially available systems.

Happily, within the past year, significant progress has been made on some of these fronts. Algorithmic improvements to the search — not yet published — have yielded an *order of magnitude speed-up*, with no loss of accuracy or generality, over a large test set. All of the system components needed for ordinary use (on, e.g., English text) is now written entirely in Python and C, is readily portable to several computing platforms, and is thus able to be shared with collaborators.

This system is now ready for further feasibility trials. Some trials will be carried out this summer, in close association with the UC Berkeley Digital Library Initiative project. We are selecting one or more botanical reference books which are effectively illegible by commercial OCR systems for various reasons (uncommon typefaces, low image quality, or highly structured text), and whose contents are not yet on–line and would complement the already large and useful data base assembled in the UCB 'CalFlora' website (cf. http://elib.cs.berkeley.edu/calflora/botanical.html). We intend to retarget the DID system to each of these books, and thus provide, through the UC Berkeley Digital Library, unique scholarly resources to the botanical research community, years earlier than existing commercially available OCR systems could make possible.

So, in summary, our present technology offers a tradeoff: far higher accuracy and (uniquely) preservation of structure *versus* some manual start-up effort and significantly longer runtimes. This contrasts with current commercial OCR packages, which require no manual effort and are much faster, but which are oblivious to the document's structure and whose accuracy is fixed and unimprovable. If their error rate happens to be too high on your document, you have no way, short of manually correcting the output, to improve it. The actual trade-offs that are achievable in practice with the DID system appear to depend strongly on details of each document and the workflow surrounding it.

We understand in general terms how to pick different operating points on the DID trade-off curves: for example, how to reduce error by using more complex, and therefore more restrictive, grammars. More complex grammars not only often reduce error, but they allow more refined tagging of the output. Generally the more complex the grammar and the more symbols and typefaces that are expected, the slower and more expensive the decoding: but we are exploring new heuristics that promise speed-ups with no sacrifice of accuracy or tagging.

The most promising immediate future directions for DID research, it seems to us in the DID area at PARC, include:

- incorporating language models inferable from corpora, without large speed penalties;
- incorporating more realistic image degradation models (e.g. [12] or [13]); and
- further 'deskilling' of the retargeting task to bring it within the reach of non-expert users.

We believe the time has come to look outside PARC for commercially attractive applications where these trade-offs can be concretely explored. Here is a sketch of a possible field trial of the decoder software, as part of a semi-automatic workflow requiring the conversion of a sequence of documents to text with an accuracy far higher than commercially available OCR systems can uniformly provide.

> The engineer in the field, at first working closely with PARC, will:
>
> 1. select, from the set of documents to be converted, those which are most likely to benefit from decoding: these will typically be relatively long (tens or hundreds of pages) and possess uniform printing characteristics (e.g. only a few fonts and type sizes, and similar image 'quality');
> 2. manually transcribe – or merely correct the commercial-OCR output of – a subset of each document (a few pages at most);
> 3. run our automatic typeface-inference tool;

4. specify the document layout grammar and design the output encoding, by editing a special file (for many documents, a good model may already exist, and can be merely taken 'off the shelf'); and
5. run the decoder on the complete documents, for far higher accuracy and detailed structural tagging.

We would be happy to discuss joint feasibility trials or collaborative research with interested parties.

5 Conclusions

We have argued that present OCR engineering practice will leave many potential users underserved for years to come. In the meantime, motivated users who are willing to invest some effort in manually customizing a "retargetable" OCR system to their (class of) document(s) may succeed. We have shown that model-directed, manually retargetable OCR systems have made substantial progress since their inception almost a decade ago. Successful applications have been built: at least one is heavily used. Laboratory prototypes are making steady progress, and are ready for extended feasibility trials.

Acknowledgements

The Table Reader System was developed by John Shamilian, Tom Wood, and the present author, with significant early help from Michele Battista (all then at Bell Labs). The DID system was the creation primarily of Gary Kopec and Phil Chou (both then at Xerox PARC). The DID area at PARC in the past year included Gary Kopec, Dan Bloomberg, Les Niles, Kris Popat, and the present author. Kris Halvorsen, their Lab manager, supported and encouraged the research direction described here. We are indebted also to Profs. Robert Wilensky and Richard Fateman, and to Taku Tokuyasu, of the UC Berkeley Digital Library project.

References

[1] S. V. Rice, F. R. Jenkins, & T. Nartker, "The Fifth Annual Text of OCR Accuracy", *UNLV Information Science Research Institute Annual Report*, Las Vegas, NV, April 1996.

[2] C. Y. Suen, K. Liu, & N. W. Strathy, "Sorting and Recognizing Cheques and Financial Documents," *Proc., IAPR 1998 Workshop on Document Analysis Systems (DAS'98)*, Nagano, Japan, pp. 1-18, November 1998.

[3] A. Filatov, N. Nikitin, A. Volgunin, & P. Zelinsky, "The AddressScript Recognition System for Handwritten Envelopes," *Proc., IAPR 1998 Workshop on Document Analysis Systems (DAS'98)*, Nagano, Japan, pp. 222-236, November 1998.

[4] J. Shamilian, H. S. Baird, & T. Wood, "A Retargetable Table Reader," *Proc., IAPR 1997 Int'l Conf. on Document Analysis and Recognition*, Ulm, Germany, August 18-20, 1997.

[5] K. Taghva, J. Borsack, A. Condit, S. Erva, "The Effects of Noisy Data on Text Retrieval", *Journal of the American Society for Information Science*, pg. 50-58, vol. 45, 1994.

[6] G.E. Kopec and P.A. Chou, "Document image decoding using Markov source models," Asilomar Conf. On Signal, Systems, and Computers, October 1992. Expanded for publication in *IEEE Trans. Pattern Analysis and Machine Intelligence*, June 1994.

[7] G.E. Kopec, P.A. Chou, and D. Maltz, "Markov source models for printed music decoding," *J. Electronic Imaging*, January 1996.

[8] A. Kam and G.E. Kopec, "Document image decoding by heuristic search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, September 1996.

[9] G.E. Kopec and M. Lomelin, "Supervised template estimation for document image decoding," *IEEE Trans. Pattern Analysis and Machine Intelligence*, December 1997.

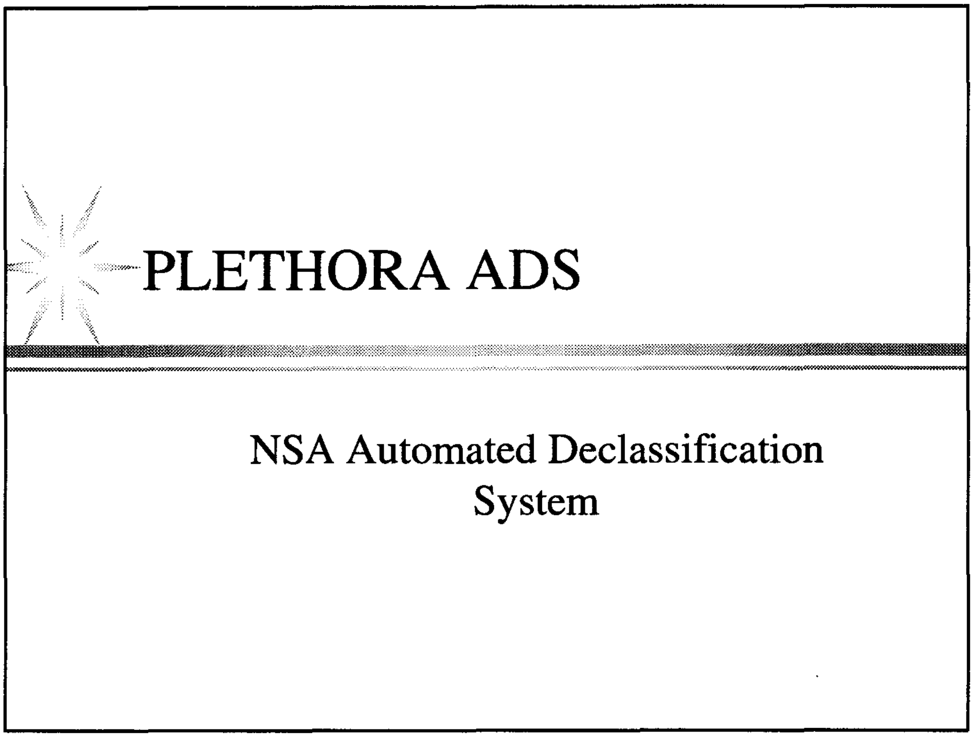
[10] G.E. Kopec, "An EM algorithm for character template estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, submitted for possible publication.

[11] G.E. Kopec, "Multilevel character templates for document image decoding," *Proc. SPIE Document Recognition IV*, San Jose, CA, February 1997.

[12] H. S. Baird, "Document Image Defect Models," in H. S. Baird, H. Bunke, and K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag: New York, 1992, pp. 546-556.

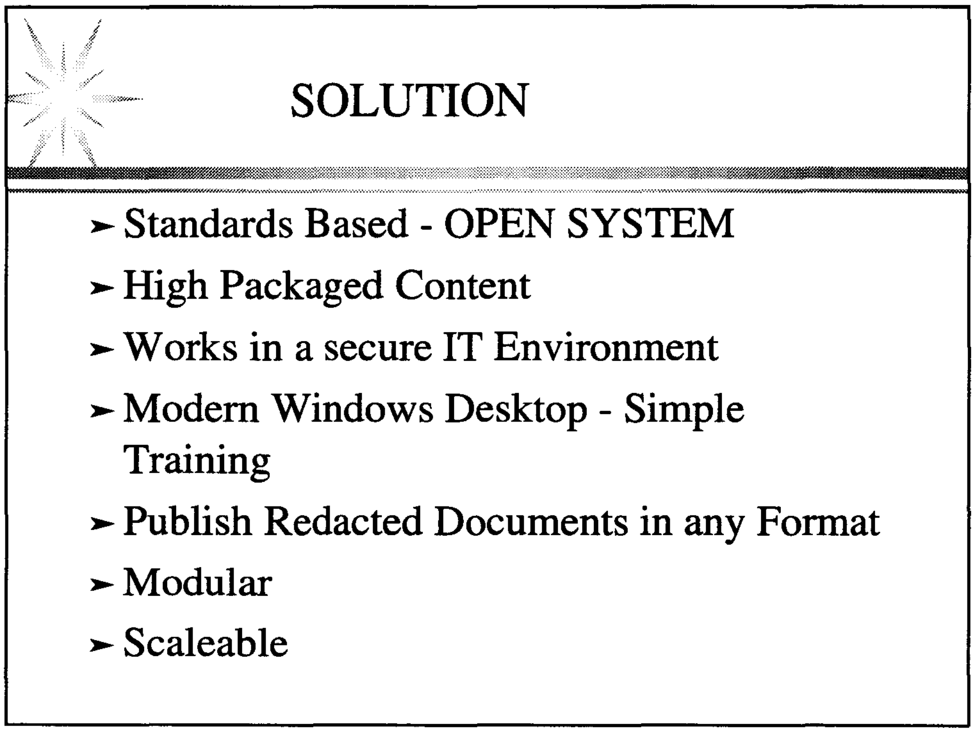
[13] T. Kanungo, R. M. Haralick, and I. Phillips, "Global and Local Document Degradation Models," *Proceedings, IAPR 2nd ICDAR*, Tsukuba, Japan, October 20-22, 1993.

Declassification



PLETHORA ADS

NSA Automated Declassification
System



SOLUTION

- ➢ Standards Based - OPEN SYSTEM
- ➢ High Packaged Content
- ➢ Works in a secure IT Environment
- ➢ Modern Windows Desktop - Simple Training
- ➢ Publish Redacted Documents in any Format
- ➢ Modular
- ➢ Scaleable

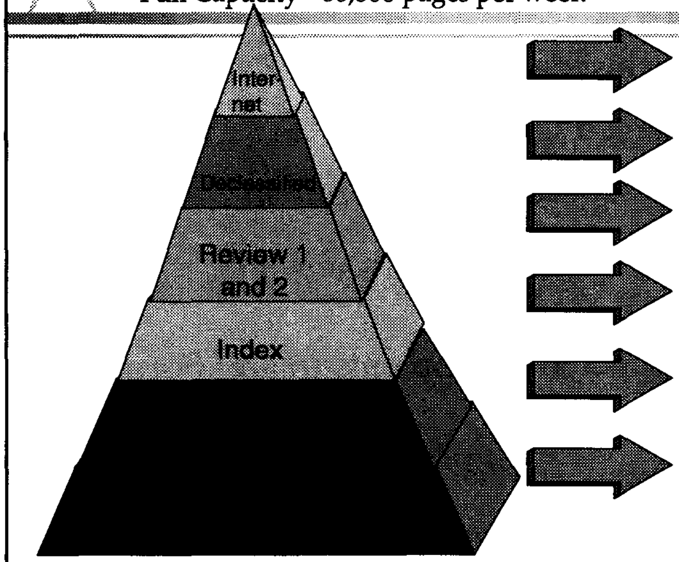
PROJECT PLETHORA MILESTONES

➢ Limited Operational Capability	Oct 97
➢ Initial Operational Capability	Mar 98
➢ Integree Declassifiers on board	Sep-Nov98
➢ Final Operational Capability	Late 99

ADS PRODUCTION STATISTICS
As of Feb 99

Full Capacity - 60,000 pages per week

PAGES COMPLETED



0

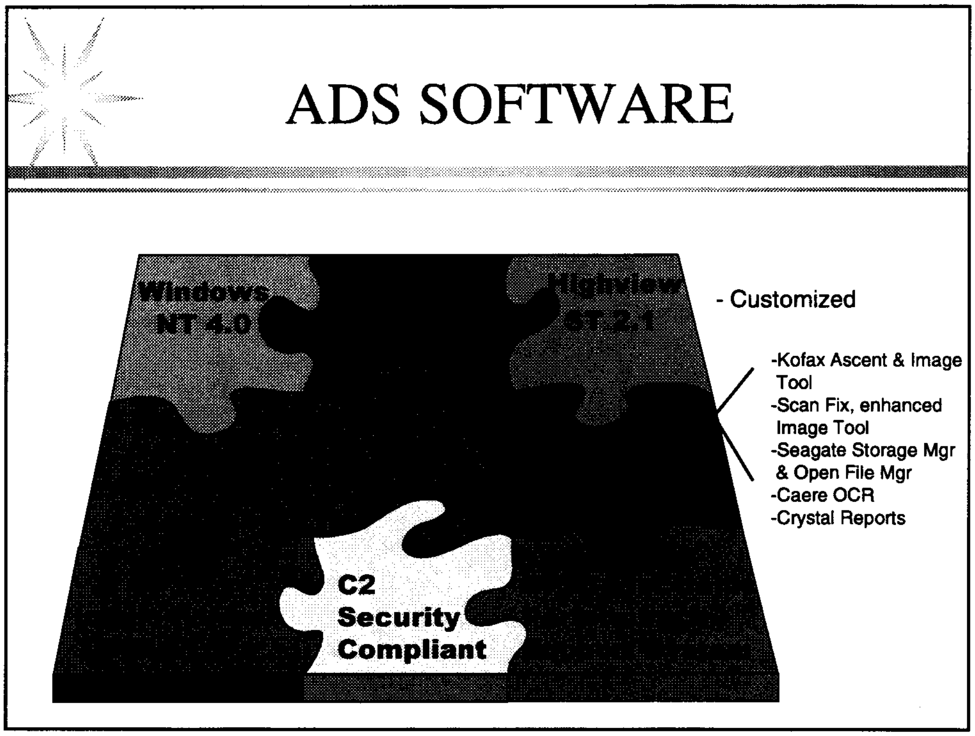
165,000

450,000

1,000,000

1,250,000

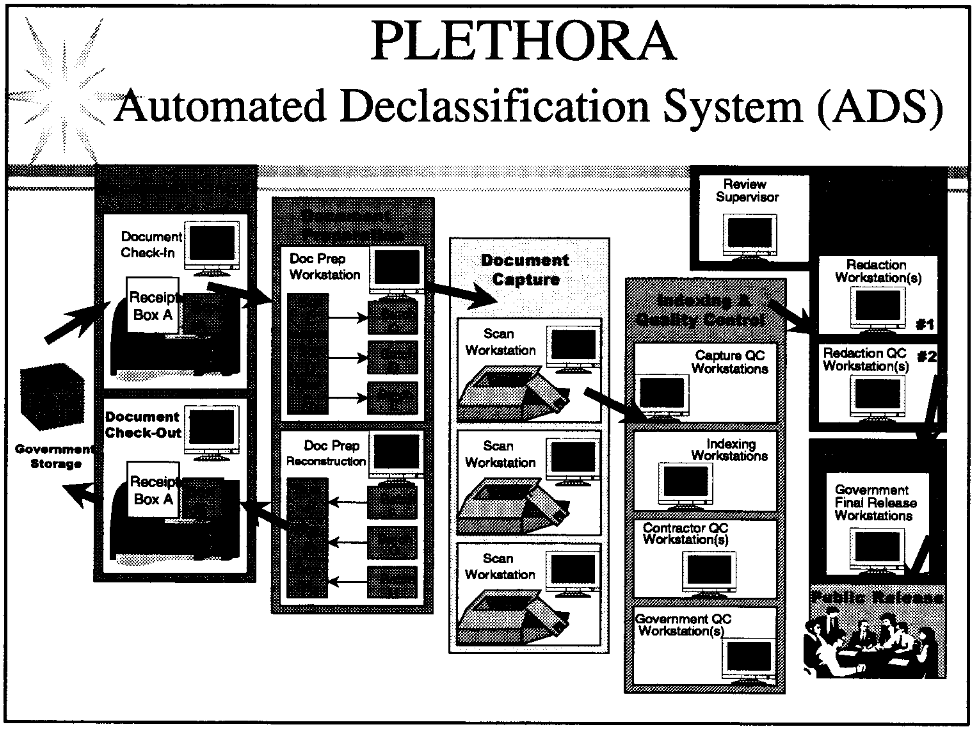
1,750,000



ADS SOFTWARE

- Customized

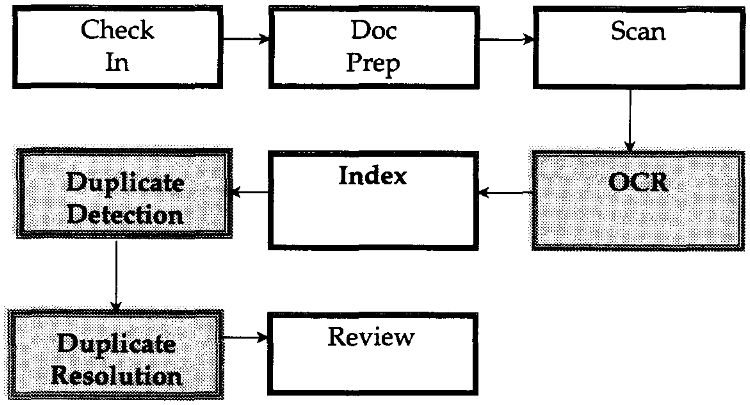
-Kofax Ascent & Image Tool
-Scan Fix, enhanced Image Tool
-Seagate Storage Mgr & Open File Mgr
-Caere OCR
-Crystal Reports



PLETHORA

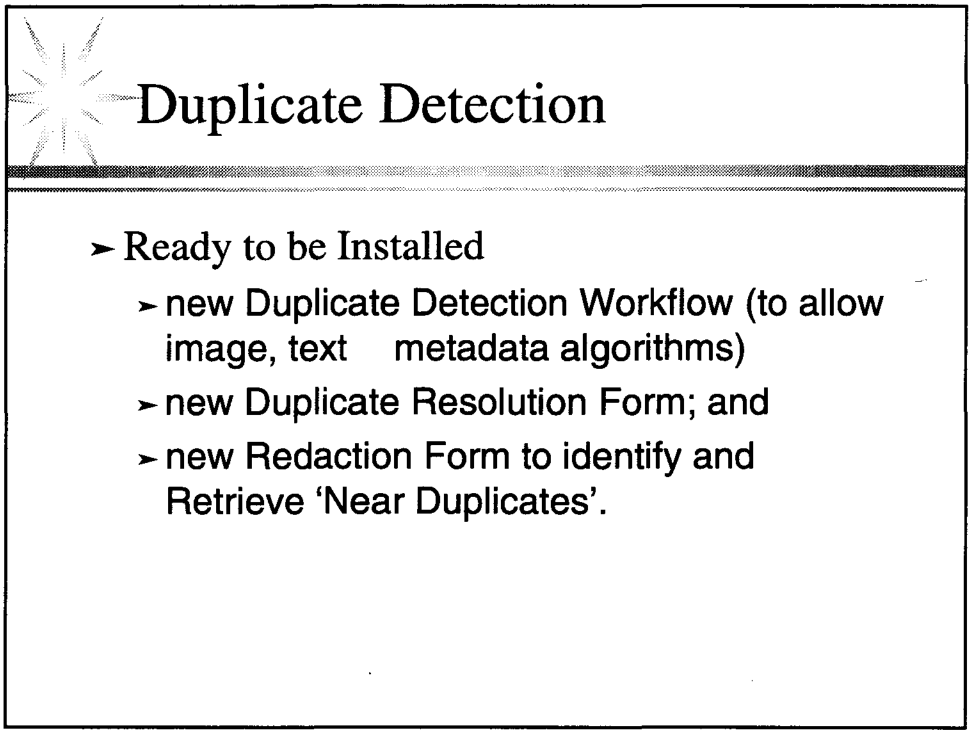
Automated Declassification System (ADS)

PLETHORA Duplicate Detection



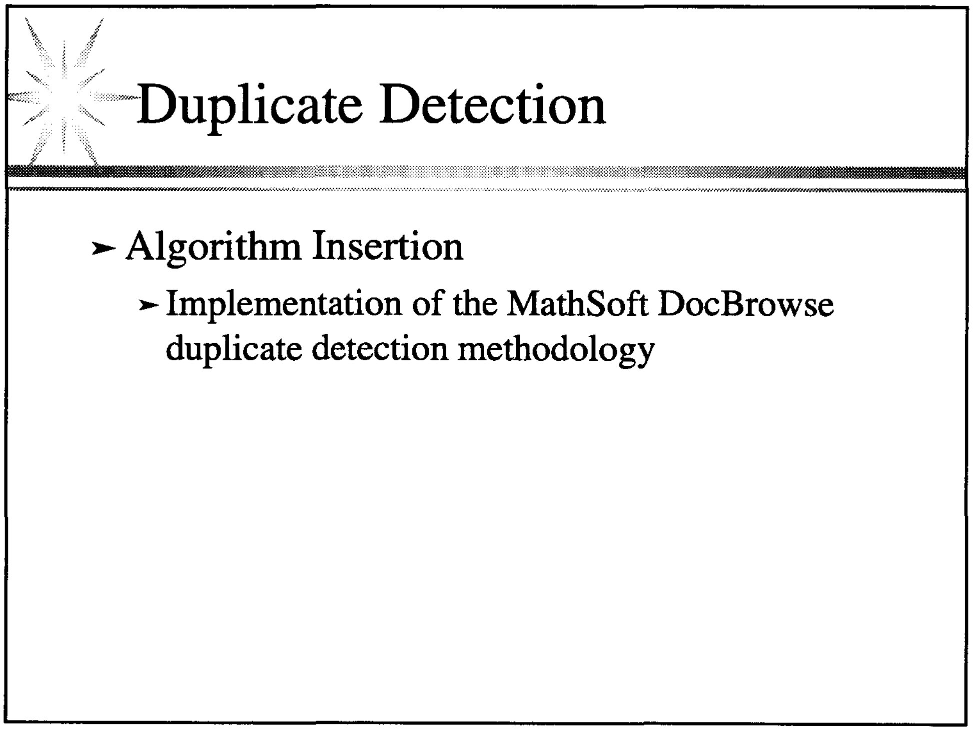
Duplicate Detection

- Current Methods
  - Exact Title
  - Exact Date
  - Manually resolve Potential Duplicates



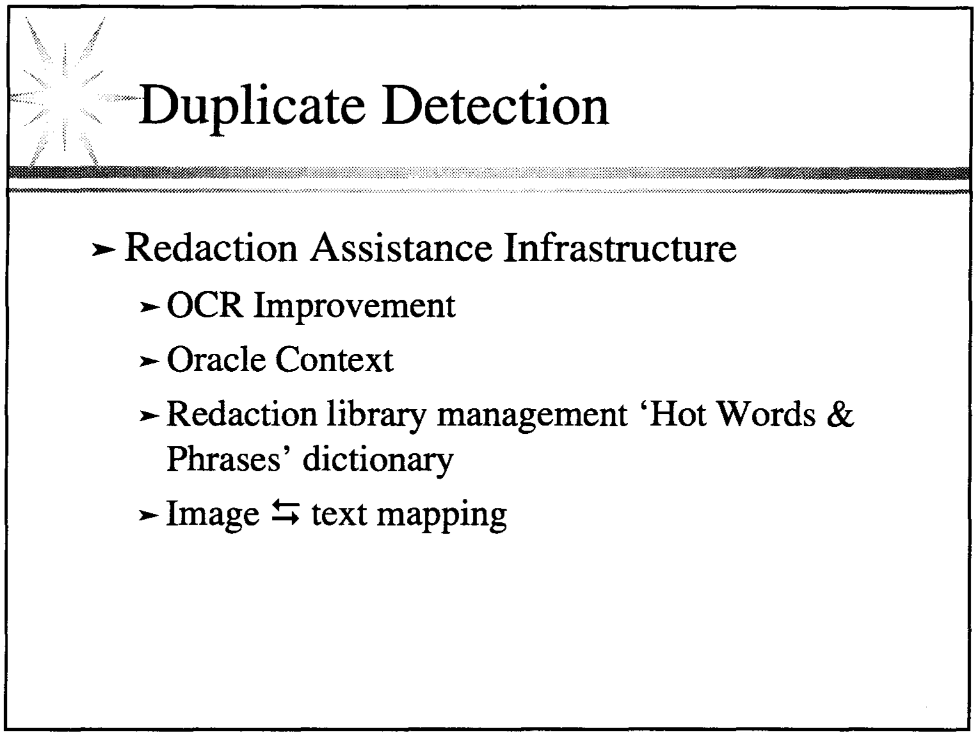
Duplicate Detection

- ➢ Ready to be Installed
  - ➢ new Duplicate Detection Workflow (to allow image, text metadata algorithms)
  - ➢ new Duplicate Resolution Form; and
  - ➢ new Redaction Form to identify and Retrieve ‘Near Duplicates’.



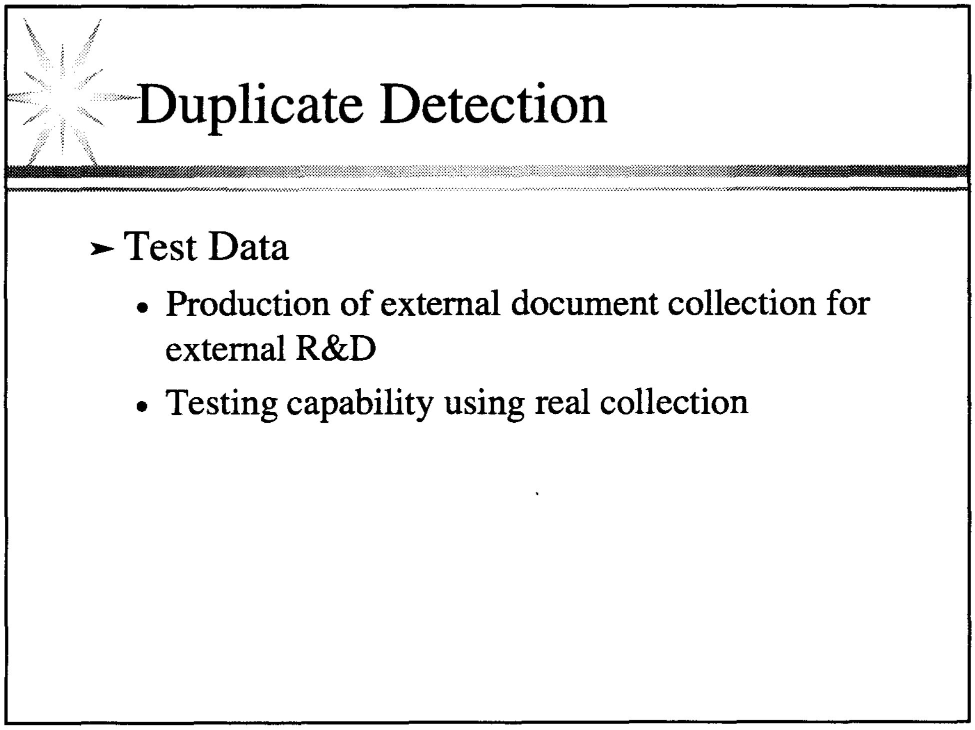
Duplicate Detection

- ➢ Algorithm Insertion
  - ➢ Implementation of the MathSoft DocBrowse duplicate detection methodology



Duplicate Detection

- ➢ Redaction Assistance Infrastructure
  - ➢ OCR Improvement
  - ➢ Oracle Context
  - ➢ Redaction library management ‘Hot Words & Phrases’ dictionary
  - ➢ Image ⇆ text mapping



Duplicate Detection

- ➢ Test Data
  - Production of external document collection for external R&D
  - Testing capability using real collection

Analysis and Design of Test Corpor for Zero-Tolerance Government Document Review Processes

By Richard S. Scotti and Carol Lilly

The George Washington University
Declassification Productivity research Center (DPRC)
Ashburn, VA 20147
Scotti@seas.gwu.edu

Abstract

Optical Character Recognition (OCR) and Duplicate Document Detection (D3) systems both present opportunities for improving the performance of government document review processes. OCR can help to bring digital computer power to declassification processes, but is of limited value for poor quality documents because of accuracy problems and associated high costs. Many of the documents in the US Government holdings, especially those produced 25 or more years ago, are of poor quality and not easily converted. On the other hand, avoidance of redundant reviews of duplicate documents could save time and money, as well as reduce security risks associated with the "mosaic effect." As new techniques, tools and commercial products in these areas are developed, it becomes necessary to benchmark and test them for their effectiveness, both against other competing tools, and against acceptable levels of performance. The DPRC is currently carrying out a study to identify the critical characteristics for OCR and D3 processes, and to assemble and disseminate test corpora in the declassification and research communities. Test Corpora should be statistically representative of the application domains, but, for practical reasons, should also be significantly smaller than the full inventories. Test Corpora development for the declassification arena is over shadowed by the zero tolerance nature of the process. This means that errors in declassification are deemed unacceptable, and must be avoided at all costs. The goal of this research is to facilitate testing and help standardize test results for meaningful comparisons, and to stimulate necessary research activities. The first product of this DPRC program, which will be available in the summer of 1999, is a 1400 page test corpus suitable for both OCR and D3 systems. It includes hard copy documents from the US Government holdings, document metadata, digital document images, and ground truth. In the case of D3 systems, current efforts are aimed at the page level, with duplicate detection based on analysis of document structure and metadata. The more general problem of duplicate detection by means of semantic analysis is beyond the scope of the present effort.

1. Introduction

Document declassification workloads in the Federal Agencies have increased significantly in recent years. Expanded use of the Freedom of Information Act (FOIA) by the Public, and the issuance of Executive Order (EO) 12958 on automatic release of federal documents by President Clinton in 1995 are symptoms of the new spirit of openness within the Government since the end of the Cold War. Several of the Agencies are trying to make up for the increased load through the development of more productive declassification processes involving new technologies. In this regard, areas of active interest include Optical Character Recognition (OCR) systems (for paper document digitization) and Duplicate Document Detection (D3) systems (for avoidance of redundant reviews). Digital documents can be manipulated, and in principle be more effectively reviewed than paper documents by persons using special computerized systems. Avoidance of redundant processing of documents promises savings. The purpose of this paper is to describe the concept for and the development of the DPRC Test Corpus for OCR and D3 systems, and to discuss its elements and features for test support.

The concept of a test corpus is based on the premise that it is possible to represent (predefined) identifiable and measurable characteristics of a very large set of documents in a much smaller, more readily manageable collection, if statistically sound procedures are followed. The validity of this concept hinges on:

1. Careful definition of the characteristics of the original collection to be represented in the test corpus
2. Understanding of the statistical properties of these characteristics within the original collection.

Test corpus development must therefore begin with an analysis of the functionality of the system to be tested, and of the relevant characteristics of the original collection that are to be represented in the corpus. For example, the abilit to *determine if two documents are duplicates of one another, based on analysis of the structure and metadata of the first page or cover sheet of each* (the functionality), depends on understanding the characteristics to be used for comparisons. We expect the size of the test corpus to increase, as the functionality and characteristics of the test corpus are increased to approach that of the original collection. The goal of test corpus design is to produce a specification for a subset of documents, including documents from the original set and other document samples as necessary, with statistical distributions of the relevant document characteristics that are similar to those within the original set.

The statistical distributions of document characteristics within US Government (USG) holdings that may affect OCR and D3 processes are unfortunately both complex and unknown. There are many reasons for this situation, including the fact that the holdings of interest were produced and assembled over many decades, during which time rapid technological developments and agency operational changes were the rule rather than the exception. Document generation, copying, distribution, management and storage processes all changed significantly during the period of our primary interest (i.e., Pre-1940 through the 1970's). Analysis and description of the statistical characteristics of the USG holdings is clearly a daunting task, *one that has not yet even been started*.

The zero-tolerance nature of the declassification process, however, suggests a simplifying perspective. Zero-tolerance means that errors in declassification are deemed unacceptable, and must be avoided. Simply put, the proposed perspective is to treat every possible event of importance as being uniformly probable, rather than statistically distributed. This approach emphasizes the inclusion of all expected difficulties, irrespective of their probability of occurrence. This is admittedly a conservative approach, because it subjects systems under test to situations or difficulties at a higher frequency of occurrence than might be expected with the original set of documents. Most importantly, it enables us to move forward with corpus development and system testing while accumulating statistical descriptions of the USG holdings over time.

The first product of the DPRC program, which will be available this summer, is a 1400 page corpus suitable for testing both OCR and D3 systems. In the case of D3 systems, our current efforts are aimed at the page level, with duplicate discrimination based on analysis of document structural and metadata. The more general and much more difficult problem of duplicate (or near-duplicate and similar document) detection by means of semantic analysis at the page segment, page, document, folder or series levels, is beyond the scope of the present effort.

The following sections present, in turn: an historical background along with some basic definitions, followed by discussions on the technical issues, the functionality and document characteristics targeted for the DPRC Test Corpus, a description of the DPRC Test Corpus itself, and finally a list of future research objectives.

2. Historical Background and Technical Issues

Before attempting to identify the features of documents and of duplicate documents that may be of use for corpus characterization, it would be helpful to first consider document technologies and document duplication practices over the years from the 1940's to the 1970's.

2.1. Document Technology

Developments in documentation and in document production, handling and storage technologies in the years since WWII, especially since the advent of electric and (then) electronic office equipment, have been revolutionary. Technology advances have also driven the marketplace. The number of commercial document-related products has also increased greatly. The simple typewriters and few common fonts available in the late 1940's and early 1950's expanded to many different font styles, and more sophisticated typewriters by the mid 1970's. By the 1980's there were available different styles of computer-based word processors, text and graphics printers, copiers, databases, etc. Paper and other document materials technologies also sprang up during the sam period and evolved rapidly. Each technology advance, each new commercial product, each new process and document management approach introduced a new set of conditions and improvements in document quality.

2.2. Physical Characteristics of Documents and Printed Text

The documents currently held by the Agencies that are subject for review under EO 12958 were created using the broad range of technologies, office tools, equipment and materials alluded to just above. Many of these documents are faded and deteriorated, and include marginalia, various stamps, and hand drawn charts and graphs. These (common) features present a range of document characteristics with the potential to adversely affect the OCR process. The key questions here are:

1. What physical characteristics (PC's) of documents (or printed text) affect OCR performance and drive the achievable levels of OCR accuracy?
2. Would it be possible use these PC's to define a set of Quality Measures that describe documents (and document collections) in regard to expected OCR performance
3. Would it be possible to use these same Quality Measures to support the design and assemble of a test corpus?

If we consider the functions that are necessary to convert a page of text to an editable-text file, several physical characteristics of the page can be identified which are likely to affect the process. Paper documents are first scanned into image or pixel maps at one or more data densities. The tone and textural contrasts of the document undoubtedly effect the scanning process. Both tone and texture affect contrast, and are themselves dependent upon the basic features of documents: (1) paper-ink color combinations, (2) paper quality, and (3) document aging (browning, drying, cracking, fraying, etc.). These factors may be reduced (or provisionally eliminated) if document images are converted to black/white images at a controlled contract level prior to further processing. The highest qualit commercial text conversion systems typically include an image enhancement stage after scanning to produce a clearer representation, with stronger contrast between text characters and the background, prior to the OCR process.

The lines and individual text characters in older documents are often skewed, as a result of inaccuracies in the original production equipment, or because of distortions introduced by paper deterioration or poor handling during document copying operations. The following page layouts and formats, commonly encountered in the USG documents of interest to us here, can also affect the OCR process:

- Single and multiple column text
- Text plus graphics and/or tables
- All of the above plus marginalia

The presence of marginalia can create more difficulties and more OCR errors, either because of difficulties identifying handwritten characters, or because the marginalia partially over lays or distorts the text which is being processed.

Recent research carried out by Cannon et. al [1] and reported in another paper at this meeting offers affirmative answers to Question # 1 (above) for the case *of fixed width font, typewritten documents.* This work demonstrates the importance of the following five pixel-level "Quality Measures" (defined and explained in the reference) as page characteristics in regard to ORC process error and/or accuracy rates:

(1) Small speckle factor
(2) White speckle factor
(3) Touching characters factor
(4) Broken characters factor
(5) Font size factor

Cannon has been able to show how these Quality Measures (can) reflect achievable performance of commercial grade OCR systems, and has also been able to use these same factors to select text modifications for significant improvements of the quality of deteriorated typewritten text and associated OCR accuracy.

The implications of this research are clear: ***It is apparently possible to use the five Cannon Quality Measures to characterize the text images in regard to achievable OCR accuracy***. As noted however, this research is currently limited to the case of fixed width font, typewritten text.

2.3. Document Duplication Practices

During this same period (from the 1940's through the 1970's), policies and practices for duplication and distribution of documents within the Agencies of the US Government also underwent significant changes. As the need for information dissemination grew, and as new communication and duplication technologies and equipment became available, it became easier to produce, and make use of duplicate copies of important documents. Carbon paper and Onion skin duplication processes, for example, were displaced by Mimeograph and Thermofax copy machines. These too were eventually replaced by Photo and Xerox copy machines. These changes are reflected in the broad variety of documents, and document media types within the USG holdings. There is also evidence [2] that many of the documents in the USG holdings are duplicates or near-duplicates of one another.

An understanding of the types and numbers of duplicate documents to be expected within an Agency can be supported by knowledge of the Agency's policies and "intentional acts". Intentional acts include the activities within an Agency by means of which duplicates were produced from parent documents and retained in the document inventories. The complete set of intentional acts is an important aspect of the "institutional knowledge-base," which can be used to identify the characteristics of expected duplicate documents. The intentional acts reflect the institutional habits of the Agency. Humans and organizations are habitual by nature. Though new staff members often do not carry out the same intentional acts as seasoned members, individual acts tend to produce regular events. Regular events will tend to produce patterns in the types and numbers of duplicates (or modifications) to be found within an Agency's document holdings.

2.4. Basic Definitions

An important distinction must be made between a copy of a document, even a poor, low resolution, scaled or skewed copy, and an intentional modification of a document, whether made by a human or by a machine acting as a proxy to a human. A critical issue here, as alluded to above, is identification of the set of regular activities by which duplicates or near-duplicates may be produced. Our further developments are based on the following definitions:

> ***Duplicate Documents***: Two documents are "***duplicates***" of one another if they are exactly the same, except that one may be a copy, or both copies of some (perhaps unknown) original, with no intentional alteration(s) to the document in regard to structure, content or meaning. Copies may include exact copies in the same or different media, size reductions, rotations and other modifications that do not affect the structure or content of the original document.
>
> ***Near-Duplicate Documents***: Two documents are "***near-duplicates***" of one another if they satisfy one of the following two cases:
>
> 1. One of the documents has been produced from the other by some specific intentional alteration(s) to the document in regard to structure, content or meaning. Examples of intentional alterations include: addition of a signature, marginalia, or other markings, which may distort the structure of a document, overlay its content and/or distort its meaning.
> 2. Both documents have been produced by an intentional alteration from a common, perhaps unknown, parent.

3. Test Corpus Design

A corpus of documents is said to be "representative" of the original collection if it has the same (or reasonabl similar) statistical distributions, *in regard to the activities to be tested.* The goal of the present effort was to develop a set of specifications for the types and numbers of sample documents to be contained in the DPRC Test Corpus to ensure its *Representativeness* for both OCR and D3 systems testing. As discussed above, this feature of a corpus depend on the specific functionality of the corpus in regard to the systems to be tested.

3.1 With Respect to OCR Processes

The OCR functionality targeted by the DPRC Test Corpus include all process steps necessary to convert a scanned image of a document to an ASCII text file. Beginning with the observation that the documents of interest were produced when relatively few optional technologies and commercial office tools were available, we adopted the following strategy for selection of a representative set of documents for the corpus:

Select documents that were produced by the same (or similar) equipment and sequences of operations during each of the relevant decades (the 40's 50's, 60's and 70's) over the range of the document characteristics of probable importance to OCR processes, including document type/format, media, visual complexity, visual quality, etc.

The matrix of document characteristics we sought to include in the documents we collected for the DPRC Test Corpus is shown in Table 1. Working from this matrix, we collected documents from the unclassified holdings at the National Archives and Records Administration (NARA), and at the public reading rooms at the DOE Office of Historical Documents, and the Museum of Military History at Fort McNair. Taken as a whole, the characteristics and combinations of characteristics represented in Table 1 represent a very large set of circumstances. While completeness requires the construction of all feasible combinations, our strategy for document selections tended to be more subjective and was based on the variety of documents and combinations of characteristic we encountered working with ma document collections in several different agencies.

Table 1. Document Characteristics Matrix for OCR Test Corpus

Type of Document	Type of Material	Type of Copy
Memo	Plain paper	Original
Letter	Colored paper	Carbon
Report	Carbon paper	Photo Copy
Telefax/Telecon	Onion skin	Page/News print
Newspaper article	Colored onion paper	Teletype
Item for the Record	Telefax/Telecon paper	Mimeograph
	Mimeograph paper	

Visual Complexity	Special Features	Originating Agency
Very Complex	Marginalia	Std agency designators
Moderatel	Stamps	
Medium	Figures	
Low complexity	Tables	
Simple	Combinations	

This approach, while rational, lacks any quantitative means for validation of Representativeness. We have therefore identified and are in process of introducing an additional tool to provide quantitative evidence to support the selections for and the document characteristics represented in the DPRC Test Corpus. This development follows, specifically for the case of OCR processes, the perspectives described in Section 2.2. We began with the following implication from Section 2.2:

Pages of text with the same (or similar) values of Cannon's five Quality Measures will result in the same (or similar) OCR process accuracy, using high quality commercial OCR systems

The range of values of the Cannon Quality Measures and their frequencies of occurrence for documents within the USG holdings are presently unknown. However, the above statement, which is well supported by Cannon's experimental results, points to the prospect of using the five Quality Measures, in the form of ***a five dimension Physical Characteristics Vector*** **(PCV)**, to define and characterize *Representativeness* of an OCR Test Corpus.

The basic concept for making use of the **PCV** is explained as follows. A document collection can be described in terms of its Cannon Quality Measures, and then quantitatively represented by its **PCV** mapping using an information visualization tool. There are several tools available for this task, including "Parentage", developed by Dr. Jonathan Cohen of DoD [3], or "Spires" and "Galaxies" [4], developed at the Pacific Northwest Laboratory (PNL). These tools provide the capability to visualize associations or relationships among elements of a data collection. In the present case, the relationships of interest are those among the pages of a corpus in regard to their (five) Quality Measures. Mapping the **PCV's** for a test corpus provides an effective means to compare the documents in the corpus to one another based on all five Quality Measures. Mapping is a step towards understanding the implications of the Quality Measures and towards developing a global description of the entire collection. The **PCV** mapping for the USG holdings could in principal provides the means for its characterization and management.

This concept is illustrated in Figure 1, which shows graphically our use of data visualization tools for corpus characterization. The objects in Figure 1 represent different documents within a collection. The connecting lines (indicative of the separation between objects) show similarity of the Quality Measures for each of the documents as follows:

Small separation indicates similar Quality Measures.
Large separation indicates dissimilar Quality Measures.

3.2 With Respect to D3 Processes

The D3 functionality targeted by the DPRC Test Corpus is described by the following (sample) process, which is intended for duplicate discrimination among single page documents, including cover pages of multiple page documents. The (sample) D3 process is described here as a series of activities or steps beginning at the document check-in or log-in phase of review, possibly before scanning or any time/labor intensive investments are made.

First: Manuall detect and log (predefined) characteristics of a document into a metadata database.
Second: Access those same characteristics for documents that have been previously received and processed
Third: Compare characteristics for documents under investigation with previously investigated documents
Fourth: Reason as to whether these document characteristics support classification of pairs of the documents as a duplicate or near-duplicate of one or more previously reviewed documents

Fifth: Assemble a list of potential class relationships among the documents under study and submit to (human) declassifiers for final decision and further processing.

Sixth: Submit high probability matches to visual (human) inspection, using on-screen electronic data, or associated hard copy of the suspected previous document(s).

Seventh: Annotate the metadata database and direct the document(s) for further processing, as necessary.

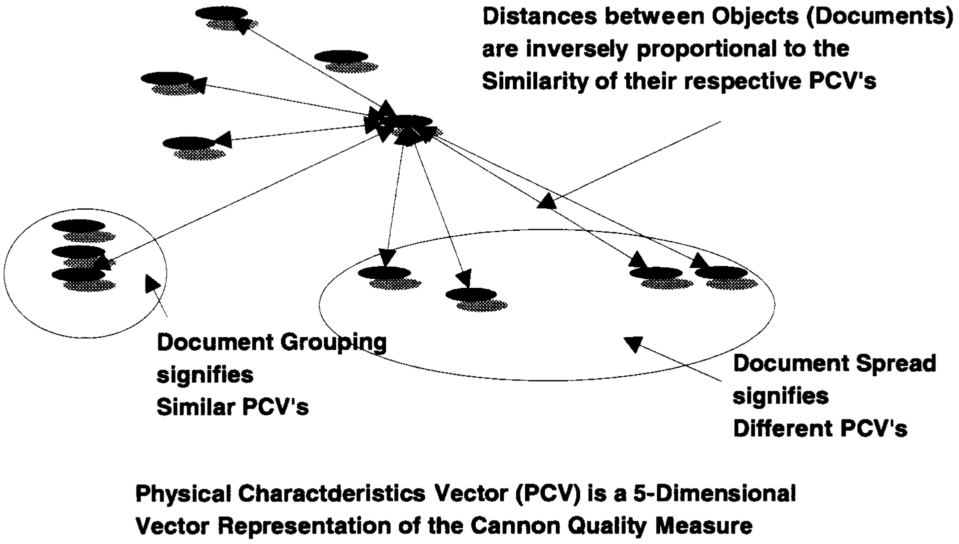

Distances between Objects (Documents) are inversely proportional to the Similarity of their respective PCV's
Document Grouping signifies Similar PCV's
Document Spread signifies Different PCV's
Physical Charactderistics Vector (PCV) is a 5-Dimensional Vector Representation of the Cannon Quality Measure

Figure 1. Use of Data Visualization to Display Quality Measures for a Set of Documents

We were able to identify a set of document characteristics for use in the (above) D3 process by means of structured interviews with the experienced government document reviewers. The characteristics, which are elements of the document metadata, are shown in Table 2. There are two different types of factors: Primary and Secondary. The four Primary Factors are those that are felt to be the easiest to identify and to apply for base document discrimination, if they exist. These include (1) Original Classification, (2) Originating Organization, (3) Creation Date, and (4) Document Number. The three Secondary Factors are typically more difficult to identify, or are either ambiguous or often absent on government documents. These include (1) Recipient (Person and/or Agency), (2) Title, and (3) Number of pages. Also shown in Table 2 are some of the combinations (Cases) of these document characteristics (factors) that support (positive) identification and highly probable duplicate or near-duplicate relationships among documents. More research is necessary to verify the utility of these combinations and to develop the logic for implementing metadata-based D3 processing.

3.3. Statistical Considerations

Two separate but related statistical concerns are worthy of further discussion in regard to assembly and validity of the DPRC Test Corpora:

(1) The frequency of occurrence within the DPRC Test Corpus of each of physical characteristics and combinations of the physical characteristics of documents which affect OCR error rate and/or duplicate discrimination.

(2) The size of the DPRC Test Corpus (page count) to provide a large enough sample for a statisticall significant test of OCR and/or D3 processing systems.

The physical characteristics for the OCR and D3 processes are shown in Table 1 and 2, respectively. The number of characteristics is indeed very large. The number of feasible combinations of these characteristics is still much larger. A statistical analysis based on such a large number of parameters is completely impractical, especially given the current lack of understanding of the USG document collections. As previously discussed, our approach to this dilemma has been to attempt to create a ***Representative*** collection of documents by following "the Arrow of Time" as an organizer of the evolution of equipment and processes within the USG. It is our hope that our document selections following this strategy span the full range of the many document characteristics, and also maintain proportions that are natural to the entire collection.

The second, and perhaps most questionable, aspect of our approach to statistical modeling is that of deferring questions on the statistics of the document collections to issues related to the statistics of the systems being tested. We propose to use an OCR/D3 process-based statistical model, rather than a statistical model of the USG document inventory. Development of an OC/D3 process-based model is a much simpler problem, because its design is based on the ***statistics of the process being tested*** or evaluated rather than the ***statistics of the USG's inventories***. Following this approach, the second concern, the number of pages for statistical significance, is driven by the *Variance* of the process being tested. We note that in a series of experiments that preceded the present DPRC effort, Nartaker [5] experienced rather large variances during his systematic tests of four commercial OCR systems, and a UNLV OCR system. Nevertheless, the "acceptable" level of experimental error determines the number of samples necessary for statistically significant testing. Knowledge of the process *Sample Variance* is sufficient for specification of sample size in order to bound experimental error. Mathematically, the relationship between Process *Sample Variance* and sample size is given by:

$N_s = \{(1.96\ S_s)/\text{ Delta Error Rate}\}^2$, where

- N_s is the necessary sample size, or for the present case, the number of documents of any particular type
- S_s is the sample Standard Deviation
- Delta Error Rate is the "acceptable" level of deviation between the expected error rate and that observed

Our approach, therefore, proposes that the DPRC Test Corpus will be adequate if it has

(1) ***Sufficient Breath.*** That is, if it is made up of documents that contain the full range of characteristics found in the USG inventory. A broad coverage is most desirable.

(2) ***Sufficient depth.*** That is, if it contains numbers of documents to ensure bounded error during testing of the processing capabilities of the OCR and/or D3 system being evaluated.

Table 2. Matrix of Metadata Combinations for Duplicate Document Discrimination

		Primary Factors				Secondary Factors		
Case Number	Number of Factors	Original Classif.	Originating Agency	Creation Date	Document Number	Recepient/ Agency	Document Title	Number of pages
1	4 + 1	x	x	x	x		x	
2	4	x	x	x	x			
3	3 + 1		x	x	x		x	
4	3 + 1	x	x	x			x	
5	3 + 1	x	x		x		x	
6	3 + 1	x		x	x		x	
7	3	x	x		x			
8	3	x		x	x		x	
9	3 + 1		x	x	x		x	
10	3 + 1	x	x		x			x
11	3 + 1	x	x	x			x	
12	3 + 1	x	x		x		x	
13	3 + 1		x	x	x		x	
14	3 + 1	x	x	x		x		
15	3 + 1	x	x	x				x
16	3 + 1	x		x	x		x	
17	2 + 1		x		x			x
18	2 + 1		x		x		x	
19	2 + 1			x	x			x
20	2 + 1			x	x		x	
21	1 + 2			x			x	x
22	1 + 2			x		x	x	
23	1 + 2				x		x	x
24	1 + 2				x	x	x	
25	1 + 2	x					x	x
26	1 + 2	x				x	x	

4. Description of the DRPC Test Corpus

The DPRC Test Corpus presently consists of 1400 pages that were judiciously selected from the USG unclassified inventories at NARA, DOE and the Military Library of History. These documents originated from a large number of agencies including DOE, DOS, US Army, DOD, War Department, AEC, and the CIA. As discussed above, the documents were systematically selected over the historical period of interest (pre 1940's through the 1970's) to include the full range of document characteristics believed to be of importance in regard to both OCR and D3 processes. The sample documents contained in the corpus, by implication, span the range of the parameters listed in Tables 1 and 2. The corpus has been hosted in an MS Access database to facilitate management of its man features. Figure 2 shows the Screen of the DPRC Test Corpus MS Access database. The following data elements are visible in Figure 2.

Document ID (Document Identification Number): A unique number is attached to each sample in the corpus.

General Characteristics (Meta-features of sample):

Originating Agency
Year Created
Type of document
Type of paper
Type of copy (process)
Visual complexity
Special features
Creating equipment

Document Image (*a Button*): A scanned TIFF image made from the original document in the USG inventory

Ground Truth (*a Button*): An ASCII text file with 100% accurate text (characters and spaces) manually generated and verified from the original sample document image

Duplicate Document Detection factors: Metadata to support discrimination of duplicates or near-duplicates Based on metedata comparisions:

Originating Agency
Creation date
Recipient person/organization
Document number
Original classification
Title
Carbon Copy number
Number of pages

Image Quality Measures and OCR Accuracy Rate: Metadata to support OCR characterization and processing:

Small Speckle factor
White Speckle factor
Touching Character factor
Broken Character factor
Font Size factor
OCR Accuracy

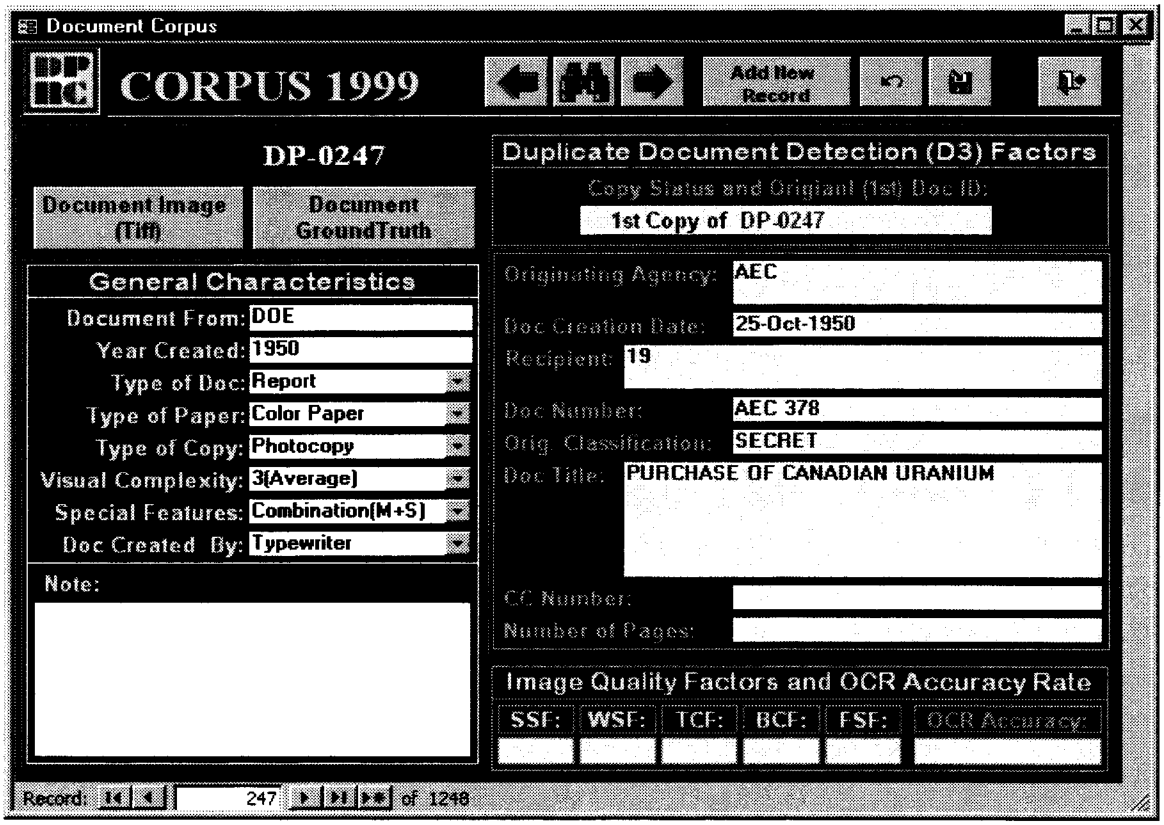
Document Image and Ground Truth **buttons** open windows to display the associated images. See Figures 3. The database functionality provided by MS Access allows a user to browse the sample documents/records in a variety of modes, to sort and list responses to database queries, and to generate reports on the information within the database.

5. Summary and Future Research Objectives

This report has described the on-going program at the DPRC to develop and disseminate a test corpus suitable for both OCR and D3 processes. There is general agreement that having such a test corpus will represent a valuable step forward for the community, for a variety of reasons. A test corpus could help to better define the real problems that currently limit declassification performance. A test corpus could also help to better define technology gaps and research needs for the immediate and longer-term future. The activities and accomplishments during the first phase of the effort have been undertaken with a view that they will feed into a series of follow-on phases listed below:

1. Develop **PCV** metrics based on the Cannon Quality Measures to enable effective use of data visualization tools for characterization and management of document information, as discussed above.
2. Develop standard test procedures and evaluation criteria for both OCR and D3 systems.
3. Apply the DPRC Test Corpus and standard test procedures to evaluate the current premises and assumptions represented in the corpus.
4. Develop a DPRC Test Corpus CD ROM system for dissemination in the declasssification and research communities.

Finally, this work was supported by a continuing grant from the DOE Office of Declassification, under the technical direction of Tom Curtis, and by a contract relationship with the Office of Information Management, under the technical direction of Harry Cooper.



DP-0247

Duplicate Document Detection (D3) Factors

Copy Status and Origianl (1st) Doc ID: 1st Copy of DP-0247

Originating Agency: AEC

Doc Creation Date: 25-Oct-1950

Recipient: 19

Doc Number: AEC 378

Orig. Classification: SECRET

Doc Title: PURCHASE OF CANADIAN URANIUM

CC Number:

Number of Pages:

General Characteristics

Document From: DOE

Year Created: 1950

Type of Doc: Report

Type of Paper: Color Paper

Type of Copy: Photocopy

Visual Complexity: 3(Average)

Special Features: Combination(M+S)

Doc Created By: Typewriter

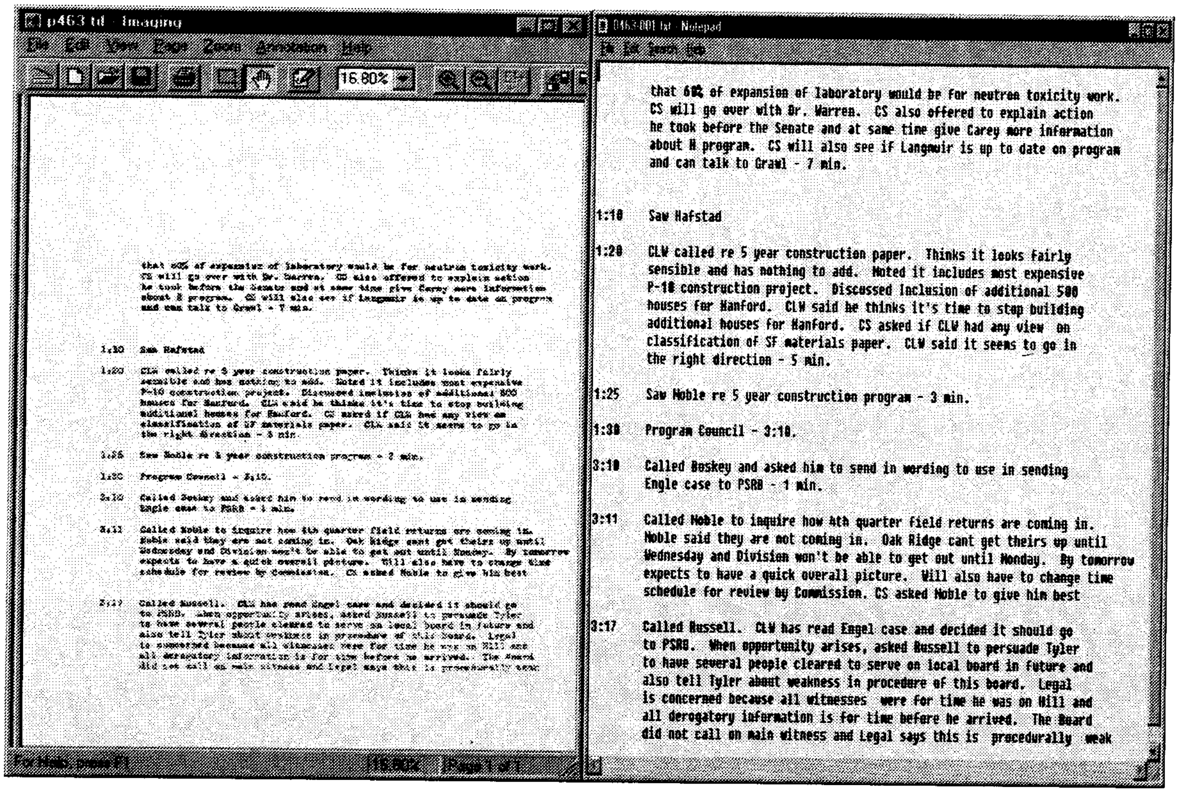
Note:

Image Quality Factors and OCR Accuracy Rate

SSF:	WSF:	TCF:	BCF:	FSF:	OCR Accuracy:

Record: 247 of 1248

Figure 2. The DPRC Test Corpus Database Screen



```
        that 60% of expansion of laboratory would be for neutron toxicity work.
        CS will go over with Dr. Warren.  CS also offered to explain action
        he took before the Senate and at same time give Carey more information
        about H program.  CS will also see if Langmuir is up to date on program
        and can talk to Grawl - 7 min.


1:10    Saw Hafstad

1:20    CLW called re 5 year construction paper.  Thinks it looks fairly
        sensible and has nothing to add.  Noted it includes most expensive
        P-10 construction project.  Discussed Inclusion of additional 500
        houses for Hanford.  CLW said he thinks it's time to stop building
        additional houses for Hanford.  CS asked if CLW had any view  on
        classification of SF materials paper.  CLW said it seems to go in
        the right direction - 5 min.

1:25    Saw Noble re 5 year construction program - 3 min.

1:30    Program Council - 3:10.

3:10    Called Boskey and asked him to send in wording to use in sending
        Engle case to PSRB - 1 min.

3:11    Called Noble to inquire how 4th quarter field returns are coming in.
        Noble said they are not coming in.  Oak Ridge cant get theirs up until
        Wednesday and Division won't be able to get out until Monday.  By tomorrow
        expects to have a quick overall picture.  Will also have to change time
        schedule for review by Commission. CS asked Noble to give him best

3:17    Called Russell.  CLW has read Engel case and decided it should go
        to PSRB.  When opportunity arises, asked Russell to persuade Tyler
        to have several people cleared to serve on local board in future and
        also tell Tyler about weakness in procedure of this board.  Legal
        is concerned because all witnesses  were for time he was on Hill and
        all derogatory information is for time before he arrived.  The Board
        did not call on main witness and Legal says this is  procedurally  weak
```

Figure 3. The Pop-up Windows for Document Image (Left pannel) and Ground Truth (Right pannel)

References

1. Michael Cannon, et. al, "An Automated System for Numerically Rating Document Image Quality," Proceedings 1997 Symposium on Document Image Understanding Technology, Annapolis, MD p162.
2. Richard Scotti, "Technology Programs at the GWU Declassification Productivity Research Center," Proceedings of the 4th Annual Intelligence Community Information and Classification Management Conference, October 1998, Sterling, VA, p249
3. Jonathan Cohen, private communication in regard to "Parentage," a proprietary data visualization code.
4. Nancy Miller, et.al, "The Need For Metrics In Visual Information Analysis," Workshop on New Paradigms in Information Visualization and Manipulation, Sixth ACM International Conference on Information and Knowledge Management (CIKM '97), November 13-14, 1997, Las Vegas Nevada
5. Thomas Nartker, et. Al, "OCR Accuracy of the LSU Data Set," Information Science Research Institute, University of Nevada, Las Vegas; and Louisiana State University, February 1997

The Voice Operating GUidancE System (VOGUE)

By Richard S. Scotti, Youngsuck Oh and Carol Lilly

The George Washington University
Declassification Productivity research Center (DPRC)
Ashburn, VA 20147
Scotti@seas.gwu.edu

Abstract

Classification and declassification document reviews usually involve comparing information within documents to "sensitive" descriptions or combinations of information represented in formal Guides. In conducting a document review, an analyst must first recognize potentially sensitive topics, and then locate references in the Guides before appropriate actions can be determined. In the simplest cases the guidance is contained in easily identified and readily accessible sections of these Guides in the form of topic characteristics or rules. These rules are then applied to the information in the pages being reviewed. In more complex cases, sensitive situations are more difficult to describe and the analysts must interpret guidance, often from several documents, before it can be applied. This research concerns an investigation of the effectiveness of a voice-activated human-computer interface (HCI) to support analysts during declassification document reviews. This paper will report on the design, development and test of a prototype system to demonstrate a concept of operations for a voice-activated guidance system. Hardware and software computer systems have been assembled and tested to demonstrate the following specific reviewer capabilities:

1) Ability to simultaneously see documents images and relevant declassification Guides on the same screen
2) Ability to simultaneously use voice, and/or mouse and keyboard to operate the syste
3) Ability to search multiple declassification Guides based on sensitive words in the documents
4) Ability to apply Guide search results during a document revie

Using VOGUE, reviewers are able to conduct declassification reviews from a simple, multi-window screen that displays document images, and declassification Guides, as well as review results.

1. Background

Classification and declassification document reviews involve the process of comparing information within documents to "sensitive descriptions or combinations of information represented in formal guides. In conducting a document review, an analyst must first recognize potentially sensitive topics, and then locate references in the guides before appropriate actions can be determined. In the simplest cases the guidance is contained in easily identified and readily accessible sections of these guides in the form of topic characteristics or rules. These rules are then applied to the information in the pages being reviewed. In more complex cases, sensitive situations are more difficult to describe and the analysts must interpret guidance before it can be applied.

A number of research activities are currently investigating the possibility of providing computer support to analysts when the document being reviewed is available in machine readable, editable text form; i.e., digital ASCII characters. Alternatively, for older and degenerated documents, generation of digital representations is often prohibitively time consuming and expensive. In this case, analysts must work with images of the pages of the documents rather than digital, editable text representations. The purpose of the present research was to investigate the effectiveness of a voice-activated user-machine interface to support analysts during this type of declassification document reviews.

2. Scope

The research entailed the design, development and test of a prototype system to demonstrate a concept of operations for a voice-activated guidance system. It, therefore, involved both hardware and software computer systems, and the following specific project activities:

- Development of a concept of operations for a voice activated guidance syste
- Identification of system architecture, and component hardware and software elements
- Development of a prototype voice activated guidance syste
- Analysis of the prototype results to identify the potential utility of such a system as well as feasible enhancements and research activities for further development.

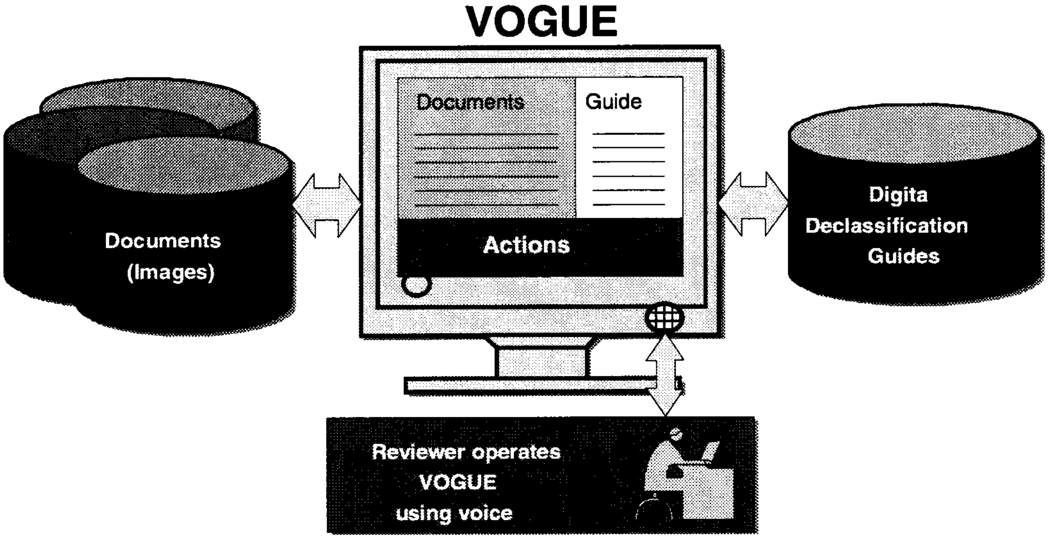
3. Development Strategy

The development strategy focused on following tasks:

1) define the VOGUE system concept based on document review process
2) identify voice recognition and text indexing and search software
3) select necessary hardware and software for the syste
4) assemble and integrate prototype systems to see the VOGUE functionalit
5) demonstrate simple VOGUE syste

To accomplish these tasks we concentrated on the use of available Commercial Off The Shelf (COTS) software to the maximum extend. The aim was to gear the functionality of the VOGUE system to support enhanced reviewer performance within currently available system resources.

4. VOGUE Concept Diagram



The concept of operations for VOGUE illustrated in the diagram above is based on the following activities:

1) Document images from a Document (images) Database are presented to the operator for review in the left-hand panel of the computer screen. The operator can select the specific document(s) for review from the inventory in the document (images) database. The operator can scroll through and read the document image, redact sensitive passages from the image of any page using a graphics tool and/or add notes anywhere on the page using a note pad system
2) Guide documents are presented to the operator in the right hand panel of the screen. The operator can select the specific guide to be used from those available in the Guides Database. The operator can scroll through a guide that has been opened, or use a search engine to identify relevant sections of the guide for consideration. The search engine can be operated manually using the cursor and keyboard, or automatically using the VOGUE voice command mode. In either case, the operator has the capability to view the guide sections that are relevant according to the system and select one (or more) section(s) that relate to the material being reviewed for purposes of redaction.
3) The operator can record activities carried out during the review as well as the results of the revie in an Actions record. The fields of the Action record can be loaded into the Document record as metadata and then returned with the document to the Documents (images) Database.
4) VOGUE enables the operator to use voice commands either in conjunction with or in place of the mouse and keyboard for (almost) all activities related to: document selection and presentation, guide selection and search, document review, document redaction, document/page annotation, review action recording, updating document metedata and saving and returning documents to storage

5. Functional Requirements

The functional requirements for VOGUE are based on the requirements of the reviewer to process classified documents. Reviewers typically do the following tasks in their declassification review process:

1) read classified documents
2) determine sensitive information
3) identify relevant declassification guides
4) perform pre-determined actions based on review results

The following capabilities were identified as functional requirements for the prototype VOGUE system. The prospective users of VOGUE should establish final functional requirements in a subsequent development stage of the VOGUE development project.

5.1. Graphical User Interface (GUI): The VOGUE system should be operated under Windows NT. Most of government agencies are using Windows NT as main computing environment.

5.1.1. View documents: Documents are stored as scanned TIFF images. The system should be able to display TIFF images on the screen.

5.1.2. View declassification guides: Declassification guides are in various formats. (e.g. HTML, MS Word Documents, and etc) The system should be able to display various guide formats on the screen.

5.1.3. Operate and navigate the system using combination of voice, mouse and keyboard: Users should be able to operate VOGUE system using not just voice but normal computer operation tools such as mouse and keyboard.

5.2. Database: Database should provide a storage facility for document images, declassification guides, and review information. It should also be able to communicate with the VOGUE syste and support the basic functionality of VOGUE.

5.2.1. Store document images: The VOGUE system should be able to access document images.

5.2.2. Store declassification guides: The VOGUE system should be able to access declassification guides.

5.2.3. Store declassification review information: Reviewers should be able to store review data and any other relevant information.

5.3. Declassification Guides Searching: The system should be able to index available declassification guides and produce search results, based on criteria specified by the reviewer.

5.3.1. List available guides: The system should be able to list all of the available declassification guides.

5.3.2. Search against sensitive words in declassification guides: The system should be able to provide at least key word search capability. (More sophistocated search capabilities ma be sought in the future)

5.3.3. Show summary of search results: The system should provide condensed version of search results. It should also be able to show any part of the declassification guides that satisf the search criteria.

5.3.4. Show full text search results: The system should return the full content of any portion of the declassification guides that satisfy the search criteria.

5.4. Document Image Editing: The system should provide the capability to manipulate document images. It should be able to view and edit TIFF images.

5.4.1. List document images to be reviewed: The system should be able to list and select document images.

5.4.2. Display document images: The system should be able to display TIFF images on the screen.

5.4.3. Provide image edit capability: The system should provide editing capability in support of review actions.

5.4.3.1. Zoom In/Out: The system should be able to navigate images.

5.4.3.2. Marking: The system should be able to mark sensitive information on the image.

5.4.3.3. Stamping: The system should provide a capability to stamp pre-determined information on the document image under review.

6. Software/Hardware Details

The following hardware and software components were selected, based primarily on capabilities and price, from those available off-the-shelf. In most cases preference was given to low cost or no cost components. For example, shareware or free software components were selected over those that might provide greater capabilities. Our rationale was based on the prototype nature of this project and on budget limitations. Selected components were integrated together to achieve the above functional capabilities:

6.2 Hardware components

Item	Testing System	Notes
CPU	PII 333 MHz	• Minimum CPU requirement is Pentium class with MMX technology, but faster CPU (at least 266MHz Pentium II) is preferable
RA	128 MB SDRAM	To run NT server with Web Server and Index server, 128MB is a minimum requirement
Hard Drive	EIDE 4 GB	• Storage capacity depends on number of document images and declassification guides • Storage for Index server depends on number of documents and size of corpus
Sound	SoundBlaster 64 Gold	• Dragon system provides compatible and recommended H/W Lists • http://www.dragonsys.co
Microphone	Came with Dragon Dictate	• Same as above
Monitor	17 inch with 1280 x 1024 at 75Hz	• Reviewers prefer 21 inch monitor with 1280 x 1024 at higher refresh rate

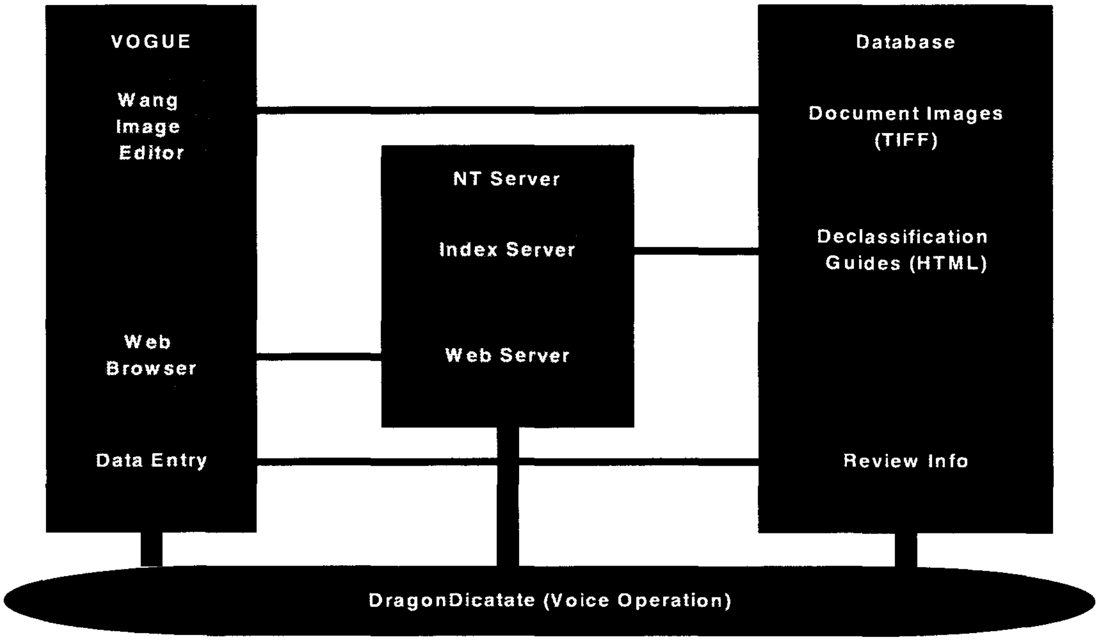
6.3 Software components

Software	Product	Notes
OS	Windows NT4 Server or Windows NT4 Workstation	• NT is a standard OS in intelligent communit • Option Pack contains MS Index Server 2.0 and other necessary web components
Voice Operation	DragonDictate 3.0	• Discrete speech • Hand-free voice operation
Development Tools	Visual Basic 5.0 DragonDictate Macro Language	• The VOGUE syste developed by using Visual Basic
Imaging	(Wang) Imaging for Win N	• Built into N • Image editing capabilit
Database	Access 97	• Store Image Document, Declassification guides, and Review information
Web Server	Internet Information Server (IIS) for NT Server or Peer Web Server(PWS) for NT Workstation	• Required to use MS Index server

Web Browser	IE 4.01	• Current VOGUE system is not tested with Netscape • Navigator for indexing
Searching and Indexing **Searching and Indexing cont.**	Microsoft Index Server 1.1 Microsoft Index Server 1.1	• Index the full text and properties of documents on IIS-based server • Simple search and full content search • Search using Web Browser (IE4 or later) • Requires IIS or PWS • New version 2.0 is available • with NT Server Option Pack • http://www.microsoft.com/

7. VOGUE Structure and Components

The VOGUE system uses the Wang Image editor to manipulate document images. A Web browser provides a front-end access capability to declassification guides. Data entry forms are used to store revie information resulting from the review process. The NT Server with Web Server and Index Server provide indexing and searching functionality to improve declassification processes. Index Server is used to create and maintain index information of available declassification guides. This index information is used to find relevant information based on reviewer criteria. The relational database contains document images, declassification guides, and review information. Finally, DragonDictate provides voice operation capabilit to the VOGUE syste



7.1. Wang Image Editor: Wang Imaging for Windows is a built-in component of Windows NT. It provides required editing capability to the reviewer such as stamping necessary information and highlighting sensitive sentences.

7.1.1. Wang Imaging for Windows N

7.1.2. Show document image (tiff format)

7.1.2.1. Zoom In/Out

7.1.3. Edit document image based on review results

7.1.3.1. Highlighting

7.1.3.2. Marking

7.1.3.3. Stamping

7.2. Web Browser: Microsoft Internet Explorer 4.01 is used to ensure compatibility with other components. (e.g., MS Index Server, Internet Information Server, etc.)

7.2.1. MS Internet Explorer 4.0 or later

7.2.2. Basic tool to view declassification guides

7.2.3. View declassification guides (Full text)

7.2.3.1. View search results

7.2.3.1.1. Condensed search results: Show summary of search results containing search information

7.2.3.1.2. Full content search results: Show full content of results containing search results

7.3. MS Index Server: MS Index Server is an add-on component of Windows NT server. It is available from Microsoft, free of charge. It requires Windows NT as well as Web Server to operate. Microsoft released a new version (Index Server 2.0) as part of the NT Option Pack. This is a key component for being able to conduct an efficient declassification review process.

7.3.1. MS Index Server 1.1

7.3.2. Content-indexing and searching

7.3.2.1. Index full text and properties

7.3.2.2. Index HTML documents

7.3.2.3. Index MS Word and Excel

7.3.3. Automatic maintenance

7.3.3.1. Index creation and updates

7.3.3.2. Optimization

7.3.4. Web-based interface

7.3.5. Customized Query forms

7.3.5.1. Compound words querying

7.3.6. Requirement

7.3.6.1. Internet Information Server for Windows NT 4.0 Server

7.3.6.2. Peer web Server for Windows NT 4.0 Workstation

7.4. Access Database: Database will store and maintain relevant information throughout the declassification process. Document images and declassification guides can be managed independently to VOGUE.

7.4.1. Table for document images

7.4.2. Table for Declassification Guides

7.4.3. Table for review information: Review results will be stored in this table.

7.5. Voice Operation: DragonDictate can be easily integrated with any application developed b using Visual Basic. Based on industry review it has a great performance as well as functionality.

7.5.3. Operate application with voice inputs and commands

7.5.4. Voice Text Dictation

7.5.5. Text to speech: It can talk back to you to confirm your voice input

8. VOGUE Word Lists and Operation Instruction

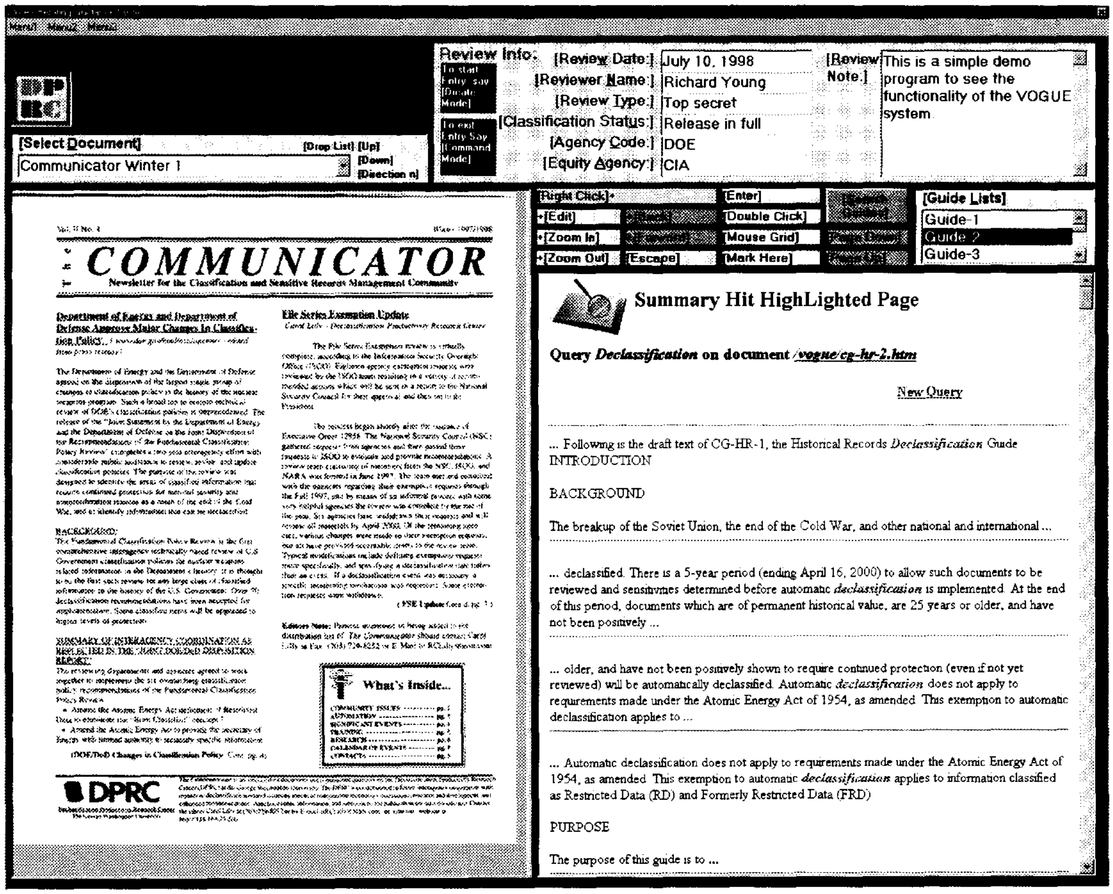
Operation	**Word Lists**	**Notes**
Before you start VOGUE		• Open DragonDictate • Turn on the microphone using keyboard or mouse
To open VOGUE	[Bring Up] "VOGUE"	• To open VOGUE syste
To close VOGUE	[Close] "Window	• To close VOGUE syste
To navigate VOGUE	[Select Document], [Review Date], [Reviewer Name], [Review Type], [Classification Status], [Agenc Code], [Equity Agency], [Review Notes], [Search Guides], [Guide Lists]	• To navigate data entry fields or command fields
To enter data or information	[Dictate mode]	• Data can be entered into an data entry fields using voice dictation
To edit Document Images	[Zoom in], [Zoom Out], [Edit], etc.	• DragonDictate provides word lists related to WANG Image editor
To search Declassification Guides	[Back], [Forward], etc.	• DragonDictate provides word lists related to Internet Explorer Web browser
DragonDictate general word lists (partial lists)	[Up], [Down], [Page Up], [Page Down], [Drop List], [Double click], [Mouse Grid], [Mark Here], [Enter], [Escape], [Go to Sleep], [Wake Up], [Microphone Off]	• DragonDictate provides word lists for general windows operation

DragonDictate Mode	**Summary**
[Command Mode]	• Control Windows application by speaking instead of by keyboard and

	mouse • Speech command can be combined with mouse and keyboard commands
[Dictate Mode]	• Enter text directly into your applications by speaking • Support dictation as well as text formatting
[Sleep Mode]	• Inactivate voice control • [Wake up] will activate voice operation
[Mouse Grid Mode]	• Control mouse position on the screen by using voice • Use mouse grid by using voice • Control mouse movement by selecting grid point

9. Sample VOGUE Screen

The VOGUE screen consists of three interrelated windows: (1) the document review window (left side) for presentation of an image or ASCII text file to be reviewed, (2) the guidance window (right side) for search and presentation of the identified guidance topics, and (3) the results window (upper-right) for capture and presentation of the results of the review.


Menu1 Menu2 Menu3
To start Entry say [Dictate Mode]
To exit Entry Say [Command Mode]
[Drop List] [Up]
[Down]
[Direction n]
[Right Click]
[Edit]
[Zoom In]
[Zoom Out]
[Escape]
[Enter]
[Double Click]
[Mouse Grid]
[Mark Here]
New Query
COMMUNICATOR
Newsletter for the Classification and Sensitive Records Management Community
Department of Energy and Department of Defense Approve Major Changes In Classification Policy
File Series Exemption Update
What's Inside...
DPRC

Review Info:

[Review Date:] July 10, 1998

[Reviewer Name:] Richard Young

[Review Type:] Top secret

[Classification Status:] Release in full

[Agency Code:] DOE

[Equity Agency:] CIA

[Review Note:] This is a simple demo program to see the functionality of the VOGUE system

[Select Document] Communicator Winter 1

[Guide Lists] Guide-1 Guide 2 Guide-3

Summary Hit HighLighted Page

Query *Declassification* on document */vogue/cg-hr-2.htm*

... Following is the draft text of CG-HR-1, the Historical Records *Declassification* Guide INTRODUCTION

BACKGROUND

The breakup of the Soviet Union, the end of the Cold War, and other national and international ...

... declassified. There is a 5-year period (ending April 16, 2000) to allow such documents to be reviewed and sensitivities determined before automatic *declassification* is implemented. At the end of this period, documents which are of permanent historical value, are 25 years or older, and have not been positively ...

... older, and have not been positively shown to require continued protection (even if not yet reviewed) will be automatically declassified. Automatic *declassification* does not apply to requirements made under the Atomic Energy Act of 1954, as amended. This exemption to automatic declassification applies to ...

... Automatic declassification does not apply to requirements made under the Atomic Energy Act of 1954, as amended. This exemption to automatic *declassification* applies to information classified as Restricted Data (RD) and Formerly Restricted Data (FRD)

PURPOSE

The purpose of this guide is to ...

10. Results of Prototype Demonstration

This research prototype was focused on demonstrating a level of functionality to support the declassification review process. The prototype system demonstrated the following capabilities:

1) Reviewers were able to simultaneously see documents images and relevant declassification guides in the same screen
2) Reviewers were able to use voice, and/or mouse and keyboard to operate the syste
3) The system provided a capability of searching declassification guides based on sensitive words in the documents
4) The system provided search results for review and storage

The prototype VOGUE system successfully demonstrated all of the intended system functionality, as specified in section 5, above. Reviewers were able to conduct declassification review processes with a simple VOGUE screen that can handle both document images and declassification guides. Reviewers were able to use voice, and/or mouse and keyboard interfaces to operate the VOGUE system. Reviewers should be able to identify for them selves which operating method works best for their situation, based on their own experience, work load, etc.

Several issues were raised in regard to VOGUE operations during the initial prototype demonstration. These were related mainly to VOGUE's performance, compared to current manual processes:

1. The system appeared to be too slow and did not demonstrate a clear advantage over normal/manual operations using a mouse and keyboard.
2. Voice training appeared to be a key factors for improve performance and/or accuracy of data entry. (Dragon Systems claims little training is required to effectively use DragonDictate)

These issues were addressed and subsequently mitigated by means of further system tuning. The final prototype system demonstrated that the expanded choice of input devices provided by VOGUE is indeed practical. VOGUE provides reviewers a powerful optional means of interfacing with the computer, but its effectiveness depends on how and for which specific operations the reviewers elect to use it.

The prototype demonstration also showed that further consideration should be given to optimization of the VOGUE operating environment. Background noise, for example, was very important during the demonstration, because of the use of an omni-directional microphone. DragonDictate tries to convert any sound to meaningful words, and questions and explanations of the system during the demonstration were often wrongly interpreted by the system as operator commands. This problem could be mitigated by better room arrangement and use of a (more) directional microphone of the type used by telephone operators. The prototype also demonstrated that operator skill in the use of the microphone can be trained, and goes a long way towards improving system effectiveness.

VOGUE Performance Summary

Training Time for Dragon Dictate	2.5 hrs
Ease of Training	Simple (packaged with Dragon Dictate)

Difficulty of use for Vogue syste	Novice (requires practice at simple commands)
Speech Rate (for demonstration)	70 words/ 1.5 to 2 minutes or 32 commands / 1.5 to 2 minutes
Performance Level	95% accurate for demo; accuracy increases as Vogue "learns" your speech patterns

Microsoft® Index Server (MIS--bundled free of charge with NT 4.0) apparently provides adequate capabilities for indexing and full text searching of documents. As an add-on software component with Windows NT, MIS is readily available for general applications of VOGUE. It is also designed for use with documents stored on Intranet or Internet sites. The following list (provided by Microsoft) shows the advantages and capabilities of MIS as a search engine:

- **Indexes all documents:** Allows the end user to query indexes and entire documents on Intranet or Internet sites that are stored on Windows NT® Server 4.0 operating system with Internet Information Server. The search engine can find documents in just about any format: text in a Microsoft Word document, statistics on a Microsoft Excel spreadsheet, or the content of an HTML page.
- **Customized query form:** Allows the Webmaster to create a customized query form enabling end users to choose certain perimeters of their search. This form modification allows a user to search by either contents or other document properties such as author and subject.
- **Customized results page:** Allows the Webmaster to customize the results page shown after a search. Variables include any document properties such as title, author, date, size, and a document summary. The number of hits shown per page can also be set, and results can be sorted by an property. Index Server 2.0 also supports hit highlighting where the search words are shown highlighted in the document.
- **Automatic maintenance:** Provides a "Zero Maintenance" environment, especially where a server will be running 24 hours a day, seven days a week. Once set up, all the operations are automatic. This includes; automatic updates, index creation and optimization, and crash recovery in case there's a power failure.
- **Administrative tools:** There are a number of built-in tools to help administrators optimize their query service. The performance monitoring capability gives administrators key information to gauge site performance—including the number of queries processed and the response time.
- **Multiple languages:** Provides built-in language support allowing end users to query documents in seven different languages. Documents written in Dutch, English (U.S. and International), French, German, Italian, Spanish, and Swedish can be searched.
- **Extensible architecture:** Uses *content filters* to extract the textual information contained within a formatted file. Content filters are associated with particular document formats. Content filters comply with the IFilter ActiveX™ programmatic interface, which has been published b

Microsoft. By writing a content filter, software authors can expose their contents to Index Server for indexing and retrieval by end users.

11. Directions for Further Research

While it is clear that the VOGUE concept has good potential for applicability to real world document declassification operations, practicalities mandate that the VOGUE system be refined and further tuned in order to provide the level of advantages necessary to justify its installation and training expenses. The following is a list of recommended future research activities to improve the VOGUE system.

- Approaches to tuning the VOGUE system for increased performance and effectiveness of should be researched. These should be baselined against current normal/manual operations for realistic comparisons. The first step is to develop a better understanding of the current system and the functional requirements of the users. This knowledge should also help provide measures or "system metrics" to use when comparing manual operations to a system including VOGUE. As a minimum, this research should focus on system speed (e.g. general operation, screen navigation, and data entry) and system flexibility and scalability.
- Research should also be directed to systems analysis interactions with declassification revie personnel from the CIA and other agencies for the purpose of better understanding their operation and computer-interface needs and requirements. The prototype VOGUE system was based on our general understanding of the review process. Interaction with reviewers should greatly improve understanding of the actual requirements of the declassifiers.
- Further research on the VOGUE operating environment should be carried out to optimize syste performance. Operating environmental concerns include the configuration of the physical operating space as well as the integration of VOGUE into the existing declassification revie processes.
- New technology should be considered for adoption/integration into VOGUE whenever there is promise for significant improvement in system performance. Voice recognition and related technologies are changing rapidly. Further research should (as a minimum) consider
    1) voice/speech recognition
    2) storage of document images and declassification guides
    3) intelligent guide search and retrieval.
- Finally, the VOGUE system should be handed over as a pilot to the most adventurous of the declassifiers for use and feedback. This step should provide solid information on what works, what does not and future directions for progress.

Database Partitioning and Duplicate Document Detection Based on Optical Correlation

Francine Prokoski, Ph.D.
MIKOS Ltd., Fairfax Station, VA
mikos@gte.net

ABSTRACT

Certain classes of duplicate documents can be rapidly and accurately detected by first partitioning the database through characterizing each page of the document as to the amount and location of white space, the number of lines of text, and the presence of graphics. This is followed by optical correlation to compare a target document against database documents which have the same characterization. The characterization effectively partitions the database such that only a small percentage of the database needs to be directly compared. The comparator process used, called FlashCorrelation[Æ], is fast and simple to implement, as is the characterization process. The resulting duplicate detection method has been demonstrated with small databases of 600 documents in which duplicates have been created through rescannings, degraded copying, light and heavy overwritings, marginalia, wrinkling, obliterations, and appendages. Results appear promising, and further testing with significantly larger databases should be the next step.

Types of Duplicate Documents

Optical correlation techniques are suitable for matching documents which are optical copies, produced through such means as carbons, mimeographs, Xeroxing, facsimile, optical scan and reprint, and photography. While subsequent annotations, whiteouts, and appendages can cause two documents to differ, their optical correlation will still indicate significant similarity. Such techniques can offer speed and robustness against local or global variations involving additions, subtractions, and histogra modifications. When the database includes electronic copies, which may have different fonts and page layouts from the original, the correlator would be required to systematically consider a range of font types and sizes and margin settings. The speed of the process makes that feasible for the software used in the present tests, but a hardware embodiment would be needed to search a wide range of variables over large database partitions. For the current declassification requirements, only optical copies are considered.

Duplicates which can be identified through the proposed method span four classes:

- Exact Duplicate (carbon, fax, Xerox, mimeograph having no significant alterations)
- Almost Exact Duplicate (signed vs. unsigned or having some other minor alterations)
- Near Duplicate (adding marginalia, cover sheet, or appendix)
- Partial Duplicate (resulting from "cut and paste" having subtractions and additions).

A threshold setting on the correlator determines the amount of variation allowed for a database document to be considered a candidate duplicate of a target document.

The Level at which the matching of two documents is to be performed can be:

- single page matches
- all pages in a document match
- all pages in one document match a subset of pages in the other document
- some pages in one document match some pages in the other document.

Depending on the number of pages in the documents, the correlation can be performed on a page basis in each of these cases, or can consider a mosaic of all pages at once. The basic technique is the same in either approach. Page-based matching will be described for simplicity in this paper.

Document Standardization

In order to perform optical correlation with maximum accuracy, the documents must be deskewed and aligned to consistent axes. The amount of scaling variation which can be tolerated is a function of the precision with which the deskewing and aligning is performed. Deskewing and location of the page origin is performed at the time the document is scanned, while the human operator is keying-in such information as: the accession number, date, origination agency, etc. Documents are all scanned at the same resolution (currently 300 dpi), and the digital files can then be directly processed by the FlashCorrelationô (FC) technique. A constant size array is used by the FC engine, and is chosen to be large enough that it encompasses the entirety of each page in the database.

Document and Database Characterization

An optical copy, by definition, must have similar allocation of white space, text, and graphics on each page. Therefore, there is no need to consider as duplicates any pair of documents which have differing allocations, beyond the extent that marginalia or other variations may be tolerated in the definition of what is being considered a duplicate document.

Preprocessing a database into partitions provides significant speed acceleration during duplicate detection. In areas which have not been subjected to modification, optical duplicates must necessaril have the same amount and location of white space, and also must have the same number of lines of text. Partitions can be established using three parameters which can be automatically calculated and added to the document header. As each document is scanned, the system automatically:

A) Divides each page into 8 x 10 cells, and specifies a binary array (80 x # pages) in which a "1" is positioned for each non-white cell and a "0" for each white cell in each page.
B) Determines the number of lines of text on each page, and produces a string of counts corresponding to each page in the document
C) Detects the presence of graphics on each page and produce a binary string in which a "0" indicates no graphics and a "1" indicates graphics for each page.

Although the white cell allocation matrix alone would in theory provide 2^{80} partitions, smudges, speckles from photocopying, and other types of visual debris may result in ambiguity as to whether a cell is to be considered white or not. In such cases, where the total number of non-white pixels is greater than zero but less than a specified limit, the cell is classified an indeterminate. Depending on the quality of the target document and the document database, on the order of 1000 to 10,000 partitions can be repeatably developed, providing an effective filtering out of impossible matches without eliminating possible candidates. Duplicate detection correlation then is performed against the partition of the database which has the same A, B, and C as a target document, and against neighboring partitions based upon the level of modifications to be allowed between duplicates, the image quality of the documents, and the extent of preprocessing for image enhancement..

In a generalized database, a "graphic" can be defined as a consecutive sequence of non-white rows with extent at least equal to five lines of single-spaced text. Documents which are current candidates for declassification do not have graphics, and so that characterization feature is not further considered in this paper.

Generated Test Sets

The initial test performed used a database including standard government forms, newspaper extracts, business letters, magazine pages, and catalogues. The second test utilized a database provided b DOE/LANL which includes redacted pages from government documents considered representative of current declassification requirements. In each test, the original documents were scanned at 300 dpi and several replicas printed with a laser printer. The replicas were then subjected to skewing, wrinkling, overwriting, coffee stains, whiteout, and smudges, after which they were rescanned and added to the database. For convenience, other modifications were made to the electronic version of the original, and then added to the database without being printed and rescanned. These included histogram changes, deletions, and additions. The results showed that the proposed method can detecting duplicates which have undergone this wide range of modifications. Illustrating the degree to which variations can be allowed in optical duplicate detection:

Figure 1: Light or Heavy Annotation (First Test Database Document)

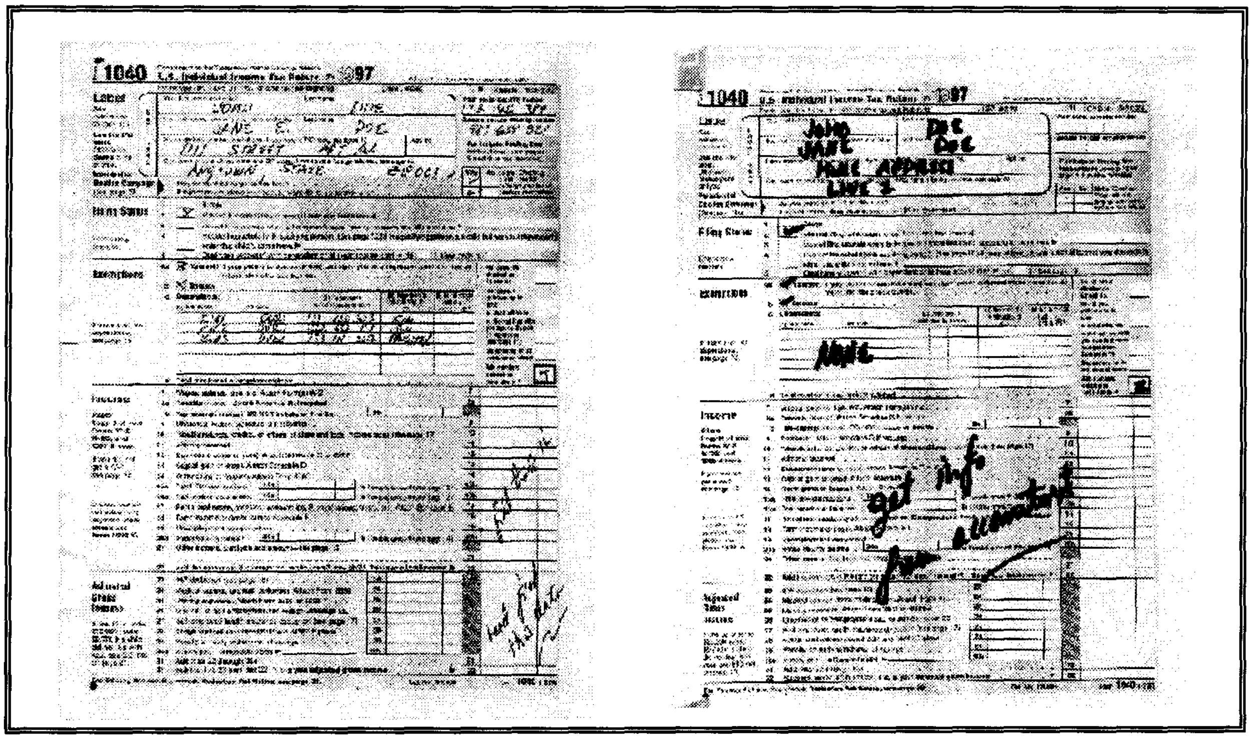
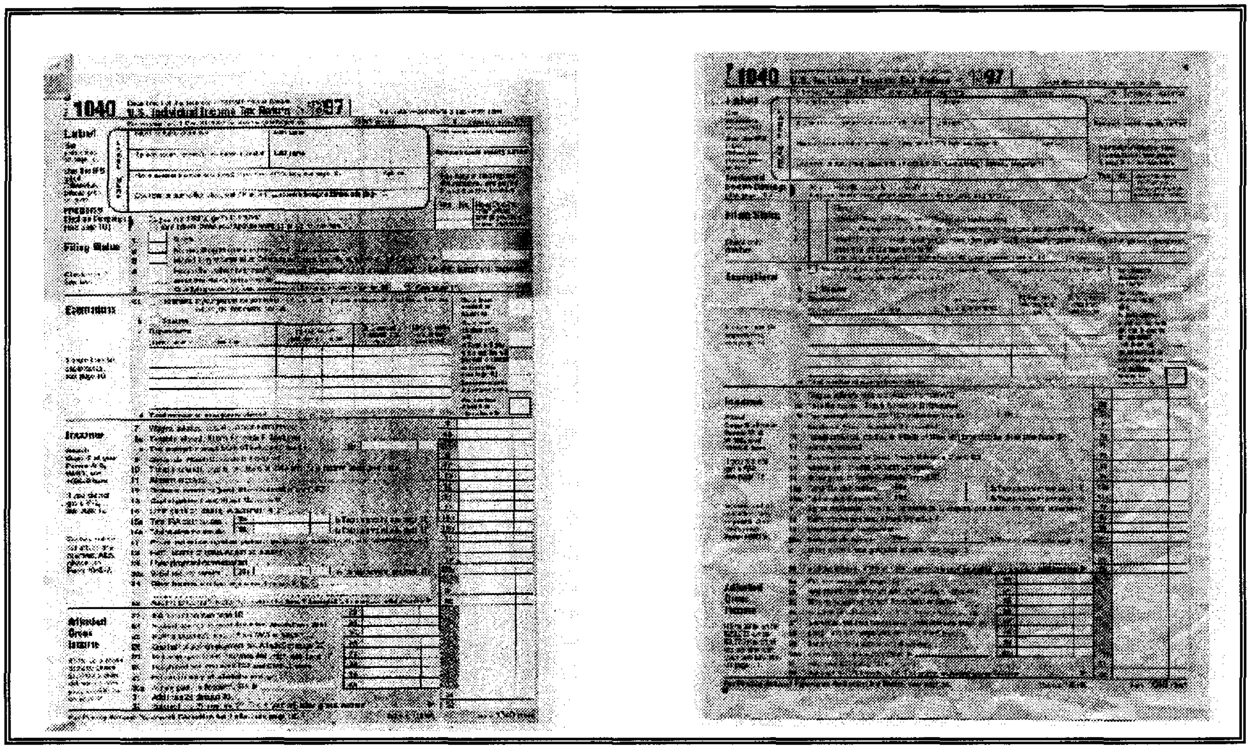

1040
1997
1040
1997
John Doe
Jane Doe
get info
for accountant

Figure 2: Xeroxing and Wrinkling (First Test Database Document)



1040 U.S. Individual Income Tax Return 1997

Label

Filing Status

Exemptions

Income

Adjusted Gross Income

1040 U.S. Individual Income Tax Return 1997

Label

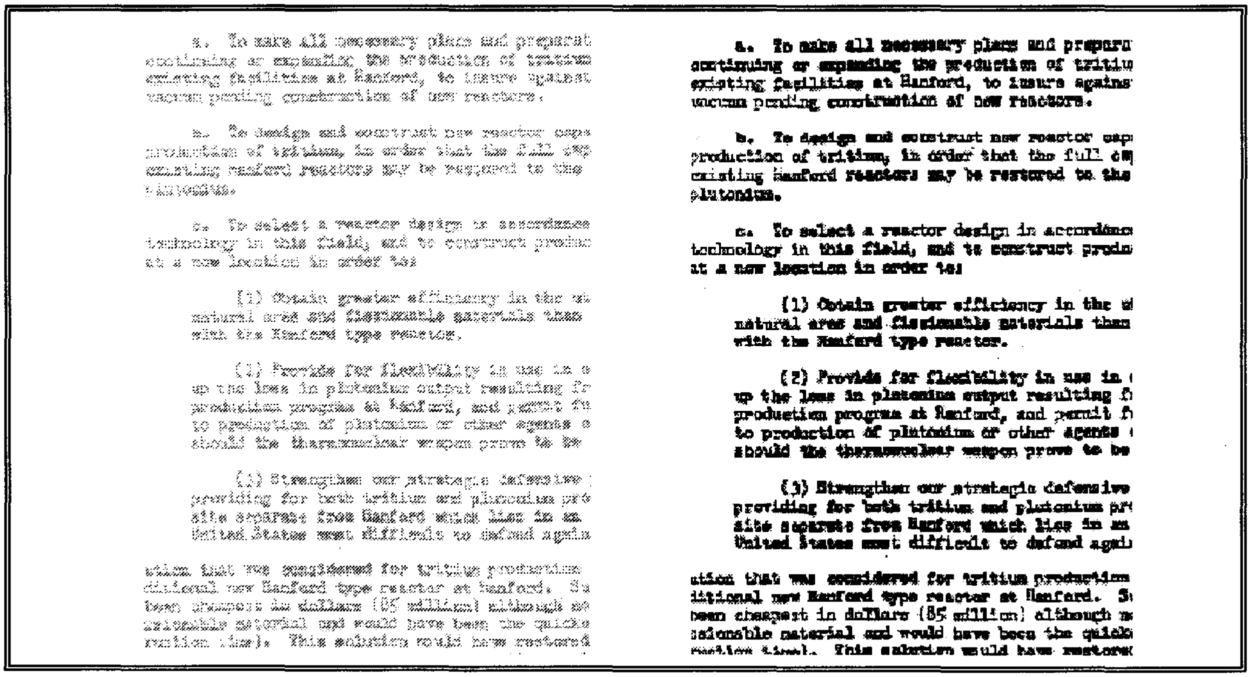
Filing Status

Exemptions

Income

Adjusted Gross Income

Figure 3: Second generation Xeroxing and Second generation Re-scan (LANL Document 371)



a. To make all necessary plans and prepara continuing or expanding the production of tritiu existing facilities at Hanford, to insure agains vacuum pending construction of new reactors.

b. To design and construct new reactor cap production of tritium, in order that the full ca existing Hanford reactors may be restored to the plutonium.

c. To select a reactor design in accordance technology in this field, and to construct produ at a new location in order to:

(1) Obtain greater efficiency in the u natural area and fissionable materials than with the Hanford type reactor.

(2) Provide for flexibility in use in up the loss in plutonium output resulting f production program at Hanford, and permit f to production of plutonium or other agents should the thermonuclear weapon prove to be

(3) Strengthen our strategic defensive providing for both tritium and plutonium pr site separate from Hanford which lies in an United States most difficult to defend agai

ation that was considered for tritium production ditional new Hanford type reactor at Hanford. S been cheapest in dollars (85 million) although m issionable material and would have been the quick ruction time). This solution would have restored

a. To make all necessary plans and prepara continuing or expanding the production of tritiu existing facilities at Hanford, to insure agains vacuum pending construction of new reactors.

b. To design and construct new reactor cap production of tritium, in order that the full ca existing Hanford reactors may be restored to the plutonium.

c. To select a reactor design in accordance technology in this field, and to construct produ at a new location in order to:

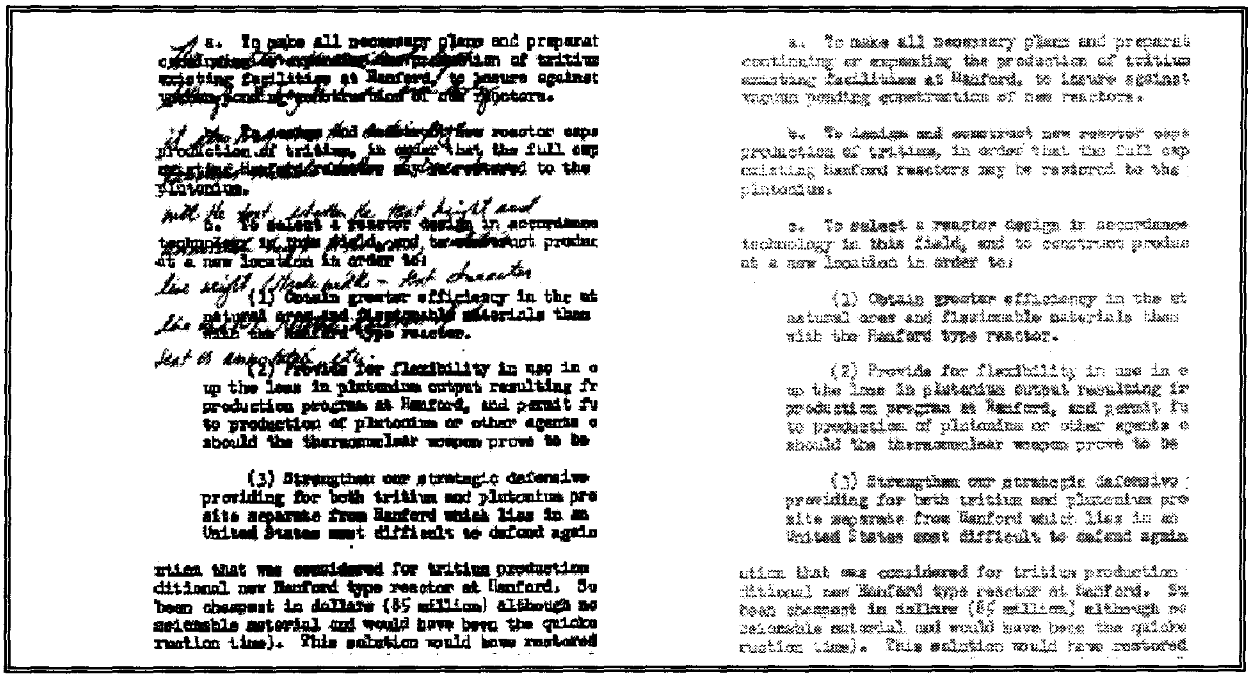
(1) Obtain greater efficiency in the u natural area and fissionable materials than with the Hanford type reactor.

(2) Provide for flexibility in use in up the loss in plutonium output resulting f production program at Hanford, and permit f to production of plutonium or other agents should the thermonuclear weapon prove to be

(3) Strengthen our strategic defensive providing for both tritium and plutonium pr site separate from Hanford which lies in an United States most difficult to defend agai

ation that was considered for tritium production ditional new Hanford type reactor at Hanford. S been cheapest in dollars (85 million) although m issionable material and would have been the quick ruction time). This solution would have restored

Figure 4: Annotated and Lightened Re-scan (Document 371)



a. To make all necessary plans and preparat
continuing or expanding the production of tritium
existing facilities at Hanford, to insure against
vacuum pending construction of new reactors.

b. To design and construct new reactor expa
production of tritium, in order that the full cap
existing Hanford reactors may be restored to the
plutonium.

c. To select a reactor design in accordance
technology in this field, and to construct produc
at a new location in order to:

(1) Obtain greater efficiency in the ut
natural ores and fissionable materials than
with the Hanford type reactor.

(2) Provide for flexibility in use in o
up the loss in plutonium output resulting fr
production program at Hanford, and permit fu
to production of plutonium or other agents o
should the thermonuclear weapon prove to be

(3) Strengthen our strategic defensive
providing for both tritium and plutonium pro
site separate from Hanford which lies in an
United States most difficult to defend again

ation that was considered for tritium production
ditional new Hanford type reactor at Hanford. Su
been cheapest in dollars (85 million) although no
fissionable material and would have been the quicke
ruction time). This solution would have restored

a. To make all necessary plans and preparat
continuing or expanding the production of tritium
existing facilities at Hanford, to insure against
vacuum pending construction of new reactors.

b. To design and construct new reactor expa
production of tritium, in order that the full cap
existing Hanford reactors may be restored to the
plutonium.

c. To select a reactor design in accordance
technology in this field, and to construct produc
at a new location in order to:

(1) Obtain greater efficiency in the ut
natural ores and fissionable materials than
with the Hanford type reactor.

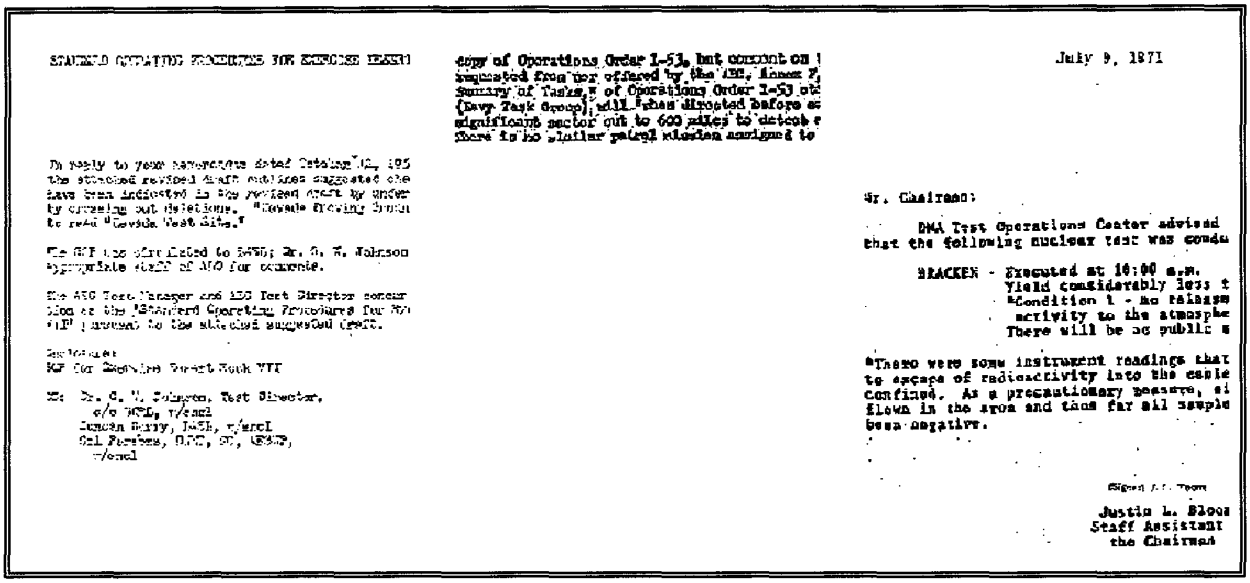
(2) Provide for flexibility in use in o
up the loss in plutonium output resulting fr
production program at Hanford, and permit fu
to production of plutonium or other agents o
should the thermonuclear weapon prove to be

(3) Strengthen our strategic defensive
providing for both tritium and plutonium pro
site separate from Hanford which lies in an
United States most difficult to defend again

ation that was considered for tritium production
ditional new Hanford type reactor at Hanford. Su
been cheapest in dollars (85 million) although no
fissionable material and would have been the quicke
ruction time). This solution would have restored

The initial testing utilized a general database of text documents, images, and combinations. While that application may be significant for future declassification efforts, the current requirements are for documents without images. The LANL-supplied database is considered illustrative of the range of characteristics in documents currently awaiting declassification. The LANL database documents have a wide variation in features, including:

Figure 5: Variation in amount and location of White Space (Documents 111, 171, and 18)



STANDARD OPERATING PROCEDURES FOR [illegible]

In reply to your memorandum dated October 12, 195
the attached revised draft outlines suggested [illegible]
have been indicated in the revised draft by under
by crossing out deletions. "[illegible]
to read "Nevada Test Site."

The SOP was circulated to [illegible]; Dr. G. W. Johnson
appropriate staff of AEC for comments.

The AEC Test Manager and [illegible] Test Director concur
tion as the "Standard Operating Procedures for [illegible]
[illegible] pursuant to the attached suggested draft.

Enclosure:
SOP for [illegible] Test Book VII

cc: Dr. G. W. Johnson, Test Director,
c/o [illegible], w/encl
[illegible] Berry, [illegible], w/encl
Col [illegible], [illegible],
w/encl

copy of Operations Order 1-53, but comment on
suggested from that offered by the AEC, Annex F,
Summary of Tasks," of Operations Order 1-53 of
(Navy Task Group), will "when directed before a
significant sector out to 600 miles to detect
There is no similar patrol mission assigned to

July 9, 1971

Mr. Chairman:

DNA Test Operations Center advised
that the following nuclear test was condu

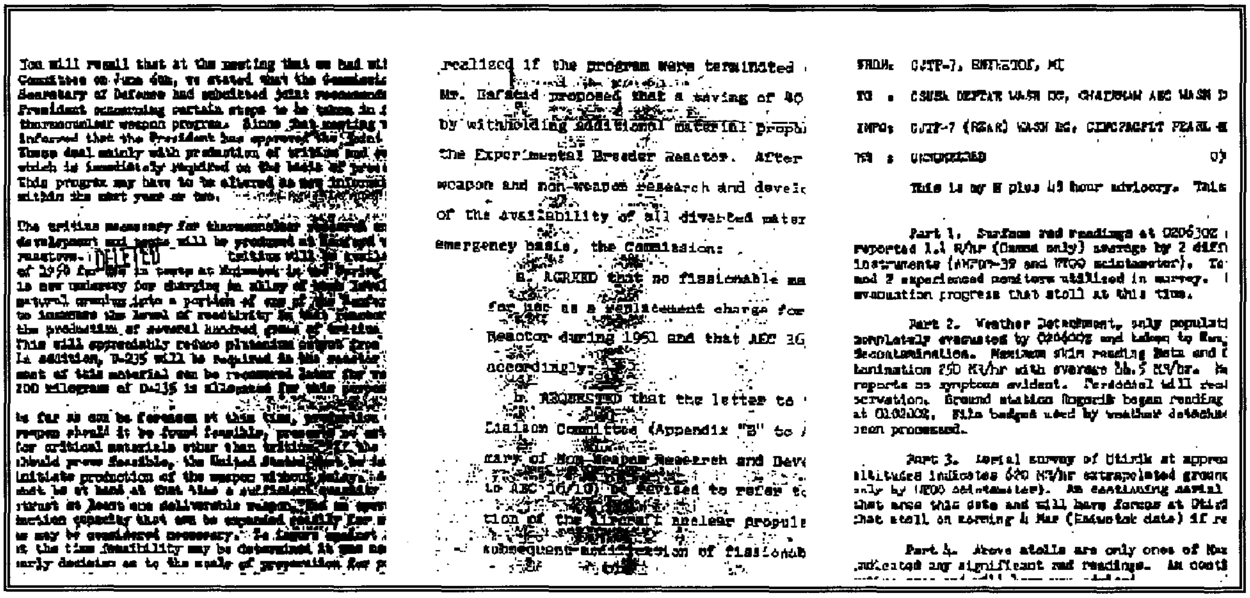
BRACKEN - Executed at 10:00 a.m.
Yield considerably less t
*Condition 1 - No release
activity to the atmosphe
There will be no public a

*There were some instrument readings that
to escape of radioactivity into the cable
confined. As a precautionary measure, ai
flown in the area and thus far all sample
been negative.

Signed J. L. Bloom

Justin L. Bloom
Staff Assistant
the Chairman

Figure 6: Variations in line spacing and streaking (Documents 370, 375, and 185)



You will recall that at the meeting that we had wit
Committee on June 4th, we stated that the Commissio
Secretary of Defense had submitted joint recommenda
President concerning certain steps to be taken in t
thermonuclear weapon program. Since that meeting w
informed that the President has approved the joint
These deal mainly with production of tritium and [illegible]
which is immediately required on the basis of presi
This program may have to be altered as new informati
within the next year or two.

The tritium necessary for thermonuclear research an
development and tests will be produced at Hanford w
reactors. DELETED tritium will be availa
of 1950 for use in tests at Eniwetok in the Spring
is now necessary for charging an alloy of [illegible]
natural uranium into a portion of one of the Hanfor
to increase the level of reactivity in that reactor
the production of several hundred grams of tritium
This will appreciably reduce plutonium output from
In addition, U-235 will be required in the [illegible]
most of this material can be recovered later for we
100 kilograms of U-235 is allocated for this purpos

As far as can be foreseen at this time, production
weapon should it be found feasible, [illegible]
for critical materials other than tritium. If the
should prove feasible, the United States [illegible]
initiate production of the weapon without delay, it
must be at hand at that time a sufficient quantity
struct at least one deliverable weapon, and an oper
duction capacity that can be expanded rapidly for s
as may be considered necessary. To insure against
at the time feasibility may be determined it [illegible]
early decision as to the scale of preparation for p

realized if the program were terminated
[illegible]
Mr. [illegible] proposed that a saving of 40
[illegible]
by withholding additional material propo
The Experimental Breeder Reactor. After
weapon and non-weapon research and develo
of the availability of all diverted mater
emergency basis, the Commission:

a. AGREED that no fissionable ma
for use as a replacement charge for
Reactor during 1951 and that AEC 3G
accordingly;

b. REQUESTED that the letter to
Liaison Committee (Appendix "B" to
mary of Non-Weapon Research and Deve
to AEC 16/10) be revised to refer to
tion of the Aircraft Nuclear Propuls
subsequent [illegible] of fissionab

FROM: CJTF-7, ENIWETOK, MI

TO : CSUSA DEPTAR WASH DC, CHAIRMAN AEC WASH D

INFO: CJTF-7 (REAR) WASH DC, CINCPACFLT PEARL H

TS : UNCLASSIFIED 0)

This is my H plus 48 hour advisory. This

Part 1. Surface rad readings at 020630Z
reported 1.4 R/hr (Gamma only) average by 2 diff
instruments (AN/PDR-39 and MT00 scintameter). Te
and 2 experienced monitors utilized in survey.
evacuation progress that atoll at this time.

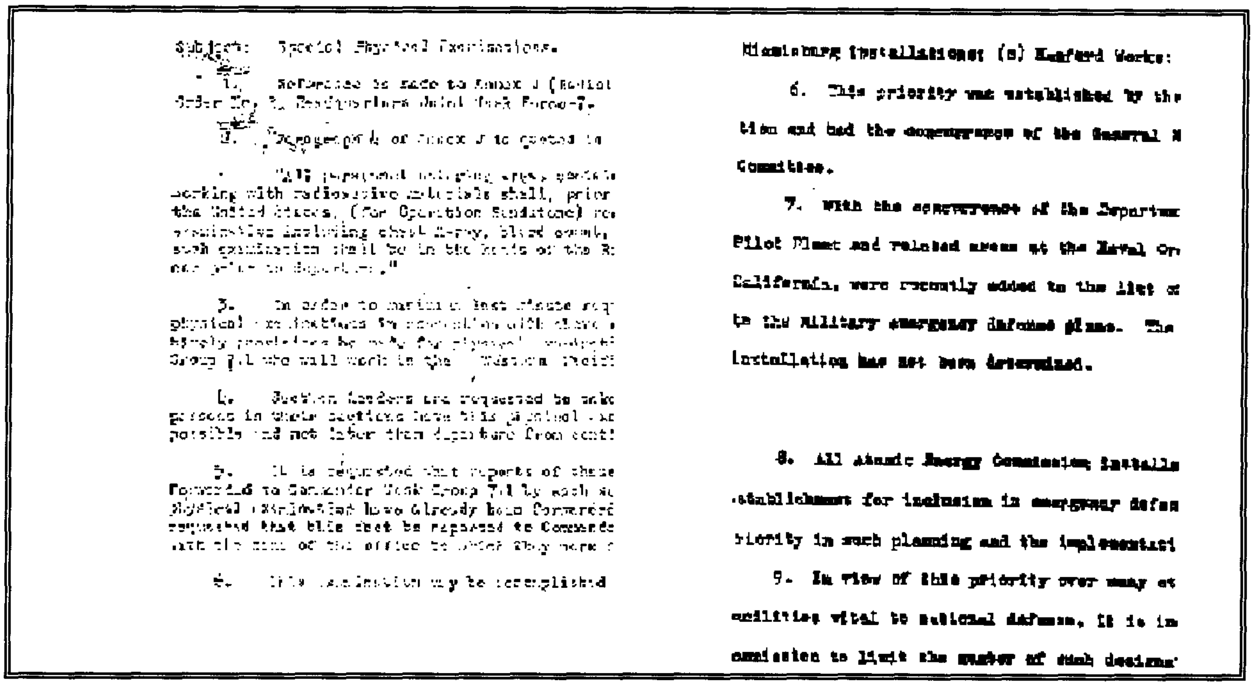
Part 2. Weather Detachment, only populat
completely evacuated by 020600Z and taken to [illegible]
decontamination. Maximum skin reading Beta and G
lamination 250 MR/hr with average 26.5 MR/hr. No
reports no symptoms evident. Personnel will rema
servation. Ground station Rongerik began reading
at 010200Z. Film badges used by weather detachm
been processed.

Part 3. Aerial survey of Utirik at approx
altitudes indicates 620 MR/hr extrapolated ground
only by MT00 scintameter). An continuing aerial
that area this date and will have forces at Utiri
that atoll on morning 4 Mar (Eniwetok date) if re

Part 4. Above atolls are only ones of Mar
indicated any significant rad readings. An conti

The types and extent of variations present or expected in a database are considered in establishing the characteristics used for partitioning, and the selection of neighboring partitions to be included in the duplicate searches. Additional analysis of the row sums used to determine the number of lines of text on a page can also produce specifications on the font size(s) and the font(s) used. For the small database sizes used to date, that additional partitioning produces no gain. However, for very large databases, or for detection of electronically rendered copies, it could provide significant further characterization of each document.

Figure 7: Variations in Fonts and Intensity (Documents 482 and 9)



Subject: Special Physical Examinations.

1. Reference is made to Annex J (Radiol
Order No. 2, Headquarters Joint Task Force-7.

2. Paragraph 4 of Annex J is quoted in

"All personnel [illegible] [illegible]
working with radioactive materials shall, prior
the United States, (for Operation Sandstone) re
examinations including chest x-ray, blood count,
such examination shall be in the hands of the R
[illegible] prior to departure."

3. In order to maintain [illegible] [illegible]
physical examinations in accordance with [illegible]
[illegible] [illegible] [illegible] [illegible]
Group 7.1 who will work in the [illegible] [illegible]

4. Section Leaders are requested to make
persons in their sections have this physical exa
possible and not later than departure from cont

5. It is requested that reports of these
forwarded to Commander Task Group 7.1 by each se
Physical examinations have already been performed
requested that this fact be reported to Commande
with the name of the office to which they were

6. This examination may be accomplished

Hisslsburg installations: (e) Hanford Works:

6. This priority was established by the
tion and had the concurrence of the General M
Committee.

7. With the concurrence of the Departm
Pilot Plant and related areas at the Naval Or
California, were recently added to the list o
to the military emergency defense plans. The
installation has not been determined.

8. All Atomic Energy Commission installa
establishments for inclusion in emergency defen
riority in such planning and the implementati

9. In view of this priority over many ot
facilities vital to national defense, it is im
commission to limit the number of such designa

Features of FlashCorrelationÆ

FlashCorrelationÆ (FC) is a highly efficient patented method for comparing complex patterns, such as faces, fingerprints, ballistics, signatures, and generalized documents containing text, graphics and images. FC has been implemented in software and shown to have advantages over other techniques for facial recognition, ballistics matching, and 2-D bar coding. FC offers the speed of optical correlation with the flexibility of digital correlation. Compared to other pattern-based matching schemes, it is faster, less computationally intensive, forgiving of variations in color and resolution, and cheap to implement. The technique requires no adaptive training and runs on any PC.

FC compares two or more patterns to determine their degree of similarity regardless of the complexit of the images, and in spite of the addition of noise, local changes, and variations in resolution and focus. A Flash Correlation Artifact (FCA) is produced if and only if two patterns are highly correlated. The FCA also provides quantitative valuation of the degree of correlation, allowing the matching engine to be tuned to the desired level of match. In the present application, the threshold can be set to select all documents which have at least a desired level of match, ranging over exact duplicates, almost exact duplicates, and partial duplicates.

Document comparison can be done simultaneously for all pages of a short document (10 pages or less), or page-by-page, or each page can be divided into cells and corresponding cells compared. The contribution from each cell to the total page correlation can be weighted by the location of the cell on the page. For example, if marginalia is to be allowed in duplicates, then cells along the edge of the page are given reduced weight. When the document may have localized damage such as from coffee stains, or when annotation may be present anywhere on a page, the measure of similarity between two pages can be based on the maximum correlation value from a subset of the cells.

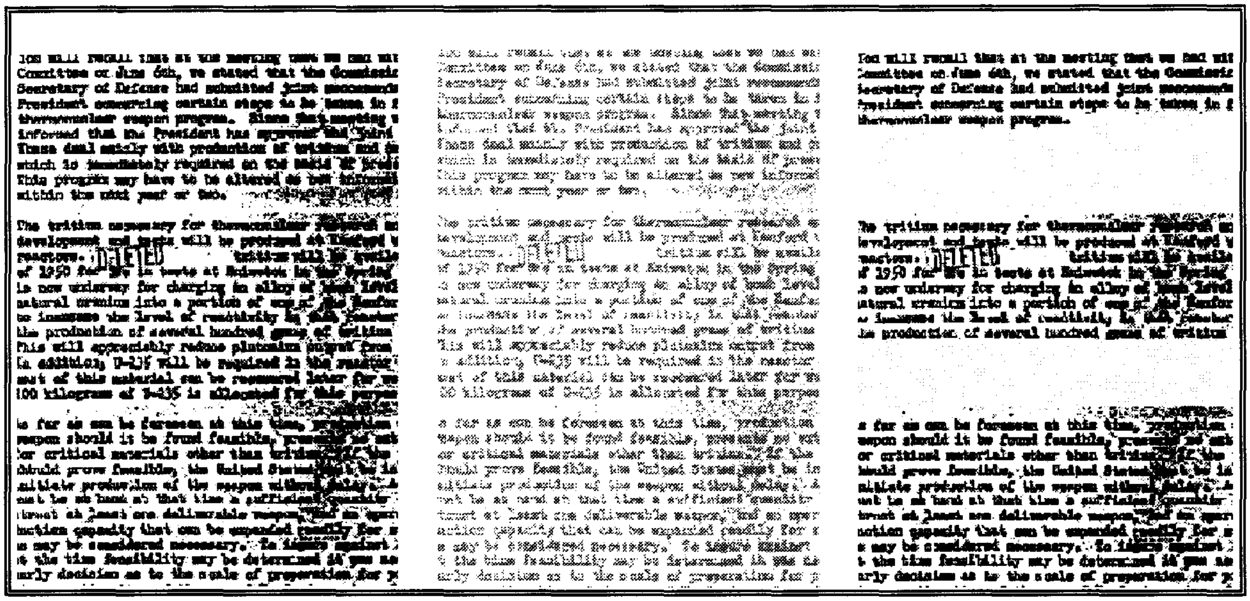
Multiple page documents can be compared in a single operation. Short documents can be stacked to produce a single "stacked page". Presence of an FCA between two stacked pages is sufficient to indicate that the stacks contain at least one pair of highly correlated pages. Conversely, the absence of an FCA allows a stack of pages to be discarded at once, thereby speeding the process of searching a database for matches.

For huge databases, the search task can be divided among multiple platforms, which each processing a subset of the database and selecting documents meeting a threshold for correlation with the target document.

Example of FlashCorrelation/Æ Processing:

The optical correlation process of FC is illustrated by the following examples.

Figure 8: Original, Light Xerox, and Partly Deleted Copies of LANL 370



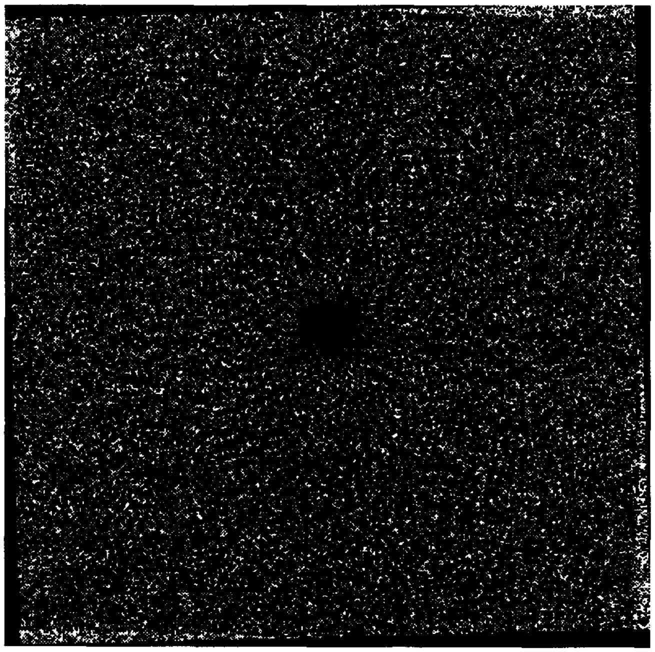
You will recall that at the meeting that we had wit
Committee on June 6th, we stated that the Commissio
Secretary of Defense had submitted joint recommendi
President concerning certain steps to be taken in t
thermonuclear weapon program. Since that meeting w
informed that the President has approved the joint
These deal mainly with production of tritium and p
which is immediately required on the basis of presen
This program may have to be altered as new informati
within the next year or two.

The tritium necessary for thermonuclear research a
development and tests will be produced at Hanford w
reactors. DELETED tritium will be availa
of 1950 for use in tests at Eniwetok in the Spring
is now underway for charging an alloy of high level
natural uranium into a portion of one of the Hanfor
to increase the level of reactivity in that reactor
the production of several hundred grams of tritium
This will appreciably reduce plutonium output from
In addition, U-235 will be required in the reactor
most of this material can be recovered later for we
100 kilograms of U-235 is allocated for this purpos

As far as can be foreseen at this time, production
weapon should it be found feasible, presents no seri
for critical materials other than tritium. If the
should prove feasible, the United States must be in
initiate production of the weapon without delay. A
must be on hand at that time a sufficient quantity
tritium at least one deliverable weapon, and an oper
duction capacity that can be expanded rapidly for s
as may be considered necessary. To hedge against a
the time feasibility may be determined if you ma
early decision as to the scale of preparation for p

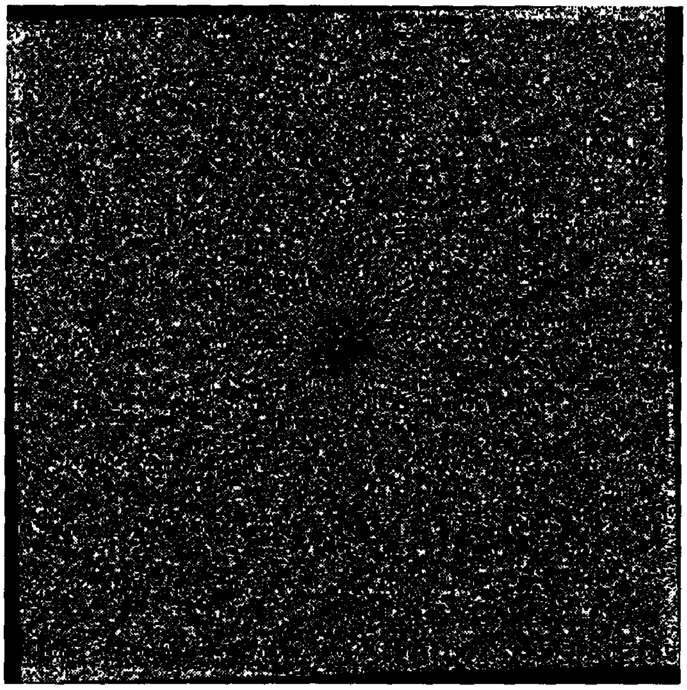
Since the location, size, and density of the FCA region is known, the correlation between two documents is derived from the density of the FCA region. The matrices which are utilized in the FC process are scrambled encodings of the original documents, produced at the time they are scanned into the database.

Figure 9: FCA produced from FC of the Original and its Light Xerox



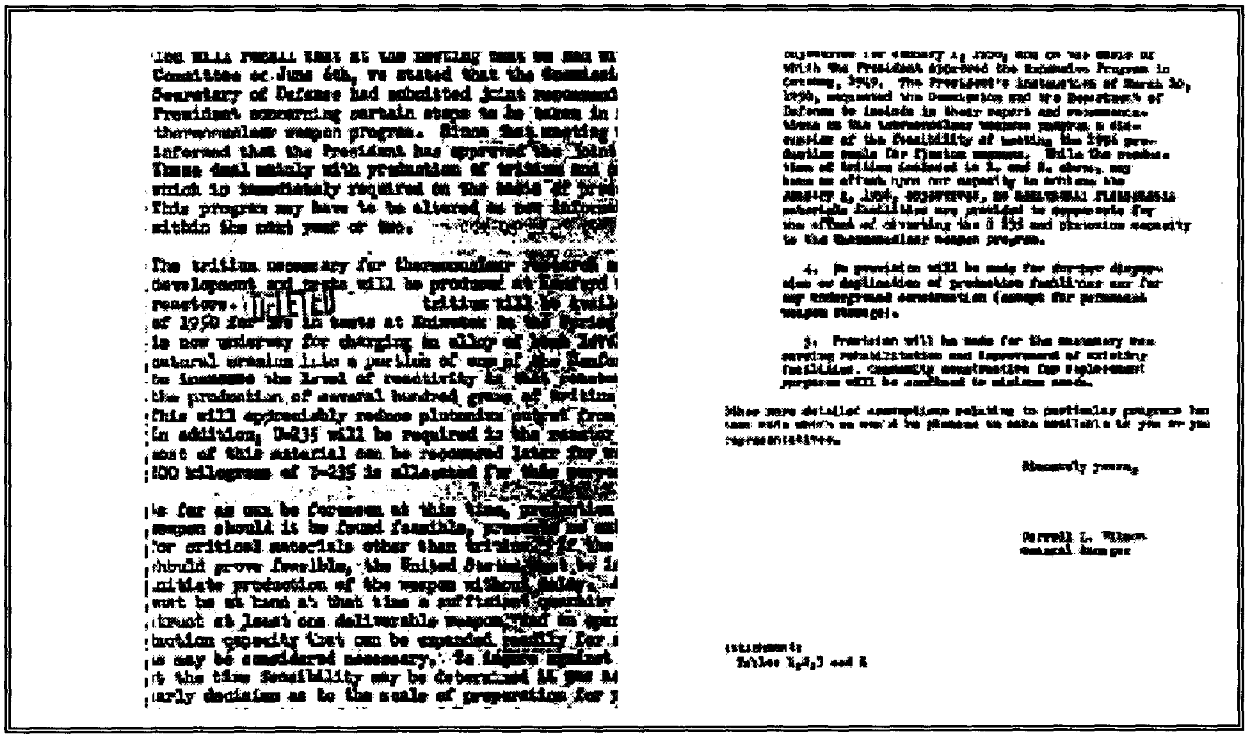
The FCA indicates a strong correlation between the two documents, leading to the Lightened Replica being listed as a Candidate Duplicate of the Original.

Figure 10: FCA Produced from the FC of Original and Partial Replica



An FCA is present, although not as strong as in the prior example. The Partial Replica will be considered a Candidate Duplicate of the Original if the density of the FCA exceeds a specified threshold. If Partial Replicas are to be considered Duplicates, then the threshold is set accordingly.

Figure 11: LANL Documents 370 and 365



[illegible] at the meeting [illegible]
Committee on June 6th, we stated that the Commissi
Secretary of Defense had submitted joint recommend
President concerning certain steps to be taken in
thermonuclear weapon program. Since that meeting
informed that the President has approved the joint
These deal mainly with production of tritium and
which is immediately required on the basis of pro
This program may have to be altered as new informa
within the next year or two.

The tritium necessary for thermonuclear research a
development and tests will be produced at Hanford
reactors. DELETED tritium will be avail
of 1950 for use in tests at Eniwetok in the Spring
is now underway for charging an alloy of [illegible]
natural uranium into a portion of one of the Hanfo
to increase the level of reactivity in that reactor
the production of several hundred grams of tritium
This will appreciably reduce plutonium output from
In addition, U-235 will be required in the reactor
most of this material can be recovered later for w
100 kilograms of U-235 is allocated for this purpo

As far as can be foreseen at this time, production
weapon should it be found feasible, presents no sub
for critical materials other than tritium. If the
should prove feasible, the United States must, to i
initiate production of the weapon without delay, a
must be at hand at that time a sufficient quantity
permit at least one deliverable weapon and an oper
duction capacity that can be expanded readily for
as may be considered necessary. To insure against
at the time feasibility may be determined it was d
early decision as to the scale of preparation for

[illegible] for January 1, 1950, and on the basis of
which the President approved the Expansion Program in
October, 1949. The President's instructions of March 10,
1950, requested the Commission and the Department of
Defense to include in their report and recommenda-
tions on the thermonuclear weapon program a dis-
cussion of the feasibility of meeting the 1954 pro-
duction goals for fission weapons. While the produc-
tion of tritium indicated in 1. and 2. above, may
have an effect upon our capacity to achieve the
January 1, 1954, objectives, no additional fissionable
materials facilities are provided to compensate for
the effect of diverting the U 235 and plutonium capacity
to the thermonuclear weapon program.

4. No provision will be made for further disper-
sion or duplication of production facilities nor for
any underground construction (except for permanent
weapon storage).

5. Provision will be made for the necessary time-
saving rehabilitation and improvement of existing
facilities. Community construction for replacement
purposes will be confined to minimum needs.

When more detailed assumptions relating to particular programs have
been made which we would be pleased to make available to you or your
representatives.

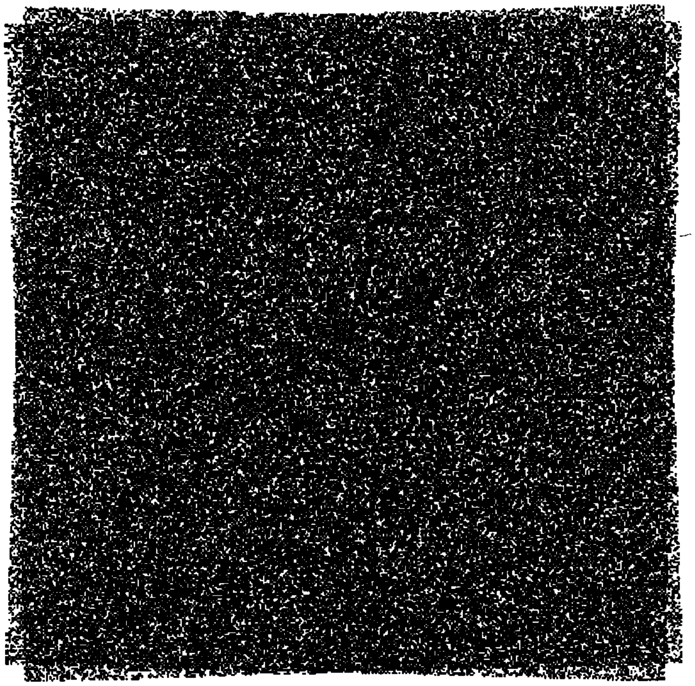
Sincerely yours,

Carroll L. Wilson
General Manager

Attachments:
Tables 1,2,3 and 4

When the FC engine encounters two different documents, as here, no FCA is produced.

Figure 12: No FCA Results from FC of Documents 370 and 365



Test Results to Date - Without Partitioning

Two tests were performed using 530 and 460 single page documents. The first set included graphics, photographs, photographs, newspaper text, and business correspondence with both greyscale and color variations. The second set was created from the LANL/DOE database of sample redacted typewritten pages, involving only bit mapped text. In both tests, the name of each document was set so that the FC system could automatically determine whether its selections of candidate duplicates were correct or not. That facilitated making many runs at different threshold settings. Initially no partitioning of the database was performed. The resulting error curves had cross-over points (where the percent of false negatives equals the percent of false positives) of about 1.5% in each test. The test sets were constructed to include skewing, over or under exposed copying, light and heav annotation, and marginalia. Selection errors were found to be due to induced skewing and overly dar or light Xeroxing. The other induced changes did not produce errors.

It is not known to what extent the resulting test sets truly represent the extent and prevalence of these effects in the actual databases to be processed. Therefore, the actual error rates to be achieved through operational deployment of FC may be higher or lower than that obtained during these initial tests. Effective deskewing with a residual error of less than 0.1 degree will essentially eliminate errors resulting from skew, which constituted half of the errors in the tests. Documents which are partl obliterated by over or under saturation can be analyzed by considering as candidate duplicates an document which has at least a certain number of cells which correlate above threshold.

Test Results to Date - With Partitioning

Partitioning of a database should improve the error rates as well as speed the process. For the test sets created, the partitioning steps described above caused only and all manipulated duplicates of the same document to occupy the same partition, meaning that partitioning alone resulted in duplicate detection with 0% errors without the need for FC. Additional testing with much larger databases would demonstrate the degree of error reduction to result from partitioning prior to FC. For operational deployment, partitioning along the lines suggested above should provide significant improvements in accuracy and speed, while requiring only trivial additional time during initial scanning of each document into the database.

Steps in Implementing FlashCorrelationô(FC) for Duplicate Document Search

FC processing can be incorporated into redaction systems, or performed on a stand-alone basis or in background mode. It can automatically partition a database and search for all duplicates within each partition, or it can be used on a routine basis to search a database of previously redacted documents to find a potential duplicate for each new document being entered into the database to be processed. The same basic processing is performed in either usage.

Current Pre-Processing

The current pre-processing of documents for redaction includes the following steps:

1. Manually create a header by keying-in originating agency, accession #, date, # pages
2. Scan capture each page at 300 dpi
3. De-skew, compress, and store as .TIFF Group 4 format
4. QA/QC and OCR with full text indexing (without QA on output)
5. Select possible duplicates from metadata and route to duplicate detector
6. Assign rest for redaction; also assign the failed possible duplicates
7. After redaction, route for seal and release

The current preprocessing time required per document is estimated to average about 30 seconds per page on Highland Technologies Incorporated processing systems. The header size is estimated to be 20 KB.

Additional Pre-Processing to Implement FC

The following additional processing is done automatically at the same time:

1. For each page, compute row sums and column sums
2. Set page origin from left and top where text starts (non-zero sums). When pages may be ver dirty, the human operator can override the automatic origin designation.
3. Divide each page into cells (8x10 at present).
4. For each cell, compute row sums
5. For each cell, total all row sums; if total is less than the "white threshold" Tw, assign a "0" to that cell in an 80-bit string representing all cells (white threshold may be greater than 0 to allow for some dirt or stray pixels).
6. For each document, produce the characterization to be stored with the metadata:

A = an array of 80 bits x number of pages showing all non-white cells as 1's.

7. For each page, total the row sums across the 8 cells and analyze them to determine the number of lines per page. A line break is defined as a near-zero row sum for at least Rb consecutive rows following a non-zero row sum for at least Rl consecutive rows. The values for Rb and Rl are determined from the resolution of the images being processed. As an example, if a single spaced page contained 60 lines of type and scanning at 300 dpi produces 3000 row sums, there would be 60 oscillations in row sums from 0 to a max and back to 0. The actual shape of the oscillation for each line would depend on the font used, the number of capital letters, and underlining. At 50 pixel rows per text line, there could be a break of 25 near-zero rows followed by 25 non-zero rows.
8. For each document, produce the additional characterization to be stored with the metadata:

B = a string of two characters per page, giving the number of lines per page in order.

9. For each document, for each page, store also with the metadata:

C = an FC array of approximately 640 x 480 bytes (which is a rotated, binarized, reduced resolution replica of the page.

The time required per page is 8.7 milliseconds. The addition to the header is 1 KB per page.

FC Search for Duplicate Documents

At any time after the preprocessing has been performed, an FC search for duplicates can be performed by a dedicated database processor, or by the preprocessing system, or by a system used for redaction. In all cases, the following automated steps apply:

1. A digitized Target Document and its characterizations A, B, C are brought into the FC processor.
2. Partition(s) of the database which match the characterizations A and B are determined.

3. For each document in the selected partition(s), the array C is loaded for each page.
4. The FCA strength of each page of each document in the partition is calculated and compared to a decision threshold.
5. The rules selected for determining a duplicate document are applied. Examples: (1) To be considered a Candidate Duplicate of the Target Document, a document must be the same number of pages as the Target, and each corresponding page must have an FCA above threshold. Or (2) A Candidate Document can be of different length than the Target, but each page of the shorter document must match a page of the longer document, that is, have an FCA higher than threshold, in order to be considered Candidate Duplicates.
6. The system outputs the identifier of Candidate Duplicates to the Target Document.

Processing Time Summar

The stated processing times resulted from conducting the tests on a Pentium II/300 PC:

1. Characterizing a page (extracting information A, B, and C) requires 8.68 milliseconds, and can be automatically done while the page is being scanned and the metadata keyed-in.

2. FlashCorrelationô of a target page and another page from the database partition (C arrays) and comparison to the threshold requires 2.52 milliseconds.

3. Applying a decision rule to multi-page documents would add probably another millisecond, depending on the complexity of the rule.

Further Considerations

The next step is to conduct a large scale test of the two-step method presented and analyze the results as to accuracy and throughput. Improvements to the method can then be considered. They ma include: Variations on the characterization features used for partitioning the database. Adaptive setting of limits to define "white cells". Use of autocorrelations for automated adaptive setting of the threshold to determine Candidate Duplicates. Further processing to reduce residual skew. Automated dynamic contrast enhancement during preprocessing.

MIKOS Ltd. would welcome the opportunity to process large databases, and refine its techniques, in cooperation with other companies or with government agencies.

References

1. *FlashCorrelation Method and Apparatus for N-dimensional Image Identification and Analysis*, U. S. Patent # 5,583,950, issued December 10, 1996, F. J. Prokoski.

2. *Patient ID and Fusion of Medical Images Using SIMCOS;* MEDTEC 96; Medical Technolog Conference, Orlando, Florida July 1996, F. J. Prokoski

3. *FlashCorrelation = "MPP Lite",* Informal presentation at The Conference on High Speed Computing, LANL/LLNL, Salishan Lodge, March-April 1993, F.J. Prokoski.

4. *High Security Tagging System for Evidence Marking and Verification*, Robert B. Riedel. Francine J. Prokoski, Jeffrey S. Coffin; Carnahan Conference on Security Technolog , Atlanta GA.. October 1992.

5. *Method and Apparatus for Identification of Individuals and their Conditions from Analysis of Elemental Shapes in Biosensor Data Represented as N-dimensional Images* , U S Patent 5,163,094, issued November 1992, F.J. Prokoski, et. al.

Duplicate Detection

String Techniques for Duplicate Document Detection

Daniel P. Lopresti
dpl@research.bell-labs.com

Bell Laboratories
Lucent Technologies, Inc.
600 Mountain Avenue, Room 2C-552
Murray Hill, NJ 07974

Abstract

Detecting duplicates in document image databases is a problem of growing importance. The task is made difficult by the various degradations suffered by printed documents, and by conflicting notions of what it means to be a "duplicate." To address these issues, this paper describes a framework for clarifying and formalizing the duplicate detection problem. Four distinct models are presented, each with a corresponding algorithm for its solution adapted from the realm of approximate string matching. The robustness of these techniques is demonstrated through a set of experiments using data derived from real-world noise sources.

1 Introduction

As information management and networking technologies continue to proliferate, databases of document images and their associated meta-data are growing rapidly in size and importance. A key problem facing such systems is determining whether duplicates already exist in the database when a new document arrives. This is challenging both because of the various ways a document can become degraded and because of the many possible interpretations of what it means to be a "duplicate."

For example, one document might be a photocopy of another, or a fax. The copies could be visually identical, or one might have additional handwritten notes appended to it. If the original document was generated on-line, a duplicate could contain exactly the same text, only formatted in a different way (changes in font, line spacings and lengths, etc.). A duplicate might possess substantially the same content, but with minor alterations due to editing (*i.e.*, earlier or later versions of the same document). Of course, in any of these cases the scanned image of either or both of the documents may contain significant "noise" due to the way the hardcopy was handled or anomalies in the scanning process. All of these interpretations are reasonable; later a framework is described for clarifying and formalizing them.

Whatever the definition, the process of determining whether one document is a duplicate of another involves two steps:

1. Extracting appropriate information (features) from the incoming document image.
2. Comparing the features against those previously extracted from documents in the database.

What features to use, and how they are compared, are the two primary issues to be resolved. Different choices lead to models which will be appropriate for different applications.

Previous work on detecting duplicates (*e.g.*, [2, 6, 7, 19]) has concentrated mostly on exploring the first step above, turning to more traditional measures when it comes to the second. In this paper, the focus is on the models and algorithms associated with comparing document representations (*i.e.*, the second step), while features are taken to be the uncorrected text output from a commercial OCR package. A framework is given for categorizing and studying different kinds of duplicates, along with algorithms that extend the range of techniques available for searching document image databases. These methods prove to be extremely robust, even in the presence of low OCR accuracies.

The remainder of this paper is organized as follows. Section 2 presents four distinct but related models for the duplicate detection problem motivated in part by the literature for approximate string matching. Each of these is solved optimally using a dynamic programming algorithm, as described in Section 3. Implementation issues are considered in Section 4. Section 5 presents experimental results that demonstrate the robustness of these techniques across models and in the presence of real-world noise. Related work is reviewed in Section 6. Finally, conclusions and possible future research directions are

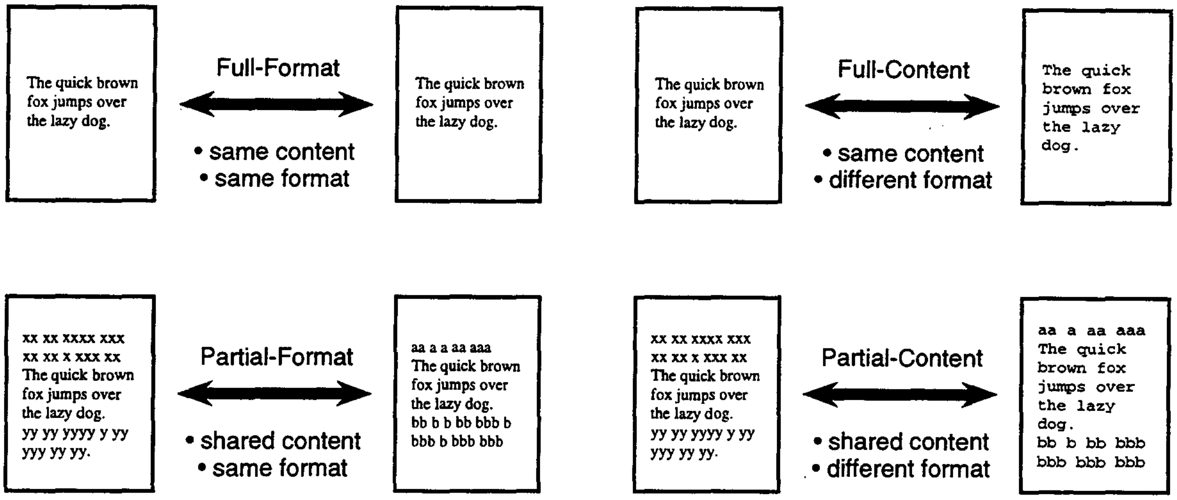

The quick brown fox jumps over the lazy dog.
Full-Format
• same content
• same format
The quick brown fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog.
Full-Content
• same content
• different format
The quick brown fox jumps over the lazy dog.
xx xx xxxx xxx xx xx x xxx xx The quick brown fox jumps over the lazy dog. yy yy yyyy y yy yyy yy yy.
Partial-Format
• shared content
• same format
aa a a aa aaa The quick brown fox jumps over the lazy dog. bb b b bb bbb b bbb b bbb bbb
xx xx xxxx xxx xx xx x xxx xx The quick brown fox jumps over the lazy dog. yy yy yyyy y yy yyy yy yy.
Partial-Content
• shared content
• different format
aa a aa aaa The quick brown fox jumps over the lazy dog. bb b bb bbb bbb bbb bbb

Figure 1: The four duplicate classes discussed in this paper.

discussed in Section 7.

2 Models

For the purposes of this paper, the assumption is that the documents of interest, while in image form, are primarily textual in content. Viewed abstractly, such a page is a series of lines, each consisting of a sequence of symbols. In this *string-of-strings* viewpoint, the term "symbol" can be defined quite liberally. It could be interpreted as meaning characters, of course, but representations at higher levels (*e.g.*, words) or lower levels (*e.g.*, basic features computed from image components) are also possible.

What, then, is a duplicate? Rather than start enumerating possibilities in an ad hoc manner, some structure can be obtained by first partitioning the problem along two dimensions: whether the duplication is full or partial, and whether the layout of text across lines is maintained or not. The reasons for this particular classification scheme are rooted in the string formalisms to be described in the next section. For now, the four possibilities are illustrated with real-world examples and to introduce the terminology:

1. If two documents are visually identical, one is a photocopy or a fax of the other, say, they are *full-format* duplicates. This category also covers documents distributed electronically (*e.g.*, as PDF or PostScript) and printed without further editing.

2. If two documents have identical textual content, but not necessarily the same formatting, they are *full-content* duplicates. This includes, for example, the same e-mail message sent to two people and printed using different-sized fonts, or an HTML document downloaded from the WWW and printed using different margin settings.

3. If two documents share significant content with the same formatting, they are *partial-format* duplicates. Exactly how long the similar regions must be will depend, in general, on the application. Two instances of this are the copy-and-pasting of whole paragraphs from one document into another, and "redacting," the editing of a hardcopy document by obscuring portions of the text so that it is no longer legible.

4. If two documents share content but their formatting is not necessarily the same, they are *partial-content* duplicates. This arises in the copy-and-pasting of individual sentences or groups of sentences. A later version of a document that has undergone several editing passes is likely to be a partial-content duplicate.

These various types of duplication are shown in Figure 1. In the next section, algorithms specialized to each of these cases are given. Note that although the text used to illustrate the figure is "clean," it will be necessary to handle a full range of document recognition errors, include characters that have been misrecognized, omitted, or added, words that have been improperly segmented, complete lines that have been missed or inserted, etc.

Before moving on, it may be instructive to consider briefly the relationships between the various kinds of duplicates. This "universe" is depicted in Figure 2, where several example data-points have also been plotted. Note that there is overlap between the classes, with partial-content duplicates encompassing all the other types.

Clearly, every format duplicate is also a content duplicate; the former is a special case of the latter.

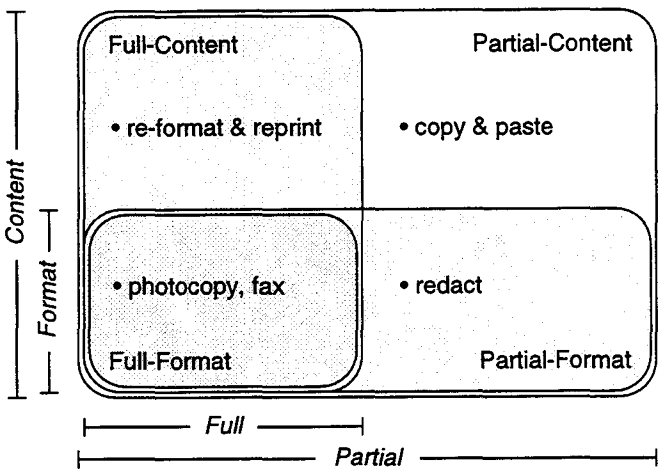


Figure 2: The universe of duplicates.

From a formal standpoint, the distinction is whether the page is treated as a 2-D stream consisting of lines made up of characters, or as a 1-D stream of characters in reading order. Note that the 2-D representation can be converted into a 1-D representation by treating the new line character as a space [19]. This implies that any algorithm for detecting content duplicates can also be used to detect format duplicates. There will undoubtedly be cases, however, where a search can be confined to, say, possible photocopies of a document. Here, an algorithm specialized to finding format duplicates will yield higher precision (*i.e.*, fewer false "hits") than the more general algorithm, which also returns potential content duplicates.

Note also that any full duplicate is also a partial duplicate. Again, there are benefits in maintaining the distinction, both in terms of retrieval precision and because the special case admits heuristics that greatly speed the computation, as is discussed in another paper [11].

3 Basic Algorithms

If it were possible to assume that OCR was perfect or nearly so, the problem of locating duplicates would be relatively straightforward. At best, this is a highly optimistic assumption. Instead, it is safer to acknowledge that OCR can be arbitrarily bad, with no specific guarantee that any n consecutive characters will come through unscathed. If, for example, the accuracy rate were 75% (a reasonable assumption in the case of faxes, small fonts, etc.) and errors are independent, the probability that a given n-gram will survive is 0.24 for $n = 5$, and 0.056 for $n = 10$. The chance that a complete sentence would make it through without errors is miniscule. Hence, schemes that depend on a majority of words or sentences being recognized correctly, while working reasonably well for clean input, may break down in the case of degraded documents.

Fortunately, the literature on approximate string matching is rich with techniques for addressing such concerns [5, 17, 20]. Moreover, the model correlates well with the physical processes that result in errors, so as a measure of similarity it is supported by intuition. Drawing from this body of work, algorithms are given for each of the four variants of duplicate detection. In the context of their respective models, all are guaranteed to return optimal solutions.

Beginning with some definitions, a *string*, $D = d_1 d_2 \ldots d_n$, is a finite sequence of symbols chosen from a finite alphabet, $d_i \in \Sigma$. String $S = s_1 s_2 \ldots s_m$ is a *substring* of string $D = d_1 d_2 \ldots d_n$ if $m \leq n$ and there exists an integer k in the range $[0, m - n]$ such that $s_i = d_{i+k}$ for $i = 1, 2, \ldots, m$. The set of all possible substrings of D is denoted D^*. In the 1-D case (*i.e.*, content duplicates), a document is simply a string. In the 2-D case (*i.e.*, format duplicates), a document is a sequence of strings, $D = D^1 D^2 \ldots D^m$ where $D^i = d^i_1 d^i_2 \ldots d^i_n$.

A standard measure for approximate string matching is provided by *edit distance* [8]. In the simplest case, the following three operations are permitted: (1) delete a symbol, (2) insert a symbol, (3) substitute one symbol for another. Each of these is assigned a cost, c_{del}, c_{ins}, and c_{sub}, and the edit distance is defined as the minimum cost of any sequence of basic operations that transforms one string into the other.

3.1 Full-Content Duplicates

As it relates to full-content duplicates, this optimization problem can be solved using a well-known dynamic programming algorithm [15, 21]. Let $Q = q_1 q_2 \ldots q_m$ be the query document, $D = d_1 d_2 \ldots d_n$ be the database document, and define $dist1_{i,j}$ to be the distance between the first i characters of Q and the first j characters of D. The initial conditions are:

$$\begin{array}{lcll} dist1_{0,0} & = & 0 & \\ dist1_{i,0} & = & dist1_{i-1,0} + c_{del}(q_i) & 1 \le i \le m \\ dist1_{0,j} & = & dist1_{0,j-1} + c_{ins}(d_j) & 1 \le j \le n \end{array} \tag{1}$$

and the main dynamic programming recurrence is:

$$dist1_{i,j} = \min \left\{ \begin{array}{lcl} dist1_{i-1,j} & + & c_{del}(q_i) \\ dist1_{i,j-1} & + & c_{ins}(d_j) \\ dist1_{i-1,j-1} & + & c_{sub}(q_i, d_j) \end{array} \right. \tag{2}$$

for $1 \le i \le m,\ 1 \le j \le n$. The computation builds a matrix of distance values working from the upper left corner ($dist1_{0,0}$) to the lower right ($dist1_{m,n}$), as illustrated in Figure 3. Once it has completed, a sequence of editing decisions that achieves the optimum can be determined via backtracking.

As indicated above, the costs in general can be a function of the symbol(s) in question. As a rule, the deletion and insertion costs are assumed to be greater than 0, while the substitution cost is greater than 0 if the symbols do not match and less than or equal to 0 if they do. In the event constant costs are used, it is convenient to refer to them as simply c_{del}, c_{ins}, and c_{sub} (when the two symbols are different) or c_{mat} (when they are the same). It is possible, and indeed sometimes desirable, to specify cost functions that are quite sophisticated. Moreover, the set of basic editing operations can be supplemented as appropriate. Both of these issues will be covered in a later section.

Algorithm *dist1* provides the basis for a solution to the full-content duplicate problem; the smaller the distance, the more similar the two documents. While OCR errors will raise this value somewhat, to the extent they are modeled by symbol deletions, insertions, and substitutions, they will be accounted for.

3.2 Partial-Content Duplicates

The previous formulation requires the two strings to be aligned in their entirety. For the partial duplicate problem, what is needed is the best match between any two substrings of Q and D. Conceptually, this corresponds to generating all substring pairs in $\{Q^* \times D^*\}$ and then comparing them using algorithm *dist1*. In practice, however, this would be too inefficient.

Fortunately, the original computation can be modified so that shorter regions of similarity can be detected in two longer documents with no increase in time complexity. The edit distance is made 0 along the first row and column of the matrix, so the initial conditions become:

$$\begin{array}{lcll} sdist1_{0,0} & = & 0 & \\ sdist1_{i,0} & = & 0 & 1 \le i \le m \\ sdist1_{0,j} & = & 0 & 1 \le j \le n \end{array} \tag{3}$$

In addition, another term is added to the inner-loop recurrence capping the maximum distance at any cell to be 0. This has the effect of allowing a match to begin at any position between the two strings. The recurrence is:

$$sdist1_{i,j} = \min \left\{ \begin{array}{lcl} 0 & & \\ sdist1_{i-1,j} & + & c_{del}(q_i) \\ sdist1_{i,j-1} & + & c_{ins}(d_j) \\ sdist1_{i-1,j-1} & + & c_{sub}(q_i, d_j) \end{array} \right. \tag{4}$$

for $1 \le i \le m,\ 1 \le j \le n$. Finally, the resulting distance matrix is searched for its smallest value. This reflects the end-point of the best substring match. The starting point can be found by tracing back the sequence of optimal editing decisions. Note there is an added requirement that the cost when two symbols match be strictly less than zero, otherwise every entry in the matrix will be 0. This computation is illustrated in Figure 4.

Algorithm *sdist1* solves the partial-content duplicate problem by computing

$$\min\{dist1(A, B) \mid A \in Q^*,\ B \in D^*\}$$

In other words, it locates the best-matching regions of similarity between the two documents Q and D. A and B, the two matching subregions, can be recovered if so desired.

3.3 Full-Format Duplicates

For the 2-D models (*i.e.*, format duplicates), another level is added to the optimization. The problem is still one of editing, but at the higher level the basic entities are now strings (lines). At the lower level, as before, they are symbols. Say that $Q = Q^1Q^2 \ldots Q^k$ and $D = D^1D^2 \ldots D^l$, where each Q^i and D^j is itself a string. For full-format duplicates, the inner-loop recurrence takes the same general form as the 1-D case:

$$dist2_{i,j} = \min \left\{ \begin{array}{lcl} dist2_{i-1,j} & + & C_{del}(Q^i) \\ dist2_{i,j-1} & + & C_{ins}(D^j) \\ dist2_{i-1,j-1} & + & C_{sub}(Q^i, D^j) \end{array} \right. \tag{5}$$

for $1 \le i \le k,\ 1 \le j \le l$, where C_{del}, C_{ins}, and C_{sub} are the costs of deleting, inserting, and substituting whole lines, respectively. The initial conditions are defined analogously to Equation 1.

Since the basic editing operations now involve full strings, it is natural to define the new costs as:

$$C_{del}(Q^i) \;\equiv\; dist1(Q^i, \phi)$$

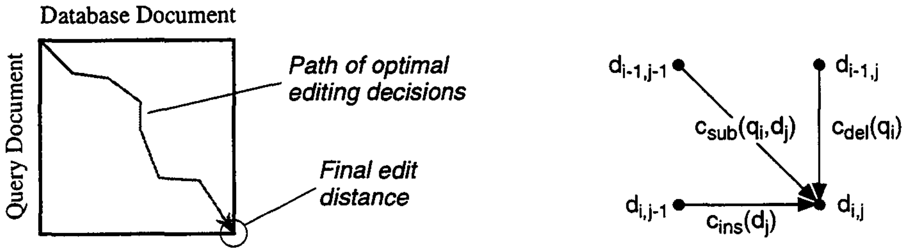


Figure 3: The basic algorithm for string edit distance (*dist1*).

$$\begin{aligned} C_{ins}(D^j) &\equiv dist1(\phi, D^j) \\ C_{sub}(Q^i, D^j) &\equiv dist1(Q^i, D^j) \end{aligned} \quad (6)$$

where ϕ is the null string. Hence, the 2-D computation is defined in terms of the 1-D computation. This is illustrated in Figure 5.

All else being equal, it can be shown that $dist2(Q, D) \geq dist1(Q, D)$ for any two documents Q and D. As noted earlier, *dist1* admits a larger class of duplicates (full-content), while *dist2* may provide higher precision for the class it is intended for (full-format).

3.4 Partial-Format Duplicates

Lastly, the extension for partial-format duplicates combines the modifications for the partial (Equation 4) and format (Equation 5) problems:

$$sdist2_{i,j} = \min \begin{cases} 0 \\ sdist2_{i-1,j} & + \quad C_{del}(Q^i) \\ sdist2_{i,j-1} & + \quad C_{ins}(D^j) \\ sdist2_{i-1,j-1} & + \quad C_{sub}(Q^i, D^j) \end{cases} \quad (7)$$

for $1 \leq i \leq k$, $1 \leq j \leq l$. Note that C_{del}, C_{ins}, and C_{sub} are defined as before in terms of *dist1* (*i.e.*, Equation 6), not in terms of the 1-D substring computation as might be expected. The granularity of this matching is whole lines. As before, the resulting matrix must be searched for its smallest value, and then traced back to find where the match starts.

At this point four different algorithms have been presented, one for each of the models described in Section 2.

4 Implementation Issues

In this section, a number of issues associated with implementing the algorithms of the previous section are addressed. The inner loops are straightforward to code. Even so, there are numerous degrees of freedom and possible extensions that, while they do not change the underlying algorithm, do alter the nature of the computation in interesting and possibly useful ways.

4.1 Input Alphabet

Generally, string algorithms are viewed as operating on character data. While this provides a natural link to the output from OCR, the algorithms are more general than this and can be used on any representation that obeys a 1-D or 2-D string model. The former views a document as a stream of symbols in reading order, where "symbol" could be any of a variety of features that might be computed from the image including characters, shape codes, word lengths, etc. The latter just adds to this a notion of lines, each a sequence of symbols, again in some reading order. The choice of which set of features to use in a given application will depend on the speed and/or robustness with which it can be computed.

4.2 Cost Assignments

The selection of an algorithm determines the editing model. However, within the context of a single algorithm, the choice of cost functions can have a significant impact. While it is fairly common for implementations of Equations 1-4 to employ constant editing costs, the general way in which the algorithms are formulated is much more powerful than this.

To illustrate, consider the question of whitespace errors which are common in OCR. By setting $c_{del}(sp) = c_{ins}(sp) = 0$, in effect not charging for such events, unimportant differences between two OCR'ed versions of the same documents can be ignored. Through an appropriate choice of cost functions, the distinction between various input representations is also eliminated. For example, characters and shape codes will yield identical results if the cost of character substitutions is determined based on shape code classes (*e.g.*, $c_{sub}(q_i, d_j) = 0$ for $q_i, d_j \in \{g, p, q, y\}$, the set of descender characters).

If the distribution of the OCR errors can be estimated *a priori* (*e.g.*, via a confusion matrix), this can be exploited by setting the editing costs to be inversely proportional to the frequencies of the error patterns in question. So, for example, if the substitution $e \rightarrow c$ is ten times more likely to occur than $M \rightarrow W$, its cost is made one tenth as much. This will yield a more sensitive comparison; values closer

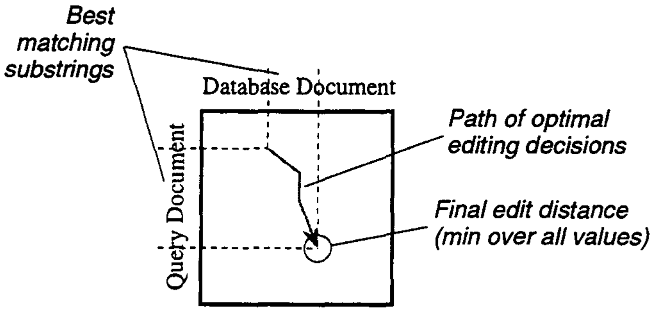

Best matching substrings
Database Document
Query Document
Path of optimal editing decisions
Final edit distance (min over all values)

Figure 4: The substring algorithm for edit distance (*sdist1*).

to the minimum when the documents are indeed duplicates under the model in question (differences due to common OCR errors), and further away from the minimum when they are not (true differences).

4.3 New Editing Operations

While the three basic editing operations (deletion, insertion, and substitution) are sufficient to capture all possible differences between two strings, the set can be supplemented with more sophisticated operations to better model an underlying error process. In the case of OCR, it may be desirable to add "split" and "merge" operations to account for mistakes in symbol segmentation [3]. The recurrence for *dist1*, for example, would then become:

$$dist1_{i,j} = \min \begin{cases} dist1_{i-1,j} & + \ c_{del}(q_i) \\ dist1_{i,j-1} & + \ c_{ins}(d_j) \\ dist1_{i-1,j-1} & + \ c_{sub}(q_i, d_j) \\ dist1_{i-1,j-2} & + \ c_{split}(q_i, d_{j-1}d_j) \\ dist1_{i-2,j-1} & + \ c_{merge}(q_{i-1}q_i, d_j) \end{cases} \tag{8}$$

for $1 \leq i \leq m,\ 1 \leq j \leq n$.

Other operations such as transpositions can also be supported. In general, as long as the number of symbols involved (the "look-back") is bounded, the recurrence can be augmented without changing the computational complexity of the algorithm.

4.4 Normalization

For exact duplicates, the distance returned by any of the four algorithms of Section 3 will either be 0 or a negative number that grows smaller as the lengths of the documents increase. For dissimilar documents, the maximum distance grows larger as the lengths increase. It is always the case that, for a given query, a smaller distance corresponds to a better match. In order for the results for different queries to be comparable, however, it is necessary to normalize the distances.

If the target interval is $[0,1]$, where 0 represents a perfect match and 1 a complete mismatch, then the following formula provides an appropriate mapping:

$$normdist = \frac{dist - mindist}{maxdist - mindist} \tag{9}$$

where *mindist* and *maxdist* are, respectively, the minimum and maximum possible distances for the comparison in question.

Assuming a full-duplicate computation, and making certain reasonable assumptions about the cost functions, the minimum is obtained when all of the characters in the query match the database document and there are no extra, unmatched characters. If the query is $Q = q_1q_2 \ldots q_m$, then:

$$mindist = \sum_{i=1}^{m} c_{sub}(q_i, q_i) \tag{10}$$

Or, more simply, $mindist = m \cdot c_{mat}$ when the costs are constant.

The maximum distance, on the other hand, is determined by the query and the set of all strings with the same length as the database document. If the cost functions are unconstrained, this in itself becomes an optimization problem. Fortunately, for constant costs there is a simple closed-form solution. Without loss of generality, let the query be the shorter of the two strings (*i.e.*, $m \leq n$). There are two possible "worst-case" scenarios: either all of the symbols of the query are substituted and the remaining symbols of the database string are inserted, or all of the query symbols are deleted and the entire database string is inserted. Thus:

$$maxdist = \min \begin{cases} m \cdot c_{sub} + (n-m) \cdot c_{ins} \\ m \cdot c_{del} + n \cdot c_{ins} \end{cases} \tag{11}$$

The partial-duplicate computations are normalized similarly.

4.5 Searching Databases

The algorithms given earlier are phrased in terms of quantifying the similarity between strings (documents). The problem of searching a database for duplicates can be cast in two ways:

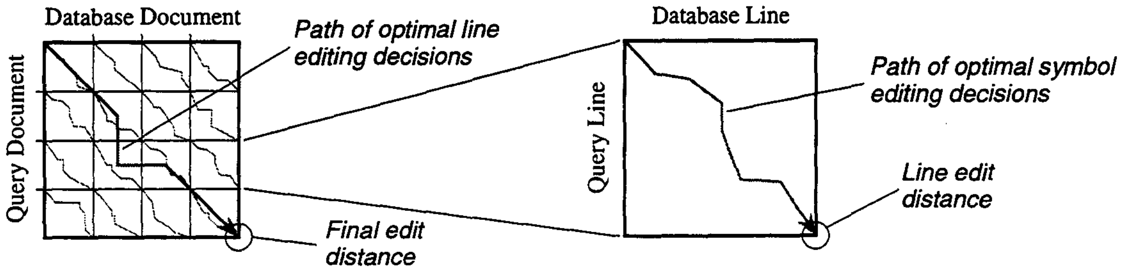


Figure 5: The 2-D algorithm for edit distance (*dist2*).

1. Return the top n matches (in ranked order).
2. Return all documents with distances below a threshold τ.

Note that the first of these requires the computation to complete before any results can be returned to the user. The second can report potential matches as they are encountered (and therefore hide some of the computational latency), but requires setting a threshold in advance. Both policies employ edit distance as a subroutine, and hence can make use of the techniques described to this point.

4.6 Speeding Things Up

Algorithms *dist1*, *sdist1*, *dist2*, and *sdist2* are optimal in the sense they return min-cost solutions to their respective problems. All require time proportional to the product of the lengths of the two documents being compared. In situations where the resulting database search is too slow, there are a variety of ways to speed things up. These include:

- Computing edit distance faster.
- Avoiding having to compute edit distance for every document in the database.
- Computing an approximation to edit distance.

These approaches can, of course, be used in combination.

Asymptotically faster algorithms and parallel VLSI architectures (*e.g.*, [9]) fall in the first category. Database indexing techniques occupy the second. The third is represented by a well-known heuristic based on the observation that, if two strings are similar, the path of optimal editing decisions must remain near the main diagonal (recall Figure 3). Hence, the computation can be restricted to a band close to the diagonal. Should the edit distance fall below some threshold as determined by the width of the band, the heuristic will return its true value, otherwise it returns a value possibly greater than the true distance (as a path other than the optimal has been chosen). This basic concept, illustrated in Figure 6, has been exploited to speed up the computation in the fields of speech recognition [16] and molecular biology [4].

Note that this heuristic applies only in the case of the full-duplicate versions of the problem, as it assumes the optimal editing path starts at $(0,0)$ and ends at (m,n). It can be shown, however, that this approach will never miss a duplicate that would have been returned by the slower, optimal algorithms.

Several new techniques for obtaining substantial speed-ups (up to two orders of magnitude) for which similar proofs-of-correctness can be given are presented elsewhere [11].

5 Experimental Results

To investigate the performance of the algorithms described in this paper, two sets of experiments were designed to explore different aspects of the problem space. The first examined duplicate detection in the presence of several real-world noise sources, while the second studied the four duplicate models and algorithms and how they relate.

For reasons of convenience, the same database was used as in previous retrieval experiments [10, 13]. This consisted of 1,000 professionally written news articles collected from Usenet. The shortest document was 364 characters long, the longest 8,626, and the average 2,974. Hence, the total size of the database was approximately 3 megabytes.

The database was used as-is (*i.e.*, no attempt was made to inject OCR errors, either real or synthetic). The query documents, however, and the intended duplicates were all "authentic": pages that had been printed, scanned, and OCR'ed. These documents were formatted in 11-point Times font with a 13-point line spacing using Microsoft Word. Each page was printed on one of two laserprinters, subjected to a noise source in most cases, scanned at 300 dpi using a UMAX Astra 1200S scanner, and then OCR'ed with Caere OmniPage Limited Edition.

For the full-duplicate computations, the edit costs were set to be $c_{del} = c_{ins} = c_{sub} = 1$ and $c_{mat} = 0$. For the partial-duplicate computations, the match cost was $c_{mat} = -1$. The study of more complex

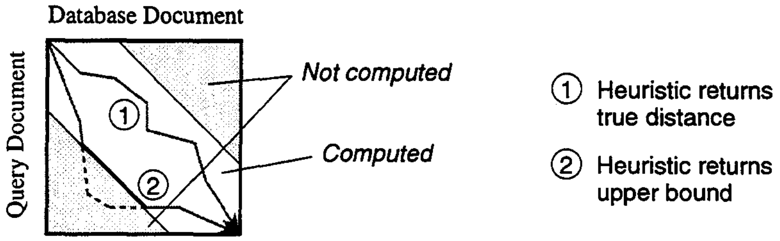

Database Document
Query Document
1
2
Not computed
Computed
① Heuristic returns true distance
② Heuristic returns upper bound

Figure 6: A heuristic for string edit distance.

costs assignments (*e.g.*, those based on confusion matrices) is left to a future paper.

5.1 Experiment 1

The goal of this experiment was to study duplicate detection under various noise conditions: copier degradations (multiple generations, excessively light or dark), faxing, and handwritten mark-up (redaction). The source document was 1,395 characters long (26 lines, 203 words). Two sets of six pages were created, one set to be inserted into the database as the intended duplicates, and the other to serve as the queries. The first set was printed on an HP LaserJet 4MPlus laserprinter, the second on an HP LaserJet 4MV. Within each set, one page was used as-is and the others were subjected to one of five different noise sources:

Faxed The page was faxed in standard mode from a Xerox Telecopier 7020 fax machine to a Xerox 7042.

3rd Generation The page was copied to the third generation on a Xerox 5034 copier.

Light The page was copied on the same copier with the contrast set to the lightest possible setting.

Dark The page was copied with the contrast set to the darkest possible setting.

Annotated Five separate text lines on the page were completely obscured using a thick blue marker pen. Different lines were excised in the query and database documents. Also, "This is important!" was handwritten in the margin.

The pages were then scanned and OCR'ed. In addition, the original ASCII text for the query document was left in the database. Hence, each of six queries was run against a database of 1,000 documents containing seven intended duplicates (six that had been OCR'ed, plus the original).

Table 1 below shows the OCR accuracies. Note that the rates range widely, dropping as low as 73.5%. While the two different versions from the same noise source are usually fairly close, they are by no means identical. As expected, a large variety of OCR errors were encountered. Beyond this, other kinds of degradations arose as well. For example, the standard headers prepended to faxes were transcribed (albeit with numerous mistakes), and the lines that had been crossed-out were completely missing from the annotated pages.

Table 1: OCR accuracies for Experiment 1.

	OCR Accuracy	
Document Type	Database	Query
OCR	96.2%	96.0%
Faxed	77.7%	83.9%
3rd Generation	95.9%	96.1%
Light	86.1%	77.8%
Dark	94.0%	95.3%
Annotated	75.6%	73.5%

Since the query documents and their intended matches have the same format, this is a full-format duplicate detection problem and the *dist2* algorithm is most appropriate. The charts in Figures 7-12 plot, for each query, the normalized edit distance for every document in the database. Note that there is always a clear distinction between true duplicates and everything else. This demonstrates that the technique is robust when faced with the sorts of OCR errors seen in practice.

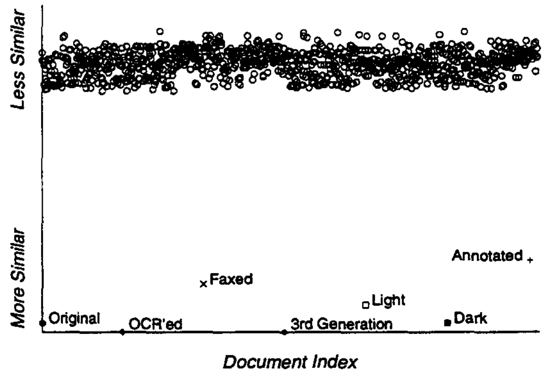


Figure 7: Full-format detection for OCR'ed query.

Studying the data further, it should come as no surprise that the annotated documents yielded the worst-case scenario. Recall that about 20% of the text was completely obscured, a figure that places

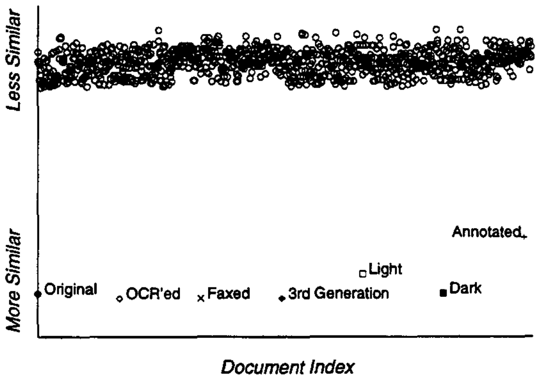


Figure 8: Full-format detection for faxed query.

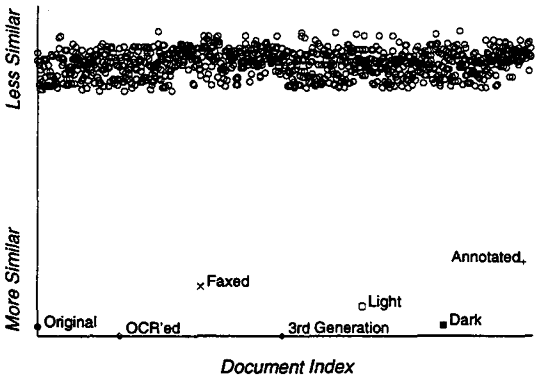


Figure 9: Full-format detection for 3rd generation query.

severe constraints on the performance of any comparison measure. Still, the normalized edit distance in most of the charts is not much greater than this value. When the annotated documents were compared to each other (Figure 12), the amount of text missing between the two amounted to 40%. Even so, and despite all the other OCR errors that must have occurred, it is possible to distinguish the duplicates from non-duplicates.

It is also interesting to note that query and database documents produced using the same noise source are usually a slightly better match (the notable exception being the case of the annotated pages). Whether it is possible to exploit this is a topic for future research.

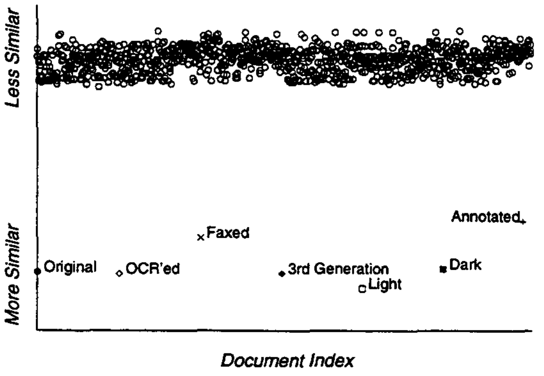


Figure 10: Full-format detection for light query.

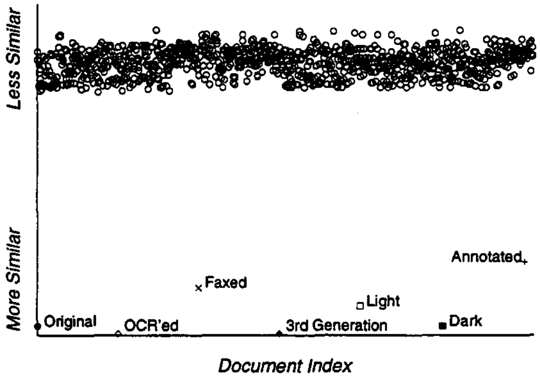


Figure 11: Full-format detection for dark query.

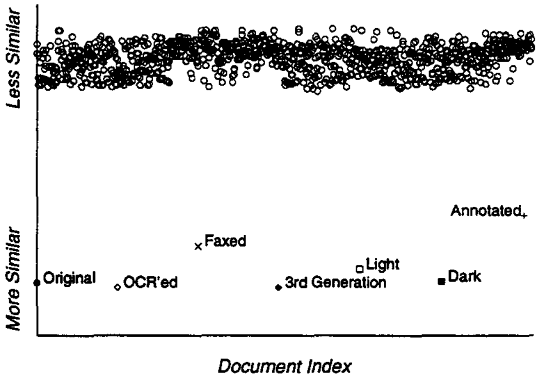


Figure 12: Full-format detection for annotated query.

5.2 Experiment 2

The purpose of this experiment was to determine how the different duplicate models relate empirically. The four algorithms described in Section 3 were run using the same source document as in the previous experiment. Duplicates were constructed from the query by:

1. Changing the line breaks to create a document that was a full-content duplicate but not a full-format duplicate.

2. Appending roughly equal amounts of unrelated text to the beginning and end of the document to create a partial-format duplicate approximately twice as long as the original.

3. Combining these first two steps to create a partial-content duplicate.

The pages were then printed, scanned, and OCR'ed. The OCR accuracies appear in Table 2. As before, the original source text was left in the database to serve as a second full-format duplicate of the query. Hence, there were between two and five duplicates in the database, depending on the model.

Table 2: OCR accuracies for Experiment 2.

Document Type	OCR Accuracy	
	Database	Query
Full-format	96.0%	95.9%
Full-content	96.1%	n/a
Partial-format	94.9%	n/a
Partial-content	96.0%	n/a

The results for this experiment are shown in Figures 13-16. Since there is a fair amount of residual similarity even in the non-matching cases, the normalized edit distances are lower than for purely random documents. Note that, as expected, algorithm *dist2* works best for full-format duplicates, and *dist1* adds to this full-content duplicates (Figures 13 and 14). The partial-format algorithm *sdist2* can detect full- and partial-format duplicates, while *sdist1* covers all four duplicate classes (Figures 15 and 16).

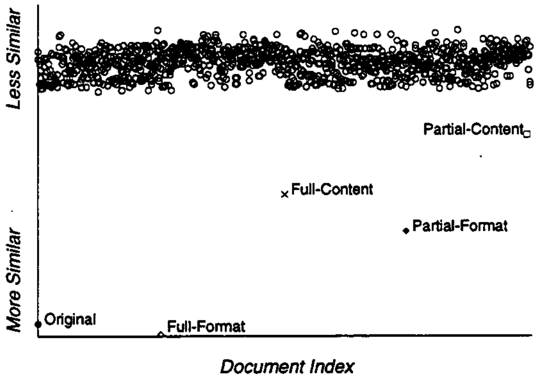


Figure 13: Duplicate detection using *dist2*.

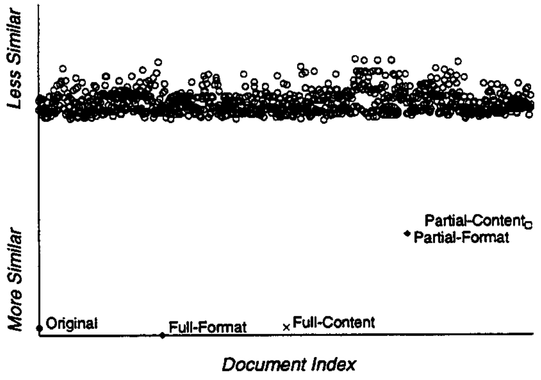


Figure 14: Duplicate detection using *dist1*.

6 Related Work

For the most part, past work on the subject has concentrated on identifying which features to extract (the first step mentioned in Section 1) and not so much on the different ways they might be compared (the second step). The latter is typically handled

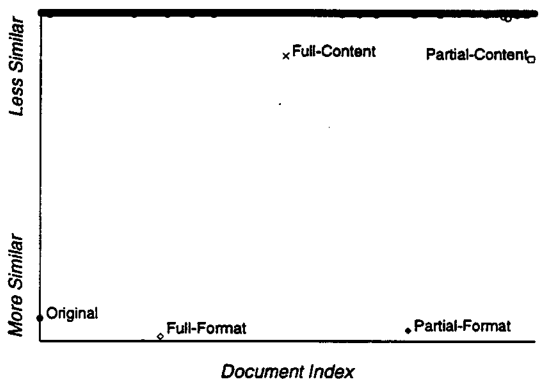


Figure 15: Duplicate detection using *sdist2*.

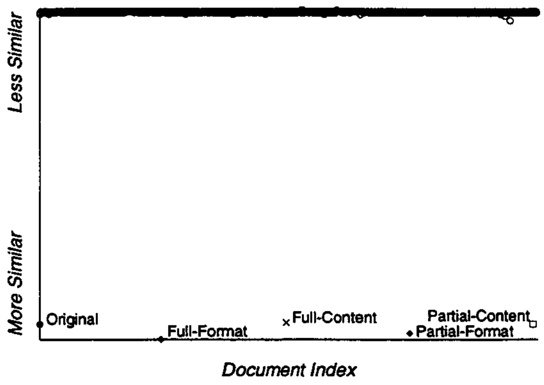


Figure 16: Duplicate detection using *sdist1*.

using one or another of the techniques from the literature.

Spitz, for example, employs character shape codes as features and compares them using the standard string matching algorithm (*i.e.*, Equation 1) [19]. In the taxonomy presented in Section 2, this corresponds to the full-content problem. Doermann, et al., also use shape codes, but extract n-grams for a specific text line to index into a table of document pointers [2]. Since this signature is computed from a single line, it does not explicitly measure the similarity of complete pages. The intention, though, is that this is a method for addressing the full-format problem. Hull, et al., describe three techniques: one based on decomposing the page into a grid and counting connected components within each cell, another using word lengths as a hash key, and one comparing image features (pass codes arising from fax compression) under a Hausdorff distance measure [7]. More details on the last method appear in [6]. The first and third of these fall in the full-format category, while the second can be classified as searching for full-content duplicates.

Also seemingly related is the general copy detection problem. There are significant differences, however, owing to the noise effects suffered by printed pages and the OCR errors they induce. Methods predicated on finding long strings of perfect similarity may not work as reliably in practice when noisy documents are included in the database. Some of the better-known schemes in this category include COPS [1] which is sentence-based, SCAM [18] which is word-based, and various algorithms for searching by computing checksums in predetermined "windows" [14].

7 Conclusions and Future Research

This paper has examined a number of issues related to the detection of duplicates in document image databases. Four distinct models for formalizing the problem were presented, along with algorithms for determining the optimal solution in each case. Experimental results demonstrate that the models match the real world, and the algorithms are robust with respect to the kinds of OCR errors that are likely to be encountered. Table 3 enumerates these classes one last time. A solid dot highlights the algorithm most suited to a particular problem, while a hollow dot indicates that the algorithm will find not only such duplicates but other types as well.

Since some of the problems seem to subsume others, an obvious question is "Why bother with the less general ones?" The answer lies in increased precision for those situations where admitting a larger class of duplicates is undesirable (*e.g.*, when the targeted duplicates are known to be photocopies). Special cases also make it possible to develop more efficient algorithms.

There are numerous ways this work could be extended. For example, there exists yet another model for approximate string matching known as "word-spotting" that applies when one of the strings must be matched in its entirety and the other is allowed the flexibility of choosing its most similar substring. This might arise when a paragraph is copied out of one document and used to query the database for other pages that contain it. Again, there is a dynamic programming algorithm along the lines of Equations 2 and 4 that solves the problem. Although the *sdist* algorithms can also catch such duplicates, they do so at a potentially lower precision.

Finally, there may be advantages to adding more levels to the symbol/line hierarchy. This could include text blocks as a collection of lines, columns as a collection of text blocks, and pages as a collection of columns. These would add new dimensions to the optimization problem, but the techniques already discussed may be generalizable. The most serious issue appears to be the requirement the system follow a unidirectional editing process at each level. Allowing arbitrary block motion overcomes this, however, and is addressed in another paper [12].

Acknowledgements

The trademarks mentioned in this paper are the properties of their respective companies.

References

[1] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the ACM SIGMOD Annual Conference*, San Francisco, CA, May 1995.

[2] D. Doermann, H. Li, and O. Kia. The detection of duplicates in document image databases. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 314–318, Ulm, Germany, August 1997.

[3] J. Esakov, D. P. Lopresti, and J. S. Sandberg. Classification and distribution of optical character recognition errors. In *Proceedings of Document Recognition (IS&T/SPIE Electronic Imaging)*, pages 204–216, San Jose, CA, February 1994.

[4] J. W. Fickett. Fast optimal alignment. *Nucleic Acids Research*, 12(1):175–179, 1984.

[5] D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, Cambridge, UK, 1997.

[6] J. J. Hull. Document image similarity and equivalence detection. *International Journal on Document Analysis and Recognition*, 1(1):37–42, February 1998.

[7] J. J. Hull, J. Cullen, and M. Peairs. Document image matching and retrieval techniques. In *Proceedings of the Symposium on Document Image Understanding Technology*, pages 31–35, Annapolis, MD, April-May 1997.

[8] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1966.

[9] R. J. Lipton and D. P. Lopresti. A systolic array for rapid string comparison. In H. Fuchs, editor, *Proceedings of the 1985 Chapel Hill Conference on Very Large Scale Integration*, pages 363–376. Computer Science Press, 1985.

[10] D. Lopresti. Robust retrieval of noisy text. In *Proceedings of the Third Forum on Research and Advances in Digital Libraries*, pages 76–85, Washington, DC, May 1996.

[11] D. Lopresti. String techniques for detecting duplicates in document databases. Submitted for publication, 1999.

Table 3: The algorithms and where they apply.

Duplicate		Algorithm			
Type	Examples	*dist2*	*dist1*	*sdist2*	*sdist1*
Full-format	photocopies, faxes	●	○	○	○
Full-content	printed HTML		●		○
Partial-format	redaction			●	○
Partial-content	copy-and-paste				●

[12] D. Lopresti and A. Tomkins. Block edit models for approximate string matching. *Theoretical Computer Science*, (181):159–179, 1997.

[13] D. Lopresti and J. Zhou. Retrieval strategies for noisy text. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 255–269, Las Vegas, NV, April 1996.

[14] U. Manber. Finding similar files in a large file system. In *Proceedings of USENIX*, pages 1–10, San Francisco, CA, January 1994.

[15] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino-acid sequences of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[16] L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson. Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-26(6):575–582, December 1978.

[17] D. Sankoff and J. B. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, MA, 1983.

[18] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In *Proceedings of the Second International Conference on Theory and Practice of Digital Libraries*, Austin, TX, 1995.

[19] A. L. Spitz. Duplicate document detection. In *Proceedings of Document Recognition IV (IS&T/SPIE Electronic Imaging)*, pages 88–94, San Jose, CA, February 1997.

[20] G. A. Stephen. *String Searching Algorithms.* World Scientific, Singapore, 1994.

[21] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21:168–173, 1974.

The 4 by 4 Duplicate Document Detection (D3) Formalism

Dr. Paul S. Prueitt
Senior Scientist, NetBase Corporation
and Director, BCN Group Inc.

Abstract

A simple formalism for duplicate detection is presented. The formalism supports the recording of human judgments, in the form of scoring, and the combination of scores with other computational evaluations to produce a class of potential duplicates based on similarity of substructure and content.

1.0: Distinction between duplicate and near-duplicate

A relationship between duplicates and near duplicates reflects a principle at the core of similarity theory. As one shifts from the question of exact matches to similarity, we are forced to consider *relationships between the parts of an object and the parts of other objects*. The analysis becomes bi-level; substructure to function or property of wholes. As such, the analysis may be readily grounded in both the neuropsychology expressed in [1] and the Mill's logic expressed in [2].

This bi-level architecture leaves out the annotation of context that is addressed in Tonfoni's paper in these proceedings [3] and is explained in some detail in [4].

1.1: Exact match is the most simple judgment. This judgment implicitly makes a judgment about what is not an exact match.

exact |——————————| different

There is only one dimension to simple judgments, with a linear scale between exact and different. The endpoints of this scale can be made to be complementary so that something is different to the degree that is not exactly the same and vice versa. So, for simple judgments, if 1/3 is the measure of exactness, then 2/3 should be the measure of difference.

1.2: Complex computational judgments allows the notion of similarity to come in between the notion of exact match and difference.

exact |————(- similar -)————| different

Measures of similarity must exist at a higher dimension than a simple linear scale between exact and different because there are different types of and expressions about similarity.

Near duplication is a special type of similarity, where some part of the "substructure" of the unit is the same and the other part of the substructure is related through some known means of intentional modification. *This technical distinction partitions the class of similarity into near-exact and non-near-exact similarity.*

exact |—–(- near–exact and similar -)—| different

Other types of similarity exist. In fact the introduction of similarity analysis leads to the question; "In what ways are these two things similar?"

exact | —–(- classes of similarity -)—–| different

Near-duplication is just one of these ways, but the metric for near duplication has a transitive property that is important to automated checking procedures.

2.0: Definition of duplicate and near-duplicate

Duplicate:

> We define two objects to be ***duplicates*** if they are exactly the same object, except one may be a copy, or both copies of some perhaps unknown original, with no intentional alterations. Alterations made by a copy machine or fax transmission are "un-intentional" and thus not the cause of a near duplicate.
>
> Corollary: Two page images may have quite different pixel structures and yet the pages may be judged exact duplicates.

Near-duplicate:

We define two objects to be ***near - duplicates*** if they satisfy one of two cases

> *In the first case*, one of the pairs has been produced from the other by some specific intentional alteration such as the addition of a signature.
>
> *In the second case*, both elements have been produced by an intentional alteration from a common, perhaps unknown, parent.

The formalism for near duplicates includes an essential dependency on information from the substructure of a unit. For near duplicate measurement the physical layout of the units being compared must be in some type of correspondence.

Using this correspondence, part of each must be duplicate and the other parts derivative from each other or a common "parent".

3.0: Four metrics:

{ exact, near-exact, non-near exact, different }

We have proposed four metrics or similarity measures about exact duplication, near-duplication, non-duplicate similarity, and difference.

The judgments are given as a ordered 4-tupil,

$$(a_1, a_2, a_3, a_4)$$

where

> a_1 is a percentage of the surface area which is an exact match,
>
> a_2 is a percentage of the surface area at is near-duplicate,
>
> a_3 is a percentage of the surface area that has non near-duplicate similarity,
>
> a_4 is a percentage of the surface area that is judged to be different.

Because of the partition between not near duplicate similarity and near duplicate, the sum of these four numbers will be 1.

4.0: Four levels:

{ page segment, page, document, collection }

If we restrict our attention to the layout structure of documents, and pages; then we have a representation of units that depends only on physical characteristics of the unit. These representations can be found at each of four levels of organization { page segment, page, document, collection }. This is called ***a 4 by 4 notation***, since there are four levels of organization and four types of judgments.

There are four types of judgments at each level. Within each level, a base unit is selected and related to any other unit of the same level. Cross level relationships are treated in a formal fashion.

The pairwise comparison in the context of large document collections is a critical issue that is addressed by effectively partitioning the search space using coding theory, in reverse, and smart hashing methods. The voting procedure [5], combined with lights out data mining techniques, can compute the necessary comparisons when some kernel of human judgments have been made to provide exemplars and training for adaptive technologies.

4.1: We can "push down" or "pull up", across levels, viewing the unit or the physical partition of the unit in ways that separates the parts or aggregates a measure of similarity into higher organizational units. For example, a document can be *pushed down* to the page level. A set of documents can be *pulled up* to the collection level.

4.2: In the case of semantic substructure, which is *not* addressed in the 4 by 4 D3 formalism, we have a number of very difficult issues, perhaps the first being about what we mean by "substructure". Surely, the answer is we mean "thematic substructure", but software does not exist that reliably computes thematic substructure. Given thematic substructure we have generalized the 4 by 4 D3 formalism to stratified similarity analysis in such a way that D3 is a special case.

5.0: Ways to partition and score D3 analytic processes

5.1: Partition: The contents of image/text collections **C** can reasonably viewed at four levels:

collections $\{C_i\}$, documents $\{d_i\}$, pages $\{p_i\}$, and page segments $\{s_i\}$.

At each level we may produce a *disjunctive* partition so that

$$\mathbf{C} = \cup\{C_i\} \quad \text{and } C_j \cap C_i = \{\} \text{ for all pairs}$$

$$\mathbf{C} = \cup\{d_i\} \quad \text{and } d_j \cap d_i = \{\} \text{ for all pairs}$$

$$\mathbf{C} = \cup\{p_i\} \quad \text{and } p_j \cap p_i = \{\} \text{ for all pairs}$$

$$\mathbf{C} = \cup\{s_i\} \quad \text{and } s_j \cap s_i = \{\} \text{ for all pairs}$$

This means that the entire contents of **C** can be thought of, and viewed from the point of view of database management, as 1) a set of collections, 2) a set of documents, 3) a set of pages, <u>or</u> 4) a set of page segments.

5.1: Scoring: Scoring can be interpreted as a judgment about the probability that a unit is correctly placed, as a retrieval element, into a category.

A specific ***"scoring algorithm"*** produces a parameterized decision (as standard scoring output) regarding whether or not each element in the full collection should be placed into a ***"similarity pool"***.

Scoring algorithms will always depend for input on the output of representational algorithms, or some transformation of the standard output type representational sets.

Reliability measures are one means to estimate the validity of a score.

6.0: Notation for combining scores

Let $\{ a_i \mid i = 1, \ldots, m \}$ be m scoring algorithms that have standard inputs the same as the standard output of one or more of the representational algorithms.

If we let s, a representational set, be a standard input, then

$$a_i(s) = k \text{ where k is a score}$$

Now suppose we have a system of rules to normalize scores to have values in the interval [0,1]. We need a system of rules since only rules of thumb have been discovered so far for the leading algorithms. The system of rules, S, maps a score to a judgment value. The mapping S has parameters for the representational method, M, the representational set and scoring algorithm, and for any additional transformations, T, that may occur using standard algorithms.

$$S(k, M, a_i(s), T) = x_i$$

where x is between 0 and 1.

A well known algorithm uses additivity.

$$d' = \Sigma\ w_i\ x_i$$

However, this algorithm is not always bounded between 0 and 1, even when the x_i s are, and must be re-normalized in order to establish some type of standard interpretation of the results. This has turned out to be non-trivial. If we combine only two scores, a_i and a_j, then the combination $k(a_i) + (1-k)\ a_j$ is bounded between 0 and 1, if k is between 0 and 1. If k is ½ then this is the average. Natural extensions of this combination depends strongly on the order of combination and is thus not unique.

In general, a literature review shows that various types of weighted averages can be used. However, the more interesting methods are those that use game theory, the language of smart agents, or distributed collaborative / cooperative decision making.

An ideal combined score algorithm would allow a flexible adjustment of the weights $\{ w_i \}$. One way to do this is through smart "query expansion", where normalization is made dependant on adaptive linkage to thesaurus and situational context. The adaptive linkage is by nature bi-level since the relationships of interest are such things as the co-occurrence of substructural invariance, as in n-gram analysis.

7.0: Cross level computation of measurement and judgments

A cross level judgment or measurement occurs when the data from one physical level, { segments, pages, documents, collections }; is propagated upward to produce a judgment or measurement at a higher physical level.

Example:

> Suppose that a one page memo is contained in a larger document, d_b. The one page memo is a document (as well as a page).

> Do we say that the document d_b is a duplicate of the memo? The answer is clearly no. We want to say that the page is contained in d_b and is a duplicate of the entire memo, and somehow convey information at the document level to say that the memo and d_b have a degree of duplication. The same set of issues apply to judgments about near-duplicate, similar, and different units.
>
> Some standard metadata such as ascension numbers, dates or titles sometimes exist for the whole document, and is useful as an informational filter. Those documents that pass through the filter can be compared to representational aggregates or storied queries.

Storage of judgments made: Human judgments about similarity can be stored so that cross level judgments can be computed with the 4 by 4 D3 formalism. For example, a document to document comparison can be computed (without human intervention) if all pairs of pages have a similarity measure. Likewise the similarity relationships between collections can be studied if document to document judgments have been stored.

Computed judgments: Each document will have a collection of words with a frequency of occurrence, for each token, across the entire document. This signature may be incomplete due to poor OCR, but will be useful for statistical retrieval methods, even with partial OCR, based on well understood methods. Moreover, it is a document level signature. However, within the document there may be several page segments that have good OCR and that maybe used to develop an OCR and/or full text (word and stemmed word frequency) representations for the complete document.

Reliability: Scoring algorithms can be combined, composed with variations on representational methods, and similarity metrics and scoring can become adaptive to either user input or algorithmic processes.

8.0: Moving from one level to the next level

Suppose that we have pairwise similarity judgments between all of the pages of a base document d_b, and all pages of a second document d_1. Suppose further that each page pair has a Boolean similarity measure so that one and only one of the elements in the set $\{a_i\}$ is 1 and the others are 0. How are we to compute the similarity of the base document to the second document? The initial answer is that we may average the page to document measure. This average is given by

$$m(d_1 ,d_2) = (1/(st)) \Sigma_i \ \Sigma_j \ m(p_i ,p_j),$$

where $m(p_i ,p_j)$ is the Boolean 4-tupil described above, and s and t are the number of pages in the two documents.

However, we must modify this average to account for an expectation that any one page will not match a random page in the second document. A partial average is taken by looking for "first" instances of match, near duplicate, or similarity. Let's consider an example.

> <u>Example:</u> Suppose that d_1 has four pages $\{p_{11}\ p_{12}\ p_{13}\ p_{14}\ p_{15}\}$ and d_1 has six pages $\{p_{21}\ p_{22}\ p_{23}\ p_{24}\ p_{25}\ p_{26}\}$ and then suppose that
>
> $m(p_{11}\ p_{21}) = (0\ 1\ 0\ 0)$
>
> $m(p_{12}\ p_{23}) = (1\ 0\ 0\ 0)$
>
> $m(p_{13}\ p_{24}) = (1\ 0\ 0\ 0)$
>
> $m(p_{14}\ p_{25}) = (1\ 0\ 0\ 0)$
>
> $m(p_{15}\ p_{26}) = (0\ 0\ 0\ 1)$
>
> $m(p_{ij}\ p_{kl}) = (0\ 0\ 0\ 1)$ for all others
>
> in this case the full average is: (3/30, 1/30, 0, 26/30).

A more realistic measure would be to take each page in the base document and check to see if $a_k = 1$ for k = 1, 2 or 3 of any page of the second document. In this case the partial average is (3/5, 1/5, 0, 1/5).

9.0: Pooling Techniques, an extended look

Polling of potential duplicate documents is the primary retrieval operation for duplication document detection. The pool is dependent on one document, called the base document, which is a document that is being examined for duplicates.

> *Prior judgments by humans can aid similarity based pooling of potential duplicates.*

Thus it is natural to filter or somehow sort all documents into sub collections. The elements of these collections can be ranked and queued in efficient workflow for human judgment (and efficient storage of judgments) regarding duplicate, near duplicate and similar judgments.

9.1: The pooling algorithms

A number of scoring algorithms are available, each designed to give an answer to a membership question having the following general form:

> *Given base document d_b find all documents that have a reasonable potential for being a near duplicate, or duplicate document of the document d_b.*

Since the analysis may be made at the page level, a membership question might also be:

> *Given a page p_b find all pages that have a reasonable potential for being a near duplicate, or duplicate page of the page p_b.*

Whether or not the membership judgment is made by an algorithm or by a human is important but may on occasion be ignored. This will allow the two types of "judgments" to be fused.

9.2: The similarity measures may have several different forms. The form can be Boolean. The form can be a value in an number interval such as [0,1]. This form of answer is most generally called "interval logic" with possibility membership, belief functions, fuzzy logic and probabilistic logic all being of this type.

A rough set that describes a stratified and overlapping membership or perhaps a knowledge structure that has been composed to layout the nature of the situation.

The most pragmatic next step, in software development, is to establish the ***minimal complexity*** regarded to treat the issue of observational scale ***with both Boolean and interval logic.***

10.0: Representational Algorithms

Initially I considered six types of representational algorithms. These use:

- Keywords and Manual or Pre-Existing Metadata,
- Layout and Shape Code,
- N-Gram,
- Wavelet,
- Oracle Full Text, and
- Oracle ConText Option.

for representations of a page/image unit.

By standardizing the metadata, we may treat each of the representational algorithms as a black box that is sent an image, text unit or image/page pair as input. In each case, a single representational set is generated as output.

This representational set will have a single number that measures the likelihood that the representational set was optimally generated if compared to a specific template for judging ground truth.

The definition of ground truth templates is relative to each of the representational algorithms.

Each representational algorithm has

- a standard input (generally an image or a text unit, or both)
- a standard output (generally a set of representational elements)
- a set of internally adjustable parameters
- a set of automatically generated "reliability" measures

Consistent with the D3 formalism, representation can be made at any of four levels. However the representation at a particular level might have to be derived from a number of steps.

The composition of these steps will produce a representational set of a certain type and a single value for the reliability measure of the completed composition.

References:

[1] Prueitt, P. (1997). Grounding Applied Semiotics in Neuropsychology and Open Logic, in IEEE Systems Man and Cybernetics Oct. 1997.

[2] Prueitt, P. (1998). An Interpretation of the Logic of J. S. Mill, in IEEE Joint Conference on the Science and Technology of Intelligent Systems, Sept. 1998.

[3] Tonfoni, G. (1999). On Augmenting Documentation Reliability through Communicative Context Transport. In SDIUT 99 Proceedings, this volume.

[4] Prueitt, P. (1996). Semiotic Design for Document Understanding, in Proceedings of the Workshop on Control Mechanisms for Complex Systems: Issues of Measurement and Semiotic Analysis. New Mexico State University and Army Research Office.

[5] Prueitt, P. (unpublished). Measurement, Categorization and Bi-level Computational Memory. http://www.bcngroup.org/area3/pprueitt/prueitt.htm

Duplicate Document Detection in DocBrowse

Richard Rogers, Vikram Chalana, Giovanni Marchisio, Thien Nguyen, Andrew Bruce
Mathsoft Data Analysis Products Division
1700 Westlake Ave. N, Suite 500, Seattle, WA

Abstract

Duplicate documents are frequently found in large databases of digital documents, such as those found in digital libraries or in the government declassification effort. Efficient duplicate document detection is important both for querying for similar documents, but also to filter out redundant information in large document databases. We have designed an algorithm to identify duplicate documents based on features extracted from the documents' digital images. We have also developed a cluster-based indexing scheme to allow efficient duplicate detection in large document databases. We present the results of evaluating our algorithms on several real and synthetic document image databases.

These algorithms are integrated with MathSoft's DocBrowse system for information retrieval from document image. DocBrowse supports duplicate document detection by allowing (1) automatic filtering to hide duplicate documents, and (2) ad hoc querying for similar or duplicate documents.

1 Introduction

Efficient detection of duplicate documents is an important problem in the context of large document image collections and has many different applications in digital libraries or the government declassification effort. There are many definitions of a duplicate document, and the appropriate definition depends on the context. We define three classes of duplicate documents:

Exact duplicates are defined as documents that have precisely the same information content and geometric layout [1] structure. The document image data, however, may differ due to degradations introduced by photocopiers, fax machines, or scanners. By identifying and removing exact duplicates, it is possible to filter out redundant information in databases. Duplicate documents may occur quite frequently in document databases, arising either from different sources, or being erroneously inserted into the database more than once.

Near duplicates are defined as documents which have almost the same information content and layout structure, but differ in isolated portions of the document. Near duplicates include, for example, the same document with and without margin notes or the same letter addressed to two different recipients. By identifying near-duplicate documents, the user may be able to quickly navigate through the database and useful additional information may be gathered about a given document. For example, if documents are found in the database which are exactly like the given document, except that they have additional annotations or margin notes, such as "top-secret", the same annotations may be applicable to the given document as well.

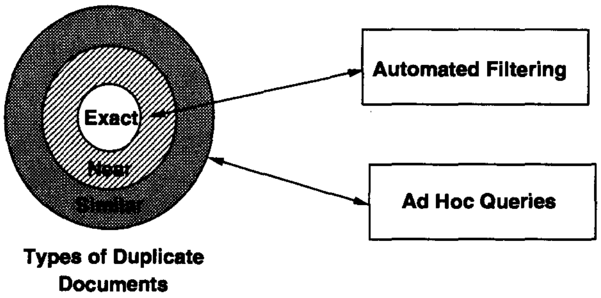
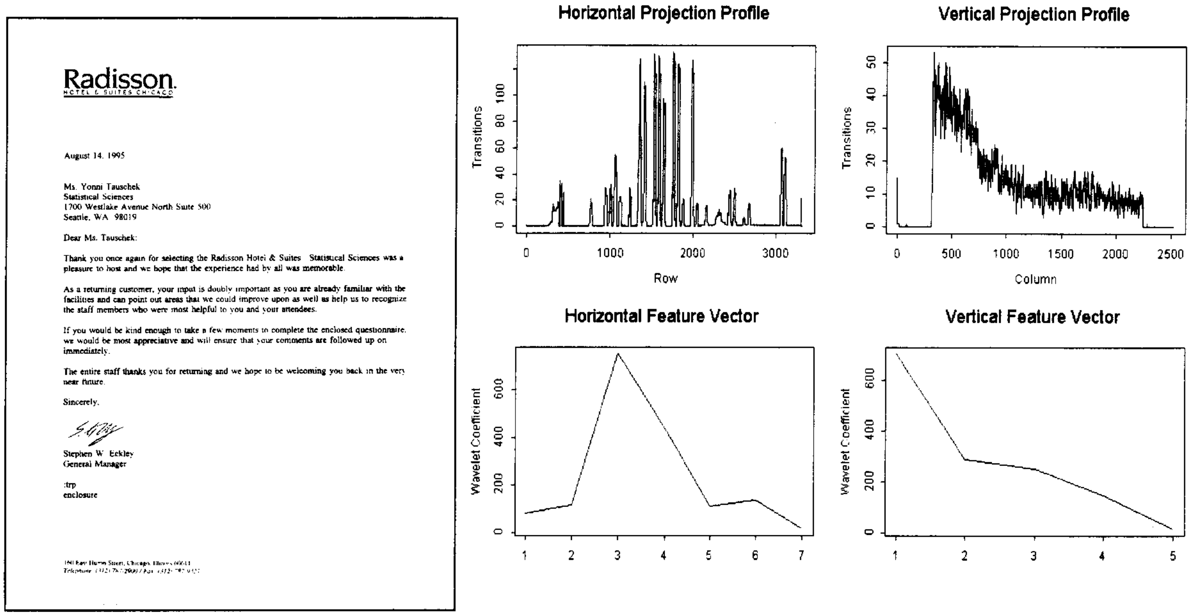


Figure 1: The three types of duplicate documents

Similar documents are loosely defined as all those documents that have similar linguistic content or just posses a similar geometric structure. By identifying documents similar to a given document, users can efficiently browse and navigate through the large document collections. Using a similar document detection method, all documents in the database can be clustered, where similar documents fall into one cluster; therefore, users of the database can look for related information within the given cluster.

Figure 1 illustrates the three types of duplicate documents. The three types define subsets of increasing size. Even though the three types of duplicate documents above can be identified using similar algorithms, they differ in the magnitudes of the allowable errors. Given a target (query) document x_0 and a document x, two types of errors may be defined:

$$\text{Type I Error: } p(x \;!\!= x_0 \mid x = x_0)$$
$$\text{Type II Error: } p(x = x_0 \mid x \;!\!= x_0). \qquad (1)$$



Radisson.
HOTEL & SUITES CHICAGO

August 14, 1995

Ms. Yonni Tauschek
Statistical Sciences
1700 Westlake Avenue North Suite 500
Seattle, WA 98019

Dear Ms. Tauschek:

Thank you once again for selecting the Radisson Hotei & Suites Statistical Sciences was a pleasure to host and we hope that the experience had by all was memorable.

As a returning customer, your input is doubly important as you are already familiar with the facilities and can point out areas that we could improve upon as well as help us to recognize the staff members who were most helpful to you and your attendees.

If you would be kind enough to take a few moments to complete the enclosed questionnaire, we would be most appreciative and will ensure that your comments are followed up on immediately.

The entire staff thanks you for returning and we hope to be welcoming you back in the very near future.

Sincerely,

Stephen W. Eckley
General Manager

:trp
enclosure

160 East Huron Street, Chicago, Illinois 60611
Telephone (312) 787-2900 / Fax (312) 787-[illegible]

Figure 2: Example document image, its horizontal and vertical projection profiles, and the resulting wavelet transform coefficient feature vector

The Type I error (false reject) is the probability of classifying the document x as different from x_0, when in reality the two documents are duplicates. The Type II error (false accept) is the probability of classifying the documents as duplicates, when in reality they are different.

Type I and Type II errors are related to the concepts of precision and recall, which are used in the context of information retrieval. Low Type I error corresponds to high recall and low Type II error corresponds to high precision. For the automatic identification of exact duplicate documents, the Type II error rate needs to very low (high precision) because documents should not be erroneously marked as exact duplicates and eventually removed from the database. For similar document querying, the Type I error needs to be very low (high recall) because the user query should not miss documents which are really near duplicates.

The remainder of our paper is organized as follows: Section 2 describes the algorithms that we have developed for duplicate document identification. Section 3 describes the precision and recall results of our algorithms. Finally, we will briefly describe the DocBrowse system and how it supports duplicate document detection.

2 Duplicate Detection Algorithms

The key steps in any algorithm for duplicate document detection are: feature extraction and feature matching. The features extracted from a given document should provide a global representation of the document and should be relatively invariant to distortions that may arise in the documents. The features should also provide a good compromise between computational complexity and discriminatory power. For example, a feature such as the "size of the document" provides a global representation of the document, is relatively invariant to distortions, and is quick and easy to compute but it may not be able to discriminate between two non-similar documents. One the other hand, if the entire document bitmap is used as a feature vector, it has good discriminatory power, but the matching may not be computationally efficient. For databases containing large numbers of document images, feature vector size is also a consideration.

This paper describes several improvements to the image-based duplicate detection algorithm described in [2]. The improvements include improved exact duplicate detection accuracy, support for near duplicate detection (e.g., redacted or annotated versions of a document are detected as duplicates), and cluster-based indexing to support efficient duplicate detection for very large document image databases.

2.1 Exact Duplicate Detection

For exact duplicate detection, our feature vector consists of the coarse scale coefficients of the discrete wavelet transform (DWT) of the image's horizontal and vertical pixel projection profiles. The two sets of DWT coëfficients are concatenated to form a single feature vector. The projection profiles use the raster pixel count divided by the image's total pixel count in order to compensate for the thickening and thinning caused by photocopying, faxing, etc. Figure 2 shows an example page image, its projection profiles, and the resulting feature vector. The DWT is computed to six levels using the S8 wavelet and periodic boundary condition. The projection profiles can be computed quickly and provide discriminatory power by summarizing the

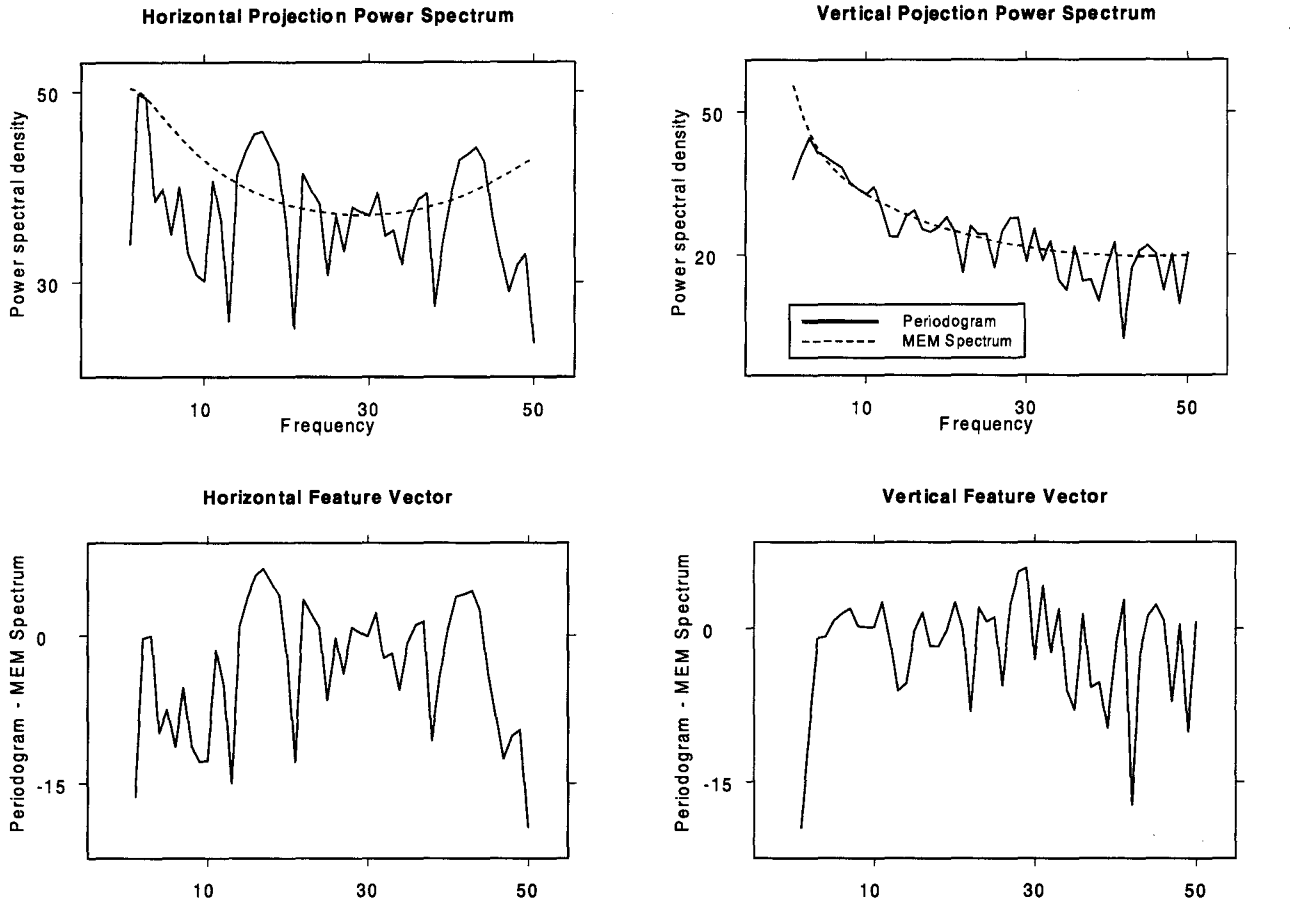


Figure 3: Low frequency portions of the periodogram and MEM power spectrum estimates for the horizontal and vertical white-to-black transition projection profiles (top row) for the example image in Figure 2. The difference between the two estimates (bottom row) is used as the clustering feature vector.

geometric layout of the document's contents. The wavelet transform provides a degree of robustness to the fine-scale variations that occur between duplicate document images. The wavelet transform also provides a substantial reduction in feature vector size. For a typical 300 dpi 8.5"x11" binary image (approximately one megabyte), our feature vector is about 760 bytes. As with many projection profile-based algorithms, our features are sensitive to page skew. We assume that document images have been deskewed prior to feature vector computation. In the DocBrowse system, deskewing is performed by the Optical Character Recognition (OCR) engine prior to duplicate detection. Additional preprocessing such as margin streak, punched hole, and speckle removal would be beneficial to exact duplicate detection accuracy. The near duplicate and clustering features described below are more tolerant of such degradations.

Feature matching is accomplished by summing the mean absolute differences (MAD) of the horizontal and vertical DWT coarse scale coefficients. Smaller sums indicate closer matches, zero indicating the images being matched are identical. The sum may be used to rank documents in similarity order, or it may be thresholded to classify document pairs as exact duplicates. Duplicate images may be subject to cropping and translational variations. To compensate for the latter, the MAD is computed for several integral shifts of the feature vectors being matched and the minimum value is used. Due to the shifting and possible cropping differences, only the overlapping portions of the feature vectors are used to compute the MAD.

2.2 Near Duplicate Detection

We could potentially use the exact duplicate detection algorithm described above to detect near duplicates as well by simply increasing the similarity threshold at which images are considered to match. However, the addition of margin notes or other localized content to a near duplicate image distorts the entire projection profile since it is divided by the image's total pixel count. This results in poor accuracy for near duplicate detection.

To overcome this problem, the near duplicate feature vectors are computed from white-to-black transition projection profiles rather than pixel count projection profiles. The transition counts are robust to thickening and thinning, and also allow the effects of localized content additions to remain localized in the projection profile since no global division takes place. The transition projections also provide the benefit of allowing resolution independence. We perform linear spline interpolation to 300 dpi on the transition profiles so that images scanned at different resolutions may be compared. The DWT and matching are performed as in the exact duplicate detection algorithm.

While the transition profiles are tolerant of localized content additions and subtractions, they lack the discriminatory power of the pixel count projections for exact duplicate detection. This results in the unfortunate situation of requiring two feature vectors to support accurate near and exact duplicate detection.

2.3 Similar Document Clustering

In order to find all of the duplicates in a document database (e.g. to remove the redundant copies), each document in turn must be used as the target of a duplicate query. If each query must perform a match against every document in the database, the time required to find the duplicates grows quadratically with the size of the database. For large databases, this time may be unacceptably large.

To solve this problem, we have developed a cluster-based indexing algorithm. By finding groups of similar documents within the database efficiently, duplicate searches may be restricted to the group of which the query document is a member since duplicates are expected to fall into the same group. The group size is much smaller than the full database, so duplicate searches are fast. The approach we take to forming groups of similar documents is to apply a statistical clustering algorithm to the set of feature vectors of the images in the database. The clustering algorithm is described in section 2.3.2. Unfortunately, the need for registration and different feature vector lengths for the near and exact duplicate feature vectors makes them unsuitable for use with statistical clustering algorithms. This led to the development of yet another feature vector, which is described in section 2.3.1. This feature vector seems to have better accuracy for exact duplicate detection than our exact duplicate features, and accuracy comparable to our near duplicate features for near duplicate detection. Thus we are hopeful that in the future, this feature vector will prove suitable for all three purposes.

2.3.1 Power Spectrum Feature Vector

The feature vector used for clustering similar documents is also based on the horizontal and vertical transition projection profiles of the document image. Rather than performing a wavelet transform, we compute periodogram power spectrum estimates of the profiles. We concatenate the 50 lowest frequency coefficients of the horizontal and vertical periodograms to form a 100 element vector. The spectra tend to have large peaks corresponding to the character and line pitch and their harmonics, so that the vectors for dissimilar documents often appear similar. To remedy this, we subtract a maximum entropy method (MEM) estimation of the projection profiles' power spectra and use the result as the feature vector for clustering similar documents. The MEM power spectrum is computed with 50 poles using Burg's method [3]. Figure 3 illustrates the feature vector computation. Since frequency spectra are linear shift invariant and all vectors consist of 100 elements, no registration is required in the matching step. We simply use the MAD of the feature vectors to measure the similarity of a pair of documents.

2.3.2 Clustering Algorithm

We group the power spectrum feature vectors of the document database into clusters using the Generalized Vector Quantization (GVQ) algorithm [4]. GVQ divides the set of vectors into a specified number of clusters. It performs a randomized search, attempting to minimize the sum of the distances of each member of the cluster to the cluster center. Unlike other randomized vector quantization algorithms [5][6][7], GVQ has been proven to converge to the globally optimal clustering. Although somewhat slower than some deterministic vector quantization algorithms, we found that GVQ produced significantly better clustering results for our task.

We use the number of documents in the database divided by 1000 as the number of clusters, resulting in an average of 1000 documents per cluster. This number was chosen based on the speed of our feature matching function, so that duplicate searches restricted to documents within the same cluster would be reasonably fast. Such restricted searches are implemented by assigning a unique number to each cluster. For each document in the database, the number of the cluster of which it is a member is stored. An inverted cluster number index allows efficient retrieval of the documents in the same cluster as a duplicate query target. Feature vector matching is then performed on this subset of the database. We expect our current implementation of this method to scale to databases on the order of 10^5 documents.

3 Duplicate Detection Accuracy

We evaluated the effectiveness of the duplicate detection algorithms on several test databases. The MathSoft database was constructed from 55 unique documents. The database contains two exact duplicate pages for a total of 57 pages. The database mostly consisted of business documents (letters, travel itineraries, brochures, etc.) in English. Each page underwent eight degradations to populate the database with four sets of exact duplicates and four sets of near duplicates, for a total of 456 pages. The exact duplicates were generated by photocopying and faxing, photocopying twice and faxing, and photocopying and faxing twice. The photocopied and faxed documents padded by 0.5 inches on each side and translated 0.25 inches down and right comprise the final set of exact duplicates. In order to evaluate the algorithms' ability to handle redactions, we created a set of near duplicates by redacting about 4 lines from each page. To test the algorithms' ability to handle margin notes, we created three sets of near duplicates where each page was stamped over an area approximately 2" x 3" with the words "MathSoft Confidential," and each page was stamped once with "DRAFT" and "MathSoft Confidential."

We also evaluated the algorithms on a database consisting of two duplicates of 311 pages of English language technical journal articles from the University of Washington Intelligent Systems Lab's UW-II English/Japanese Document Image Database (UWEJDID) [8]. This database contains one first generation scan and one scanned photocopy of each page. These pages are challenging for duplicate detection because the two instances of a page may have different amounts of material from the facing page, or one instance might contain severe margin streaks.

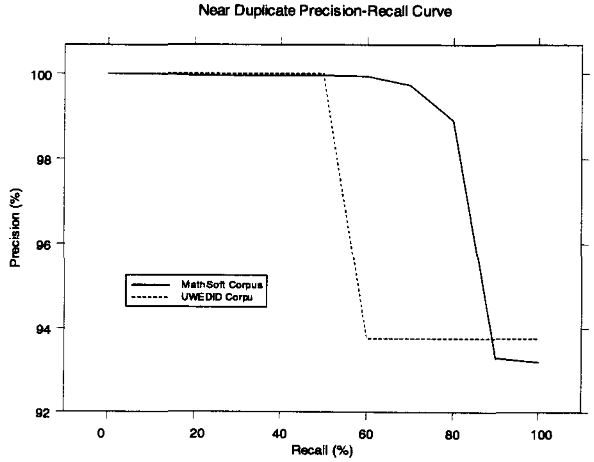
We evaluated the duplicate detection algorithms by using each document, in turn, as the target of a duplicate query. We computed the distances against all other documents in the database. The results are presented as precision-recall curves. The precision, P_n, of an information retrieval system for some cutoff-point n is defined as the fraction of the top n documents that are relevant to the query:

$$P_n = n_r / n \qquad (2)$$

where n_r is the number of retrieved documents that are relevant and n is the total number of retrieved documents. In contrast, the recall, R_n, of a system is defined as the proportion of the total number of relevant documents that were retrieved in the top n documents:

$$R_n = n_r / N_r \qquad (3)$$

where N_r is the total number of relevant documents in the database.



In a precision recall curve, the average precision values are displayed for different values of recall. The ideal precision-recall curve is a horizontal straight line where the precision is 100% for all values of recall, i.e., all the retrieved documents are always relevant. For the case of near duplicate document detection, relevant documents are the ones that are near duplicates of the query target. For exact duplicate detection, relevant documents are the ones that are exact duplicates. Figure 4 gives the precision-recall curve for the near duplicate detection algorithm. Figure 5 gives the precision-recall curve for the exact duplicate detection algorithm. The poor accuracy of the exact duplicate detection algorithm for the UWEJDID database is expected because of the large differences between many of its duplicate pages.

We also summarize the precision-recall curve by a measure known as the 11-point average precision in

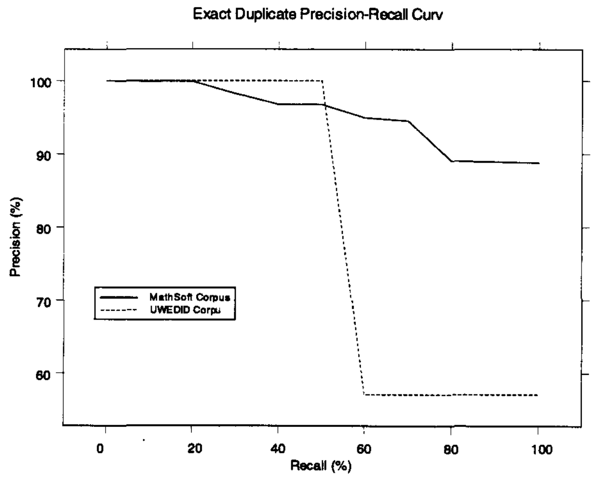


Figure 5: Precision-recall curve for the exact duplicate detection algorithm.

Table 1. The 11-point precision is defined as the average precision at 11 different values of recall, i.e., 0%, 10%, 20%, 30%, ..., 100%.

Table 1: Near and exact duplicate detection 11-point average precision

Database	Near	Exact
MathSoft	98.6%	95.3%
UWEJDID	97.2%	80.5%

Table 2: Power Spectrum feature vector near and exact duplicate detection 11-point average precision

Database	Near	Exact
MathSoft	98.5%	96.7%
UWEJDID	97.0%	97.0%

We also evaluated the power spectrum feature vector's accuracy on the two databases. Figures 6 and 7 give the power spectrum feature vector's precision-recall curves for near and exact duplicate detection, respectively. Table 2 gives the 11-point average precisions for the clustering feature vector.

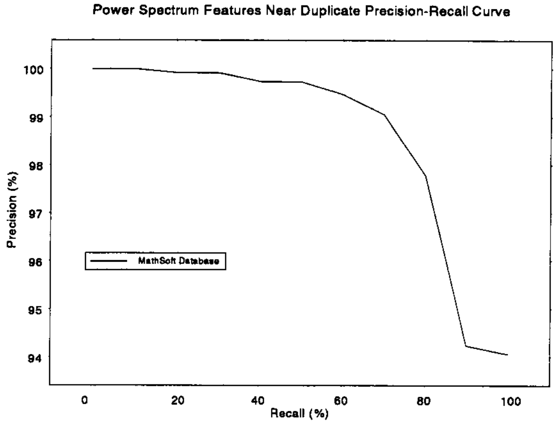


Figure 6: Power spectrum feature vector's precision-recall curve for near duplicate detection.

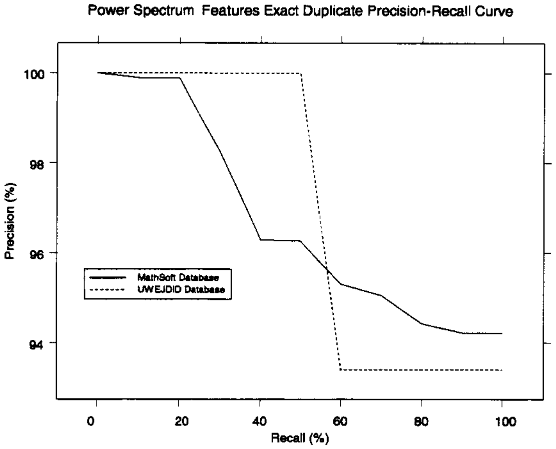


Figure 7: Power spectrum feature vector's precision-recall curve for exact duplicate detection

In order to evaluate the cluster-based indexing method, we created a database containing 12 duplicates of each of 782 unique pages for a total of 9384 pages. The pages consisted of the 55 documents from the MathSoft database; 311 English technical article pages, 214 Japanese technical article pages, and 61 English memo pages from UWEJDID; and 141 pages of declassified U.S. Department of Energy documents. The pages were synthetically thickened, thinned, margin streaked, and annotated with margin notes to create six near duplicates and six exact duplicates of each page. Figure 8 gives the precision-recall curves for the near duplicate detection algorithm and the power spectrum feature vector for near duplicate detection on this database. Since the power spectrum feature vector achieves higher precision than the near duplicate detection features on this database (94.9% vs 93.4% 11-point average precision), we are hopeful that the power spectrum features will prove suitable for all three duplicate detection tasks, thus eliminating the need to maintain three separate algorithms.

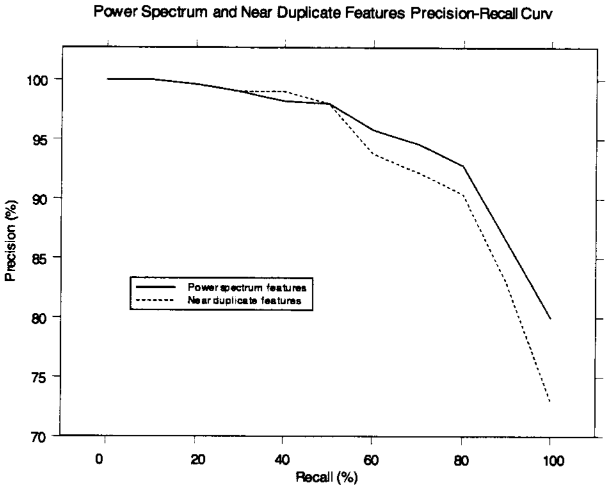


Figure 8: Near duplicate detection precision-recall curves for the power spectrum features and near duplicate detection features on the 9384 page database.

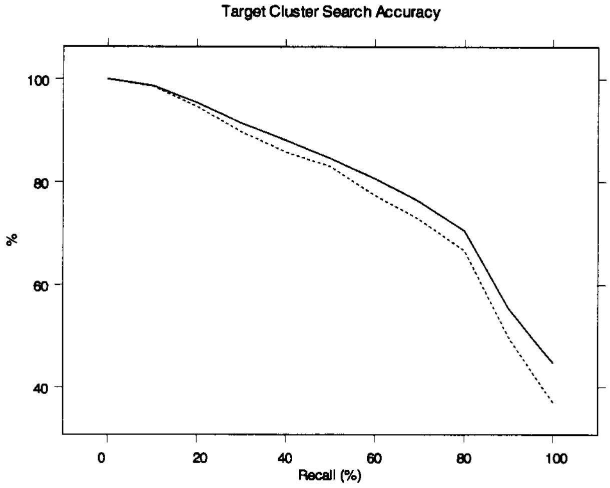


Figure 9: The solid line is the precision-recall curve for near duplicate searches restricted to the target document's cluster. The dashed line indicates the fraction of queries for which each level of recall was achieved.

To evaluate the effectiveness of our cluster-based indexing method, we used GVQ to divide the database into nine clusters. We then use each document in the database, in turn, as the target of a near duplicate query, restricting the search to documents in the same cluster as the query target. Because of the statistical nature of clustering, it is possible that duplicate instances of a document are assigned to different clusters. When this happens, it is impossible to achieve 100% recall if the search does not cover all of the clusters. To alleviate this problem, we also present results for searches of the target document's cluster and the next closest cluster as measured by the distance between the target document's feature vector and the cluster centers. Figure 9 gives the precision-recall curve for the power spectrum feature vector for searches restricted to the target document's cluster. Figure 10 gives the precision-recall curve for searches of the two closest clusters. A precision of zero is used for levels of recall that could not be reached for a query. Figures 9 and 10 also plot the percentage of queries for which each level of recall was attained. Table 3 presents the 11-point average precisions and average recalls for searches of the full database and searches restricted to one and two clusters.

Table 3: Cluster-based indexing precision and recall

Search Domain	11-point average precision	Average recall
Full database	94.9%	96.8%
Query target's cluster	77.7%	76.5%
Target and closest cluster	89.0%	88.2%

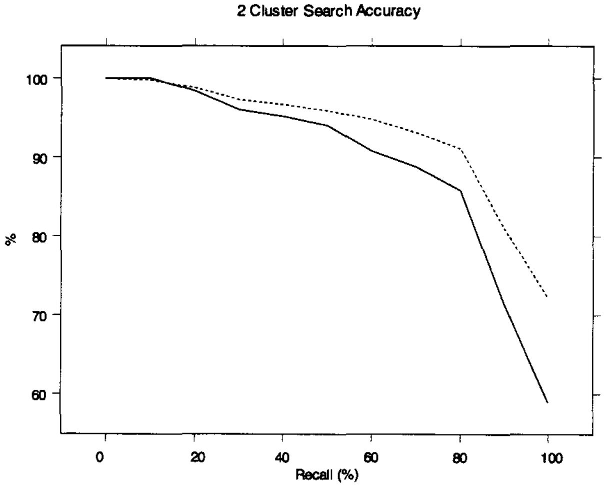


Figure 10: The solid line is the precision-recall curve for near duplicate searches restricted to the target document's two closest clusters. The dashed line indicates the fraction of queries for which each level of recall was achieved.

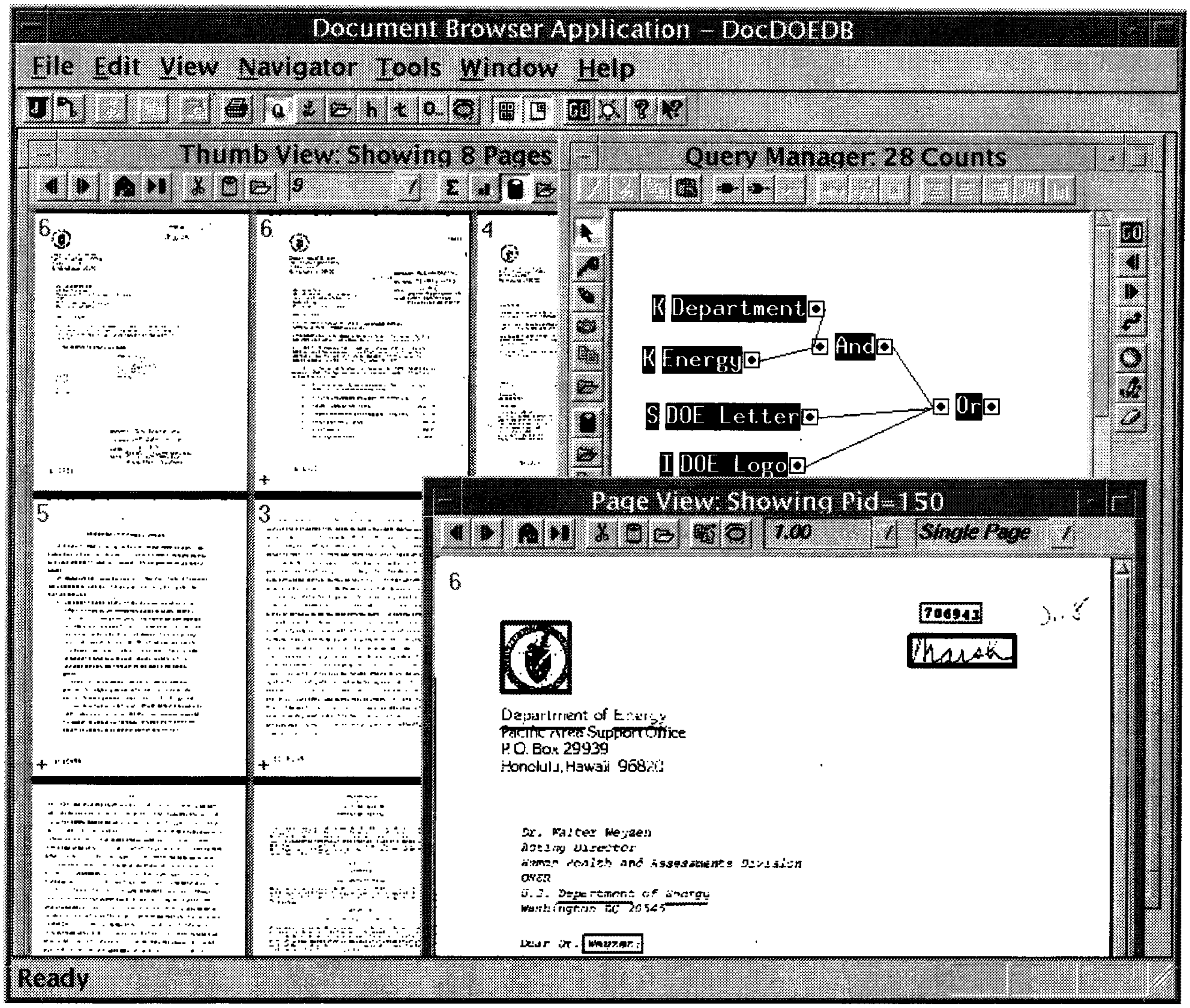
Including the additional cluster in the search markedly improves the precision and recall. This indicates that the clustering algorithm is frequently splitting duplicate documents into different clusters. Since the two cluster search performs poorly compared to the full database search, precision and recall could be improved by including more clusters in the search at the cost of longer search times. We are investigating other methods [9][10] to efficiently handle large databases without sacrificing the accuracy potential of our feature vectors.

4 Duplicate Detection in DocBrowse

We have implemented the duplicate detection and cluster-based indexing algorithms both as a platform independent statically linked library and as a Win32 Dynamically Linked Library (DLL). The algorithms are accessed through a C interface, which provides functions for computing the various feature vectors from TIFF images, feature vector matching, feature vector clustering, and cluster-based indexing of the feature vectors. The library is used in MathSoft's DocBrowse system as well as in stand-alone duplicate detection applications.

The DocBrowse system for information retrieval from document images is currently under development at MathSoft. The DocBrowse system is designed to archive, retrieve, and browse large collections of digitized paper documents. This system has many potential applications including digital library initiatives and government declassification efforts. While DocBrowse can be used purely as a high-end text-based information retrieval engine, the most unique feature of DocBrowse is its support for image content-based queries and mixed-mode queries integrating text and image content. The kinds of image-content queries supported by DocBrowse are queries on logos or other graphical zones contained on the document, queries on hand-written signatures, queries on images of words (word-image spotting), and queries on images of the entire document (similar or duplicate document detection). The motivation behind such mixed-mode queries is that: (1) optical character recognition (OCR) on digital documents usually perform poorly on highly degraded documents, such as legacy documents, and (2) documents may contain many non-text zones, such as logos and hand-written text, which the users may wish to search upon.

DocBrowse consists of three main components: (1) A browser and graphical user interface (GUI) for visual querying and sifting through a large digital document image database; users submit queries from the GUI without having to directly manipulate SQL code. Tools are provided for visually browsing the results of a query. The GUI also provides support for iterative query refinement and expansion. (2) database management system (DBMS) for storing, accessing, and processing the data, and (3) " DocLoad," an application which processes the raw document images through specialized document analysis software (OCR, page segmentation, and information retrieval), inserts this information into the database, and creates the database indices to permit efficient searching. More information about DocBrowse, including technical


Document Browser Application - DocDOEDB
File Edit View Navigator Tools Window Help
Thumb View: Showing 8 Pages
9
Query Manager: 28 Counts
K Department
K Energy
And
S DOE Letter
I DOE Logo
Or
6
6
4
5
3
Page View: Showing Pid=150
1.00
Single Page
Ready

6

706943

Department of Energy
~~Pacific Area Support Office~~
P.O. Box 29939
Honolulu, Hawaii 96820

Dr. Walter Weyzen
Acting Director
Human Health and Assessments Division
OHER
U.S. Department of Energy
Washington DC 20545

Dear Dr. Weyzen:

Figure 11: The DocBrowse system in action showing the Query Manager window, the Thumbnail Vie window and the Page View window. The query consists of two keyword terms, a logo image term ("DOE Logo"), and a similar document term ("DOE Letter") combined using Boolean operators. Duplicate document filtering is turned on; thus, the Thumbnail View and the Page View windows show the top document in a stack of documents hidden below the displayed document. The number on the top-left of each document shows the number of duplicates for each document.

papers, is available at the following web site: **www.statsci.com/docbrowse**

DocBrowse supports the query and retrieval of duplicate documents by providing two unique user interface features: (1) automatic filtering to hide exact or near duplicates and, (2) support for ad hoc queries for finding duplicates or similar documents for a selected document. The user has control over specifying a similarity value threshold to be used as the cut-off. In both these features, the user has the option to use a text-based method or the image-based matching methods described in this paper. Techniques for combining the text- and image-based duplicate detection methods are an area of ongoing research within MathSoft.

With automatic filtering, duplicate or near duplicate documents are retrieved and placed in a single stack, hidden behind one original document. The user can then choose to browse the whole stack, or just view the top document from the stack. The automatic filtering is optional and users can turn it off, if they so desire.

For ad hoc similar document queries, users can submit a document as an example query and ask the system to find documents similar to the example. This similar document query can be integrated with all the other queries supported by the DocBrowse system. Thus, a single query may consist of a similar document term, a logo search term, a keyword term, and a word image spotting term.

Figure 11 shows a screen shot of the DocBrowse system where both the duplicate document detection features are illustrated. The query manager window in the figure shows a query containing a similar document term (the "DOE Letter" term) combined with other query terms. The thumbnail view and page view windows show the filtering of duplicate documents where duplicate documents are stacked behind the original document and a number on the top-left of the document indicates the number of duplicates in the stack.

5. Conclusion

In conclusion, we have found that it is possible to efficiently and accurately identify exact duplicate documents, as well as near duplicate and similar documents, from a large database of document images. We have integrated the duplicate document detection algorithms into a complete document imaging system (DocBrowse) and found the two modes of operation, implicit filtering and stacking of duplicates and support for similarity document query, to be very useful and intuitive. Cluster-based indexing is used to support efficient duplicate detection for large databases.

References

[1] R.M. Haralick, Document image understanding: Geometric and logical layout, *IEEE Conf. on Computer Vision and Pattern Recognition* , Seattle, WA, June 1994, 385-390.

[2] V. Chalana, A. Bruce and T. Nguyen, Duplicate document detection in DocBrowse, in *Proc. SPIE Conf. on Document Recognition*, San Jose, CA, Jan 1998.

[3] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes in C, 2nd ed. (Cambridge University Press, New York, 1992).

[4] U. Moller, M. Galicki, E. Baresova and H. Witte, An efficient vector quantizer providing globally optimal solutions, *IEEE Trans. Signal Processing*, **46** (1998) 1515-1529.

[5] K. Zeger, J. Vaisey and A. Gersho, Globally optimal vector quantizer design by stochastic relaxation, *IEEE Trans. Signal Processing*, **40** (1992) 310-322.

[6] A.R. Haig, E. Gordon, G. Rogers and J. Anderson, Classification of single-trial ERP sub-types: Application of globally optimal vector quantization using simulated annealing, *Electroencephalogr. Clinical Neurophysiol.*, **94** (1995) 288-297.

[7] M. Galicki, U. Moller and H. Witte, Nueral clustering networks based on global optimization of prototypes in metric spaces, *Neural Computing and Applications*, **5** (1997) 1-13.

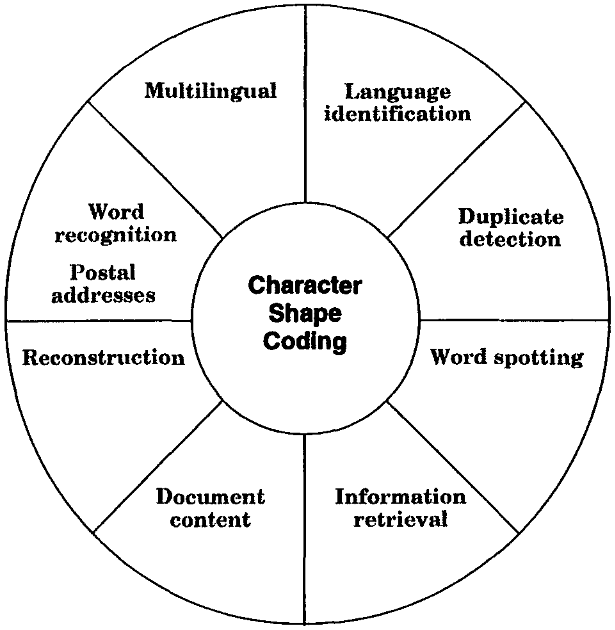
[8] I.T. Phillips, S. Chen, J. Ha and R.M. Haralick, Reference Manual for the UW English/Japanese Document Image Database II, University of Washington Intelligent Systems Lab, 1995.

[9] A. Broder, On the resemblance and containment of documents, in *Compression and Complexity of Sequences*, Positano, Itally, June 1998, 21-29.

[10] A.P. Berman and L.G. Shapiro, Triangle-inequality-based pruning algorithms with triangle tries, in *Storage and Retrieval for Image and Video VII*, B. Yeung, C. Yeo and A. Bouman, eds. (San Jose, CA, 1999) 356-365.

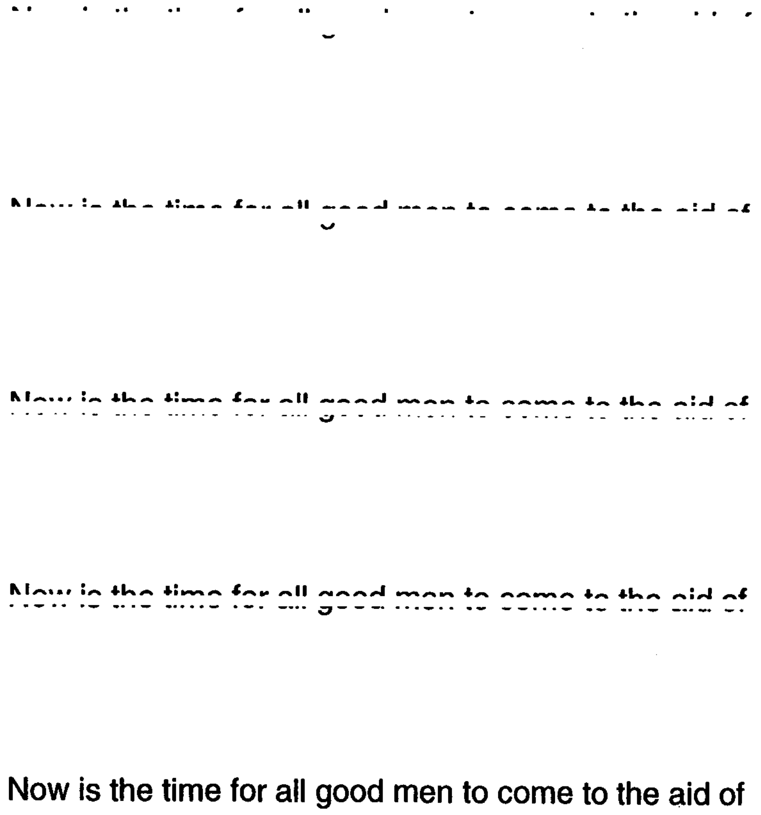
Applications of Character Shape Coding

Larry Spitz
Document Recognition Technologies, Inc.
spitz@docrec.com

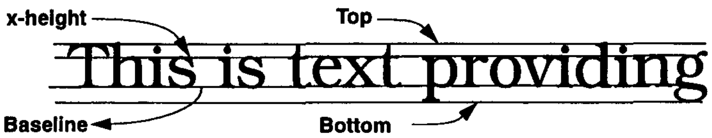


What is character shape coding?

A computationally inexpensive, robust, information-preserving core technology for the representation of text images



Character shape coding



V_0 = Ascenders, descenders, connected components
V_1 = Eastward concavity
V_2 = Southward concavity
V_3 = Vertical stems
V_4 = V_0 + Eastward concavity + crossbar
V_5 = V_3 + Eastward concavity + crossbar

Characters	V_0	V_1	V_2	V_3	V_4	V_5
amorsuvxwz	x	x	x	x	x	x
n			n			
c		e			c	
e					e	
ACGIOQSTUVWXYZflt	A			A	A	A
HMN				N		N
bhkL				b		b
BDEKR				E		E
PF				P		P
dJ				d		d
j	j					
i	i					
gpqy	g					

From image to CSC

Character codes

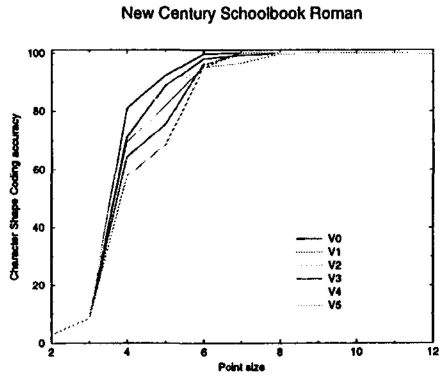
```
Confidence in the international
monetary system was shaky enough be-
fore last week's action.
```

Character shape codes

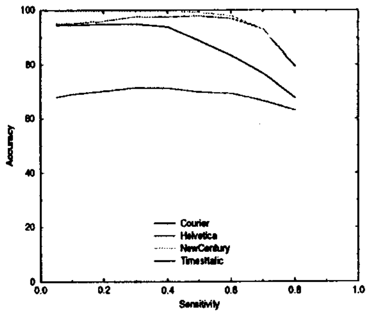
```
AxxAAxxxx ix AAx ixAxxxxAixxxA
xxxxAxxg xgxAxx xxx xAxAg xxxxgA Ax-
Axxx AxxA xxxA'x xxAixx.

AxxAAexee ix AAe ixAexxxAixxxA
xxxeAxxg xgxAex xxx xAxAg exxxgA Ae-
Axxe AxxA xeeA'x xeAixx.

AxnAAenee in AAe inAexnxAixnxA
xxneAxxg xgxAex xxx xAxAg enxxgA Ae-
Axxe AxxA xeeA'x xeAixn.

AxxAdxxxx ix Abx ixAxxxxAixxxA
xxxxAxxg xgxAxx xxx xbxbg xxxxgb bx-
Axxx AxxA xxxb'x xxAixx.

AxxAAexce ix AAe ixAexxxAixxxA
xxxeAxxg xgxAex xxx xAxAg exxxgA Ae-
Axxe AxxA xeeA'x xcAixx.

AxxAdexce ix Abe ixAexxxAixxxA
xxxeAxxg xgxAex xxx xbxbg exxxgb be-
Axxe AxxA xeeb'x xcAixx.
```

Why shape code?

- Character shape coding is very tolerant of low resolution and poor image quality

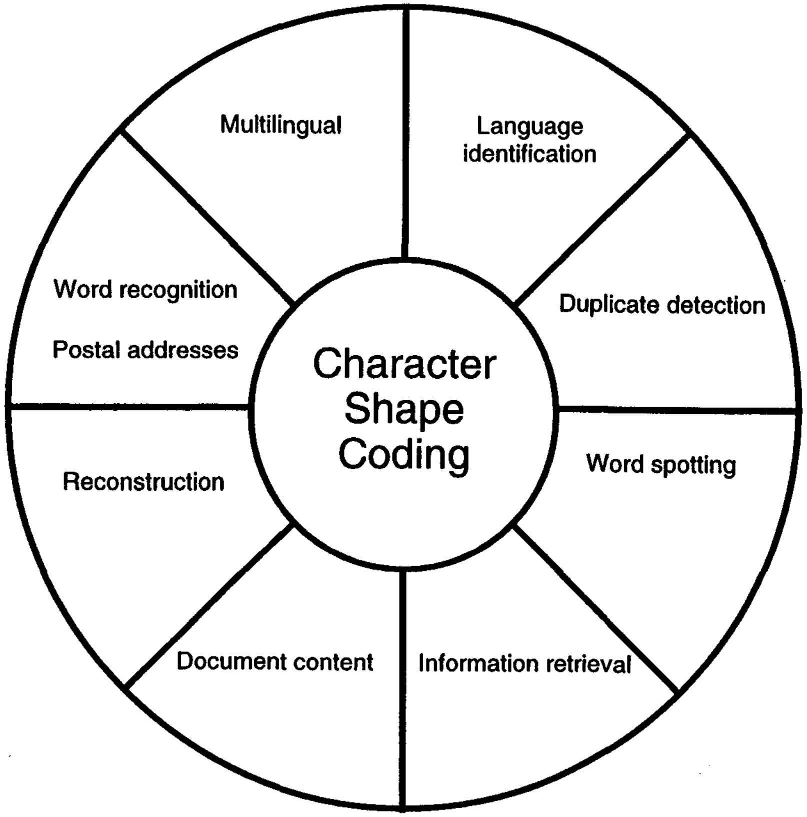


V_5



- The computational burden is low
- It provides a representation sufficient to support text-based applications

Applications



Language identification

- WST based

  Comparison of high frequency word shapes with training data

	English			French			German		
Token	Rank	Occ	Word(s)	Rank	Occ	Word(s)	Rank	Occ	Word(s)
AAx	1	8.1	the, The						
ix	2	4.1	is, in				4	3.0	im, in
Ax	3	3.8	to	1	14.4	la, le, du			
xA	4	3.5	of						
xxA	5	2.9	and				3	3.3	auf
Axx				2	7.7	les, des	1	8.6	der, das
xx				3	3.7	en			
Aix							2	5.3	die, Die

- n-gram based

  comparison of the distribution of bigrams and trigrams against distributions derived from training data

Increasing the number of languages

Language	Accuracy (%)
Afrikaans	97
Croatian	100
Czech/Slovak	44
Danish	96
Dutch	100
English	95
Finnish	75
French	92
Gaelic	86
German	97
Hungarian	94
Icelandic	96
Italian	95
Norwegian	95
Polish	100
Portuguese	100
Rumanian	93
Spanish	97
Swahili	97
Swedish	98
Turkish	93
Vietnamese	100
Welsh	97

Duplicate document content detection

- Tolerance for differences in font, layout

Four score and seven years ago, our fathers brought forth upon this continent a new nation: conceived in liberty, and dedicated to the proposition that all men are created equal.

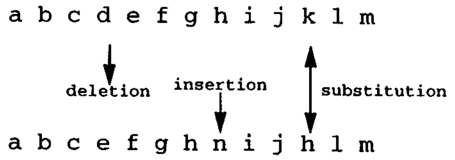
Four score and seven years ago, our fathers brought forth upon this continent a new nation: conceived in liberty, and dedicated to the proposition that all men are created equal.

- Comparison of document handles

Pxxx xcxxe xxd xexex gexxx xgx, xxx AxAbexx bxxxgbA AxxAb xgxx Abix cxxAixexA x xex xxAixx: cxxceixed ix AibexAg, xxd dedicxAed Ax Abe gxxgxxiAixx AbxA xAA xex xxe cxexAed egxxA.

Pxxx xcxxe xxd xexex gexxx xgx, xxx AxAbexx bxxxgbA AxxAb xgxx Abix cxxAixexA x xex xxAixx: cxxceixed ix AibexAg, xxd dedicxAed Ax Abe gxxgxxiAixx AbxA xAA xex xxe cxexAed egxxA.

Levenshtein Distance



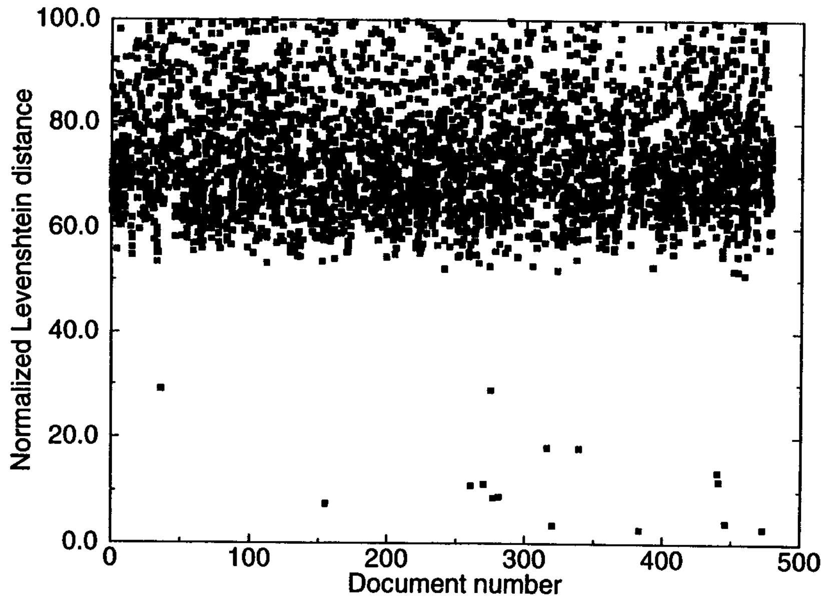
a b c d e f g h i j k l m

deletion insertion substitution

a b c e f g h n i j h l m

- $C_d = C_i = C_s = 1$
- $D = N_d C_d + N_i C_i + N_s C_s$

Normalized Levenshtein Distance



Word-spotting

- 39 index terms
- 4059 zones on 695 pages from University of Washington English Document Database I
- Recall (true positive rate) 72%, precision 82%

		Truth		Sum
		+	–	
Detection	+	908	197	1005
	–	350	25650	26000
Sum		1258	25847	27105

- 196 zones on 21 pages of *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*
- Recall 83%, precision 88%

		Truth		Sum
		+	–	
Detection	+	80	11	91
	–	16	712	728
Sum		96	723	819

Redaction/Highlighting

Four score and seven years ago, our fathers brought forth upon this continent a new nation: conceived in liberty, and [redacted] to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so [redacted], can long endure. We are met on a great battlefield of that war.

We have come to dedicate a portion of that field as a final resting place for those who here gave their lives that this nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we cannot dedicate, we cannot consecrate, we cannot hallow this ground. The brave men, living and dead, who struggled here have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here.

It is for us the living, rather, to be [redacted] here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here [redacted] to the great task remaining before us, that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion, that we here highly resolve that these dead shall not have died in vain, that this nation, under God, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish from this earth.

- In the worst case (V_0), would have also redacted {Activated, Estimated, heliostat, Indicated, Intimated, Medicated, Salivated}

Adding Lexical Information

Four score and seven years ago, our fathers brought forth upon this

Word	WST	DRE	Candidates
Four	Axxx	[Ff][or][uo][rm]	Four from
score	xexxe	score	
and	xnA	and	
seven	xexen	seven	
years	gexxx	years	
ago	xgx	ago	
our	xxx	[ow][ua]r	our war
fathers	AxAAexx	fathers	
brought	AxxxgAA	brought	
forth	AxxAA	forth	
upon	xgxn	upon	
this	AAix	this	

Comprehensive Lexicon

Word	WST	DRE	Candidates
Four	Axxx	[ACEGIJMNOPRSVWXYZbdfhlt][mruvaozxsw][murowaszvx][osmawxruz]	Ammo Arms Arum Aura Avow Cams Cars Caws Coax Coma Coos Cows Cram Craw Crow Crux Curs Cuss Czar Ears Emus Eras Errs Exam Gass Goos Gram Grow Gums Guru Isms Jams Jars Jaws Jazz Joss Mama Mars Mass Maws Moms Moor Moos Moss Mows Mums Muss Norm Nova Nows Oars Ours Ovum Para Pars Pass Paws Poor Pour Pows Pram Prom Pros Prow Puma Purr Puss Rams Raws Razz Roam Roar Room Rows Rums Sass Saws Soar Sour Sows Sums Swam Swum Vows Warm Wars Woos Worm Wows Xmas Yams Yaws Your Zoom Zoos baas bars bass boar boas boom boor boos boss bows boxs bras brow bums burr burs buss buzz dams doom door dorm doss dour dram draw drum duma duos farm foam form four from furs fuss fuzz hams harm haws hoar hoax homo hour hums lams lass lava laws loam loom loos loss lows luau tams taro tars taws tors toss tour tows tram trow tsar twos
score	xexxe	[rsvw][ec][uvaor][surv]e	reuse revue scare score serve verse verve weave
and	xnA	an[dt]	and ant
seven	xexen	se[mv]en	semen seven
years	gexxx	[gpy]e[ars][rmo]s	gears germs pears pesos years
ago	xgx	[asu][gp][oas]	ago spa ups
our	xxx	[amorsvwz][rsaouv][msrwoax]	arm ass mar maw mom moo mow mum mus oar our ova ram raw row rum saw sax sos sow sum vow war was wax woo wow zoo
fathers	AxAAexx	[ACEGJMNPRSVWYbdfhlt][mraou][btlfdk][ltfbdhk]e[rsma][su]	Amblers Artless Cablers Callers Catters Coffers Cotters Cuffers Cullers Cutlers Cutters Emblems Gabbers Gadders Gaffers Gallers Gathers Goddess Godless Golfers Gulfers Gullers Gutless Gutters Jabbers Jobbers Jobless Jolters Jotters Jutters Madders Malters Matters Mobbers Molders Molters Mothers Mudders Muffers Mullers Mutters Nabbers Nodders Nutters Padders Pallers Palters Patters Podders Pollers Pothers Potters Puffers Pullers Putters Rafters Rathers Ratters Robbers Rollers Rotters Rubbers Rudders Rutters Salters Sobbers Sodders Solders Subbers Suffers Sulkers Vatters Wadders Wafters Walkers Wallers Wolfers Yakkers balders balkers ballers bathers batters bobbers bolters bothers budders buffers bulkers bullers butlers butters dabbers dodders doffers dollers dotters dubbers duffers dullers fablers fallers falters fathers fatters fobbers fodders folders fullers halters hatless hatters holders hollers huffers hullers hutters ladders ladlers lathers lobbers lofters lollers lotters lubbers luffers lullers tabbers tableau tablers talkers tatters tollers totters tubbers tufters
brought	AxxxgAA	[Wbdf]r[oa]ught	Wrought brought draught drought fraught

Ambiguity in lexicon

	Document Specific	Comprehensive
distinct words	149	246906
distinct tokens	136	176158
ratio	1.10	1.40
singletons	127 (93%)	147295 (84%)
remaining words	22	99611
ambiguous tokens	9	28863
ratio	2.44	3.45
character position ambiguity	3%	13%

Information retrieval

TREC

- Transliteration of search terms
- Character shape coding of document images
- Results poor with short index terms
  AxxA for "lost" mapped to >1000 terms
  Refinements since study have reduced this to *only* 682
  V_5 reduces it further to 76
- Results adequate for long index terms
  "industrial" is a singleton for all CSC versions
- Domain-specific lexicon would help a great deal
  reduced ambiguity

Information retrieval

Detectability of index terms

index term	tuned lexicon	comp lexicon
God	✓	
add	✓	
advanced	✓	✓
altogether	✓	✓
battlefield	✓	✓
birth		
brave	✓	
brought	✓	✓
cannot	✓	
cause	✓	
civil	✓	✓
conceived	✓	✓
consecrate	✓	✓
consecrated	✓	✓
continent	✓	✓
created	✓	
dead	✓	✓
dedicate	✓	✓
dedicated	✓	✓
detract	✓	
devotion	✓	
died	✓	✓
earth		
endure	✓	✓
engaged	✓	✓
equal	✓	✓
fathers	✓	
field	✓	
final	✓	
fitting	✓	
forget	✓	
forth	✓	
fought	✓	
freedom		
full	✓	
gave	✓	

index term	tuned lexicon	comp lexicon
government		
ground	✓	✓
hallow	✓	
highly	✓	✓
honored	✓	
increased	✓	✓
larger	✓	
liberty	✓	✓
lives	✓	
living	✓	
measure		
met		
nation	✓	
nobly	✓	✓
note	✓	
perish	✓	✓
poor	✓	
portion	✓	✓
power	✓	
proper	✓	
proposition		
remaining		
remember		
resolve	✓	
resting	✓	
score	✓	
sense	✓	
struggled	✓	
task	✓	
testing	✓	
unfinished	✓	✓
vain	✓	
war		
whether	✓	✓
world	✓	

Document Content

- Style
  - Word count
  - Word length
  - Proper nouns
  - Article usage
  - Sentence length
- Topic Identification
  - Generation of WST indices
  - Comparison of document indices against prototypes
- Part-of-speech tagging
  - Tag of WST is union of tags of all words represented by WST
  - gxexixxx = {premiums,previous}
  - POS(premiums) = plural noun
  - POS(previous) = adjective
  - POS(gxexixxx) = {plural noun,adjective}

Document reconstruction

- Substituting WST-indexed lexical entries

Four score and {never,seven} years ago {nor,our,war} fathers brought forth upon this continent a {men,new} nation conceived {in,is} liberty and dedicated to the gxxgxxixixx what all xxex are created equal

Now we are engaged {in,is} a great civil {nor,our,war} testing whether that nation {as,on,or,so,us} {any,say} nation {as,on,or,so,us} conceived and {as,on,or,so,us} dedicated can long endure We are xxeA {as,on,or,so,us} a great battlefield of that {nor,our,war}

AAe have cause to dedicate a portion of that field {as,on,or,so,us} a final resting place {far,for} those who here gave their lives that this nation xxigbA live It {in,is} altogether fitting and proper that we should do this

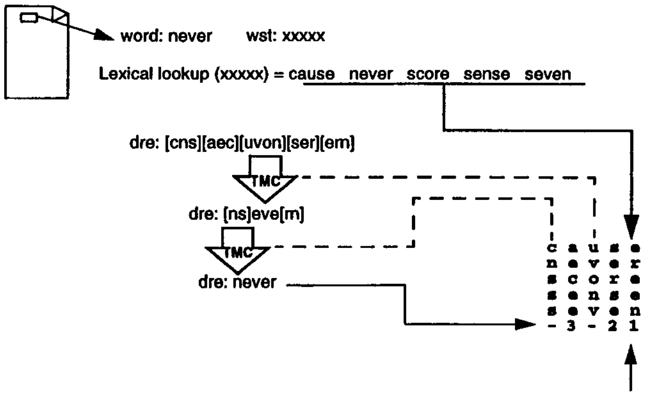
But {in,is} a larger sense we cannot dedicate we cannot consecrate we cannot hallow xbix ground the brave xxex living and dead who struggled here have consecrated it {far,for} above {nor,our,war} poor power to add {as,on,or,so,us} detract the world will AiAxAe note {nor,our,war} long xexxexxbex what we {any,say} here but it can {never,seven} forget xxbxA they did here

It {in,is} {far,for} {as,on,or,so,us} the living rather to be dedicated here to the unfinished work which they who fought here have thus {far,for} {as,on,or,so,us} nobly advanced It {in,is} rather {far,for} {as,on,or,so,us} to be here dedicated to the great task xexxxixixg before {as,on,or,so,us} that Axxxx these honored dead we take increased devotion {as,on,or,so,us} that cause {far,for} which they gave the {final,last} full xxexxxxe of devotion what we here highly resolve that these dead shall not have died {in,is} vain that this nation under God shall have a {men,new} bixxb of Axeedxxi and what gxxexxxxexx of the people by the people {far,for} the people shall not perish Axxxx xbix e {as,on,or,so,us} Ab

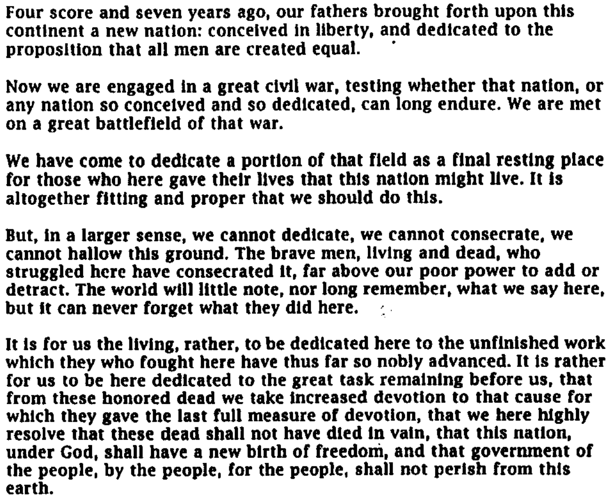
SABRE

ShApe Based word REcognition

- Character shape coding
- Word shape lookup
- Selective OCR
- Progressive resolution
- Over-recognition



- Redaction/Highlighting
  without ambiguity



Four score and seven years ago, our fathers brought forth upon this continent a new nation: conceived in liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battlefield of that war.

We have come to dedicate a portion of that field as a final resting place for those who here gave their lives that this nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we cannot dedicate, we cannot consecrate, we cannot hallow this ground. The brave men, living and dead, who struggled here have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember, what we say here, but it can never forget what they did here.

It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us, that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion, that we here highly resolve that these dead shall not have died in vain, that this nation, under God, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish from this earth.

Four score and seven years ago, our fathers brought forth upon this continent a new nation: conceived in liberty, and dedicated be the proposition what all men are created equal.

bow we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battlefield of that war,

We have come to dedicate a portion of that field as a final resting place for those who here gave their lives that this nation might live. Ft is altogether fitting and proper that we should do this.

but, in a larger sense, we cannot dedicate, we cannot consecrate, we cannot hallow this ground, the brave men, living and dead, who struggled here have consecrated it, far above our poor power be add or detract, the world will little note, nor long remember, what we say here, but it can never forget what they did here.

Ft is for us the living, rather, be be dedicated here be the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us, that four above honored dead we take increased devotion be that cause for which they gave the task full measure of devotion, that we here highly resolve that these dead shall not have died in vain, that this nation, under God, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish four this earth.

Postal addresses

- Tightly constrained lexicon
- Selective OCR for disambiguation

AxxAxx:

```
Aachen Aschau Auufer Bachra Bandau Bandow Barkow Bartow Bauler
Beckum Beelen Benken Berkau Beuden Bochow Bochum Bockau Borken
Borkow Borkum Borlas Borler Bredow Brehme Brehna Brodau Buchen
Buckau Buckow Burkau Buskow Carlow Cochem Daaden Dachau Damlos
Daskow Dechow Demker Derben Deuben Dorfen Drehna Duckow Eschau
Freden Frehne Gamlen Gartow Gerdau Geslau Gnadau Gorden Goslar
Grabau Graben Grabow Gruhno Gumtow Gustow Hartau Heeßen Herten
Horben Hosten Jeeben Jeetze Jucken Kantow Karben Karlum Kemtau
Kerben Kerkau Kerken Kerkow Kesten Konken Krahne Krakow Krebes
Krokau Krukow Kuchen Laaber Lachen Landau Lankau Laskau Lastau
Lauben Laufen Lauter Leetza Leuben Lochau Lochum Lostau Losten
Luchau Luckau Luckow Lunden Macken Manker Markee Marlow Mauden
Mechau Mechow Meeder Menden Merkur Mochau Mochow Muchow Neetze
Neußen Neufra Neuler Norden Norken Ornbau Panker Panten Parkow
Parlow Pastow Pechau Penkow Penkun Perkam Perlas Pockau Pomßen
Postau Pratau Preten Profen Puchow Rambow Randow Rantum Rastow
Reeßum Reußen Reuden Rochau Rockau Roskow Ruchow Saadow Saalau
Saalow Sachau Sandau Santow Saxler Seelen Seelow Seelze Semlow
Senden Sontra Suckow Tacken Tantow Techau Treben Trebra Trebur
Trebus Tuchen Usedom Verden Vreden Wachau Wachow Wacken Warder
Wardow Warlow Wasdow Wauden Wenden Werben Werdau Werder Werdum
Westre Xanten Zachow Zechow Zeetze Zerben Zeuden Zorbau Zuchau
```

AAAxxAxxxxx:

Altenhausen Attenhausen Ettenhausen Effenhausen

AAxAAgxxA:

Stuttgart

- Speed-up of 30x (130x for street names)

Multilingual Applications

- Additions to the CSC table

Characters	V_0	V_1	V_2	V_3	V_4	V_5
amorsuvxwz«»æ	x	x	x	x	x	x
n			n			
c		e			c	
e					e	
ACGIOQSTUVWXYZflt ÆøB	A			A	A	A
HMN				N		N
bhkL				b		b
BDEKR				E		E
PF				P		P
dJ				d		d
j	j					
įåáàâéêèîìíóòõôúùûñ ÅÁÀÂÉÊÈÎÌÍÓÒÕÔÚÙÛ¿¡	i					
gpqy	g					
äëöïüÄËÏÖÜ	U					

- Ambiguity

	English	French	German
distinct words	**246906**	**514637**	**316035**
distinct tokens (V_5)	**176158**	**296969**	**209011**
ratio	**1.40**	**1.73**	**1.51**
singletons	**147295 (84%)**	**225051 (76%)**	**155923 (75%)**
remaining words	**99611**	**289586**	**160112**
ambiguous tokens	**28863**	**71918**	**53088**
ratio	**3.45**	**4.03**	**3.02**
character position ambiguity	**13%**	**21%**	**25%**

- Thai

Summary

- Computationally efficient
- Robust to composition, resolution and noise
- Important information retained
- Supports many applications
- Combines with language models to support more applications

Papers

Measuring the Robustness of Character Shape Coding
Document Analysis Systems, 1998

Shape-based Word Recognition
International Journal of Document Analysis and Recognition, (in press)

Detecting Duplicate Documents
SPIE Symposium on Electronic Imaging Science and Technology, 1997

Moby Dick meets GEOCR: Lexical Considerations in Word Recognition
International Conference on Document Analysis and Recognition, 1997

Using Character Shape Codes for Information Retrieval
International Conference on Document Analysis and Recognition, 1997

Determination of the Script and Language Content of Document Images
IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997

An OCR Based on Character Shape Codes and Lexical Information
International Conference on Document Analysis and Recognition, 1995

Document Reconstruction: A Thousand Words from One Picture
Symposium on Document Analysis and Information Retrieval, 1995

Document Image Assessment and Enhancement

A Method for Restoration of Low-Resolution Text Images

Paul D. Thouin
Department of Defense, Fort Meade, MD 20755, U.S.A.
Chein-I Chang
Remote Sensing Signal and Image Processing Laboratory
Department of Computer Science and Electrical·Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250, U.S.A.

Abstract

Image restoration using resolution expansion is important in many areas of image processing. This paper introduces a restoration method for low-resolution text images which produces expanded images with improved definition. This technique creates a strongly bimodal image with smooth regions in both the foreground and background, while allowing for sharp discontinuities at the edges. The restored image, which is constrained by the given low-resolution image, is generated by iteratively solving a nonlinear optimization problem. Text images restored using this technique are shown to be both quantitatively and qualitatively superior to images expanded using the standard methods of linear interpolation and cubic spline expansion.

1 Introduction

Text image resolution expansion has become increasingly important in a number of areas of image processing. Optical Character Recognition (OCR) of document images continues to be of great importance as we attempt to become a paperless society. Restoring text from video surveillance imagery is often crucial to law enforcement agencies. Digital video compression algorithms can also benefit from successful text resolution expansion techniques. Common methods of interpolation, which were not designed specifically for text images, typically smooth over the important details and produce inadequate expansion. This paper proposes a new nonlinear restoration technique for text images, which creates smooth foreground and background regions while preserving sharp edge transitions. Numerous restoration methods have been published in the literature [1]-[6]. Linear interpolation tends to smooth the image data at transition regions and results in a high-resolution image that appears blurry. Cubic spline expansion allows for sharp transitions, but tends to produce a ringing effect at these discontinuities. The proposed method overcomes these limitations and produces qualitatively superior images with expanded resolution. The criteria of mean squared error (MSE) is used to quantitatively demonstrate that images restored using this new method are more accurate than images produced by existing techniques.

The goal of resolution expansion is to create an expanded image with improved definition from observed low-resolution imagery. Acquisition of this low-resolution imagery can be modeled by averaging a block of pixels within a high-resolution image. Resolution expansion is an ill-posed inverse problem. For a given low-resolution image, a virtually infinite set of expanded images can be generated by the observed data. To solve for a high-resolution image that is optimal in some sense, a Bimodal-Smoothness-Average (BSA) score is introduced to measure how well potential expanded images exhibit desirable text-like characteristics. The BSA score is defined as the weighted sum of separate bimodal, smoothness, and average measures. Minimization of this score is performed using a nonlinear optimization technique which results in a strongly text-like image. Text images typically have bimodal distributions with large black and white peaks and images restored using this new method are strongly bimodal as well. Images of text are also usually smooth in both the foreground and background regions with sharp transitions only at the edges. In addition, expanded images are constrained so the average of a group of high-resolution pixels is close to the original value of the low-resolution pixel from which they were derived. Text images restored using this new BSA score-based technique are shown to be both quantitatively and qualitatively superior to images expanded using standard methods.

The remainder of this paper is organized as follows. Section 2 describes the problem of image resolution expansion. In Section 3, three text scoring functions are introduced which exploit properties

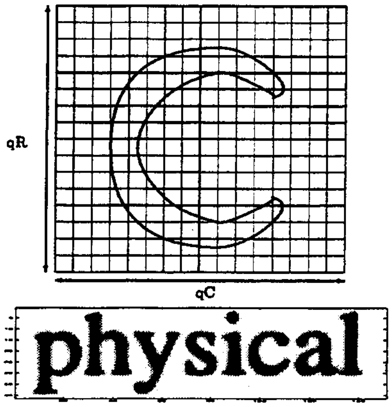


Figure 1: High-resolution imaging system

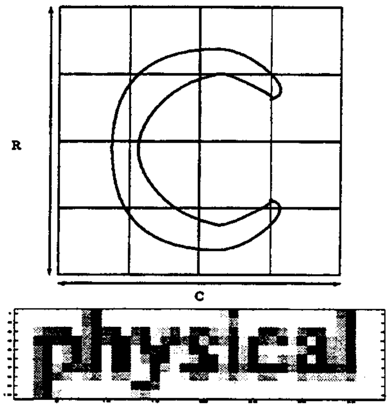


Figure 2: Low-resolution imaging system

of text images and set a foundation for the image restoration method proposed in this paper. Section 4 derives the technique used to iteratively solve for the functional minimum that results in the restored image. Section 5 presents experiments using this technique and quantitatively compares the proposed method to other methods of image resolution expansion. Finally, a summary of this proposed technique is given in Section 6.

2 Problem Statement

The image acquisition process consists of converting a continuous image into discrete values obtained from a group of sensor elements. Each sensor element produces a value which is a function of the amount of light incident on the device. For 8-bit grayscale quantization, the allowable range of values for each sensor are integers from 0 (black) to 255 (white). The sensors are typically arranged in a non-overlapping grid of square elements, smaller elements result in higher resolution imagery. Shown in Fig. 1 is a high-resolution imaging system where the number of sensors is adequate to represent the desired text image. The majority of pixels within the image are either white or black, with a small number of gray pixels occurring at the edges. Fig. 2 illustrates a low-resolution imaging system where the number of sensors has been reduced by a factor of $q = 4$ in both the horizontal and vertical directions. This low-resolution acquisition results in significant blockiness and is insufficient to accurately represent this image. Each sensor element effectively averages the image within its section of the grid, resulting in an increased amount of gray pixels. Low-resolution imaging can therefore be thought of as block-averaging high-resolution images.

The problem addressed in this paper is to restore the high-resolution image $HI_{qr,qc}$ given only the low-resolution image $LI_{r,c}$, where r and c are the number of rows and columns in the low-resolution image and q is the resolution expansion factor. The image acquisition process of obtaining $LI_{r,c}$ from $HI_{qr,qc}$ is given by

$$LI_{r,c} = \frac{1}{q^2} \sum_{s=q\cdot r}^{(qr+q-1)} \sum_{t=q\cdot c}^{(qc+q-1)} HI_{s,t} \qquad (1)$$

The value of $LI_{r,c}$ is the average of the high-resolution pixels within the $q \times q$ neighborhood. Eq. (1) represents a typical image restoration problem where we are required to restore the $HI_{qr,qc}$ based on the observed $LI_{r,c}$ via the relationship described by this equation. Since there are a great number of high-resolution images which may satisfy the constraint of the observed low-resolution image given by Eq. (1), image restoration is generally an ill-posed inverse problem.

3 The BSA Scoring Functions

The scoring function introduced in this paper is designed to measure how well a group of pixels within an image represent the desired properties of text. This function, referred to as the BSA scoring function, is expressed as the weighted sum of a bimodal score B, a smoothness score S, and an average score A, each of which will be discussed in detail in the remainder of this section. The BSA score is defined as

$$BSA(x) = \lambda_1 B(x) + \lambda_2 S(x) + \lambda_3 A(x) \qquad (2)$$

where x is a block of pixels and λ_1, λ_2, and λ_3 are Lagrange multipliers. Our goal is to design a BSA-based algorithm which can iteratively solve for the block of pixels x that minimizes the $BSA(x)$ score given by Eq. (2). The BSA score is a function of the bimodal, smoothness, and average scores which are discussed in detail in the following three subsections.

3.1 The Bimodal Score

The typical distribution of a text image contains two peaks, a large one at μ_{white}, which normally represent the page's background, and a secondary peak at μ_{black} representing the foreground text. From the histogram of the given low-resolution text image, estimates of the means for the black and white distributions are calculated. These means are used to compute the bimodal score $B(x)$, which measures how far an image block x is from bimodal. The bimodal score used in this paper is defined by

$$B(x) = \sum_{r,c}(x_{r,c} - \mu_{black})^2(x_{r,c} - \mu_{white})^2 \qquad (3)$$

where r and c are the row and column indices within the block being evaluated.

When a pixel value within x is close to either μ_{black} or μ_{white}, its contribution to $B(x)$ is minimal. The bimodal minimum score of $B(x) = 0$ means that the image is perfectly bimodal, the value of every pixel is equal to either μ_{white} or μ_{black}. Solving for the block of pixels x that minimizes $B(x)$ produces a strongly bimodal image, which is one of the desired properties of this proposed text restoration technique. The estimated means of the bimodal distribution, μ_{black} and μ_{white}, are determined a priori.

3.2 The Smoothness Score

With the exception of edges, text images tend to be very smooth in both the foreground and background regions which results in neighbors with similar values. A smoothness score, which is computed for each block of pixels, is introduced to measure this feature. For this proposed algorithm, a simple statistic using only the four nearest neighbors of each pixel is used. Other more sophisticated smoothness measures could be implemented as well. The smoothness score $S(x)$ used by this technique is given by

$$S(x) = \sum_{r,c}[(x_{r-1,c} - x_{r,c})^2 + (x_{r,c-1} - x_{r,c})^2 + (x_{r,c+1} - x_{r,c})^2 + (x_{r+1,c} - x_{r,c})^2] \qquad (4)$$

where r and c are the row and column indices within the block being evaluated. The minimum value of $S(x) = 0$ occurs when all pixels have identical values.

3.3 The Average Constraint Score

It is reasonable to require that the average of a group of high-resolution pixels is close to the original value of the low-resolution pixel from which they were derived. For each block of low-resolution pixels, an average score $A(x)$ is used to measure how well the restored high-resolution pixels meet the average constraint imposed by their corresponding low-resolution pixels. The average score for a 2×2 block is expressed by

$$A(x) = \sum_{i=1}^{4}[\mu_i - \frac{1}{q^2}\sum_{r=1}^{q}\sum_{c=1}^{q} x^{(i)}_{r,c}]^2 \qquad (5)$$

where i is the index for the low-resolution pixels, μ_i is the value of each low-resolution pixel, and $x^{(i)}_{r,c}$ are the restored high-resolution pixels corresponding to pixel μ_i. The initial high-resolution image formed by using pixel replication always has an average score of zero because it satisfies the constraint.

4 Solving for the Restored Image

The goal of the restoration algorithm is to solve for the image block that minimizes the scoring function $BSA(x)$ introduced in Eq. 2. Throughout this paper, a block x is defined both as a group of 4×4 low-resolution pixels and as the $4q \times 4q$ high-resolution pixels that are derived from them. The 4×4 size was specifically chosen because it contains enough pixels to adequately measure text characteristics but is not too large to be computationally burdensome. The goal of resolution enhancement is to create a restored image with improved resolution.

Pixel replication, where every value within a $q \times q$ neighborhood is identical to the corresponding low-resolution pixel, is used for the initial expansion. Each $4q \times 4q$ block of high-resolution pixels is restored independently using iterative optimization techniques described in this section to solve for the block which minimizes the BSA score. At each iteration, the first and second partial derivatives of the BSA scoring function are used to determine the image update. To avoid block boundary discontinuities only the center $3q \times 3q$ pixels are updated. The entire image is therefore divided into blocks that overlap by one quarter, or $q \times 4q$ pixels, and can be restored independently. This iterative minimization of the BSA score continues until convergence is reached resulting in the restored image.

Initially, each $4q \times 4q$ block of pixels x is converted to a $(4q)^2$-long vector $\vec{x}$ using raster scanning,

$$\vec{x}[q(r-1)+c] = x(r,c) \qquad \text{for} \qquad 1 \leq r, c \leq q \quad (6)$$

A small distance away from $\vec{x}$ the BSA function can be represented by its second order Taylor series approximation [7],

$$BSA(\vec{x}+\vec{\delta}) \approx BSA(\vec{x}) + [\nabla BSA(\vec{x})]\vec{\delta} + \frac{1}{2}\vec{\delta}^T H \vec{\delta} \quad (7)$$

and the change in BSA is given by

$$\Delta BSA = [\nabla BSA(\vec{x})]\vec{\delta} + \frac{1}{2}\vec{\delta}^T H \vec{\delta} \qquad (8)$$

where $\vec{\delta}$ is the small change to the image vector $\vec{x}$, $\nabla BSA(\vec{x})$ is the gradient, and H is the Hessian matrix.

Since the Hessian matrix is symmetric, only half of the matrix needs to be computed. To maximize the function $BSA(\vec{x})$, the variables in the Hessian matrix are first made independent. To do this the Hessian is diagonalized using a similarity transform. Each eigenvector of the Hessian matrix is placed in a separate column to form a unitary eigenmatrix E. That is, the product of the eigenmatrix with its transpose is equal to the identity matrix $EE^T = I$. When the Hessian matrix is pre-multiplied by the transposed eigenmatrix and post-multiplied by the eigenmatrix, the resulting matrix $E^T HE$ is diagonal. Because the Hessian is real and symmetric, it is always diagonalizable.

The Taylor series approximation to the change in the scoring function $\triangle BSA$ can now be expressed in terms of the $(4q)^2 \times (4q)^2$ Hessian matrix H, its $(4q)^2 \times (4q)^2$ eigenmatrix E, the $1 \times (4q)^2$ gradient of the scoring function $\nabla BSA(\vec{x})$, and the $(4q)^2 \times 1$ small change in the image vector $\vec{\delta}$,

$$\begin{aligned}\triangle BSA = {} & ([\nabla BSA(\vec{x})]E)(E^T\vec{\delta}) \\ & +\frac{1}{2}(\vec{\delta}^T E)(E^T HE)(E^T\vec{\delta})\end{aligned} \tag{9}$$

With the following substitutions,

$$\nabla BSA'(\vec{x}) = [\nabla BSA(\vec{x})]E \tag{10}$$

$$\vec{\delta'} = E^T\vec{\delta} \tag{11}$$

$$H' = E^T HE \tag{12}$$

Eq. (9) can be simplified to

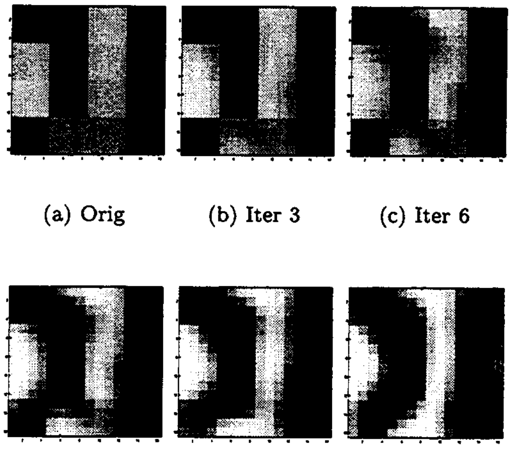
$$\triangle BSA = [\nabla BSA'(\vec{x})][\vec{\delta'}] + \frac{1}{2}\vec{\delta'}^T H\vec{\delta'} \tag{13}$$

The functional minimum is achieved by stepping in the direction

$$\vec{\delta'} = \frac{\nabla BSA'(\vec{x})}{|H'|} \tag{14}$$

in the transformed domain, which is simply $\vec{\delta} = E^T\vec{\delta'}$ in the pixel domain. For each iteration, the image update $\vec{\delta'}$ is determined. The iterations continue until convergence is reached, resulting in a desired restored image.

An example of this iterative image restoration process is shown in Fig. 3. The original 4×4 block of pixels is expanded by a factor of $q = 4$ using pixel replication to produce a 16×16 high-resolution image shown in Fig. 3(a). As the iterative restoration process proceeds in Figs. 3(b-f), the image becomes more bimodal and smooth resulting in a greatly improved image. The majority of gray pixels that occur between characters are replaced with either black or white values, resulting in a strongly bimodal distribution. The resulting image is also smooth in both the foreground and background regions while maintaing the constraint that the average of each 4×4 block of high-resolution pixels is close to the original value of each corresponding low-resolution pixel.



(a) Orig (b) Iter 3 (c) Iter 6

(d) Iter 10 (e) Iter 15 (f) Iter 30

Figure 3: Iterative text restoration example

Minimization of the BSA score produces a restored image that is the optimal combination of these bimodal, smoothness, and average measures.

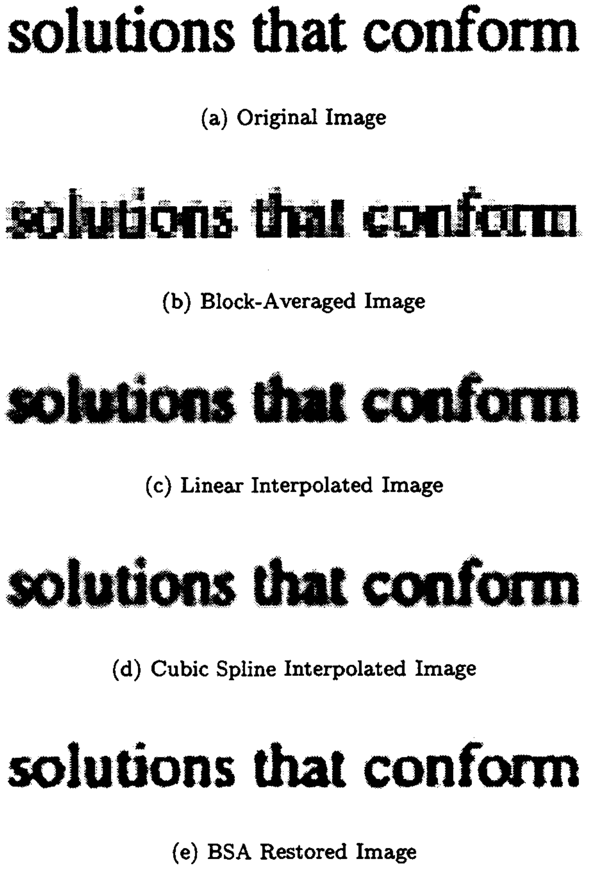
5 Experimental Results

To quantitatively measure image restoration success, low-resolution images were created by block-averaging images as described by Eq. (1). Restored images are then compared with the original to determine the success of restoration numerically. The mean squared error (MSE) was used to compare the various methods of image resolution expansion. The definition of mean squared error used in this paper is

$$MSE = \frac{1}{RC}\sum_{r=1}^{R}\sum_{c=1}^{C}(original_{r,c} - restored_{r,c})^2 \tag{15}$$

where R and C are the number of rows and columns in the images.

The proposed BSA restoration algorithm was compared to several common expansion methods, including pixel replication, linear interpolation, and cubic spline expansion. In linear interpolation, a linear fit is calculated between all pixels within each column, and then repeated for all pixels within each row. These images naturally tend to be smooth, without sharp discontinuities, producing blurry results. Cubic spline expansion [8] approximates the given discrete low-resolution pixels as a smooth continuous curve obtained from the weighted sum of cubic spline basis functions and resamples the curve to obtain the high resolution image. This method allows for sharp edges but often overshoots at these discontinuities, producing a ringing effect. The BSA text restoration technique creates smooth fore-



(a) Original Image

(b) Block-Averaged Image

(c) Linear Interpolated Image

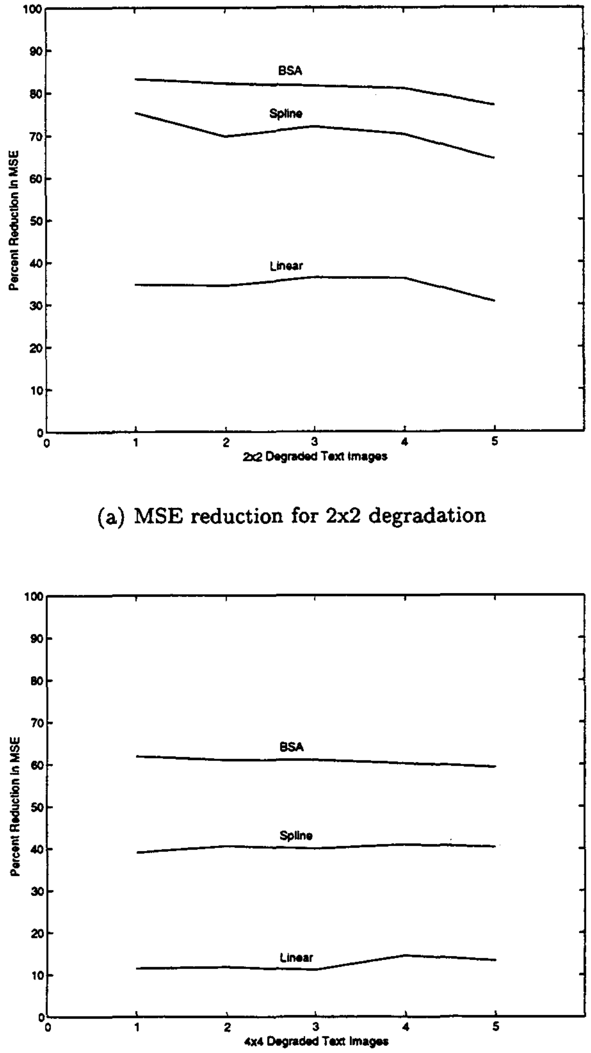
(d) Cubic Spline Interpolated Image

(e) BSA Restored Image

Figure 4: Text restoration results

ground and background regions and permits sharp edges at transition regions, while maintaining the low-resolution average constraint. Images restored with this technique are shown to be both qualitatively and quantitatively superior to other common resolution expansion methods. Restoration results for a severely degraded 41 × 376 section of a 3300 × 2544 image scanned at 300 dpi are shown in Fig. 4. An averaging factor of $q = 4$ was used to create a blocky image with a significant amount of touching characters shown in Fig. 4(b). The mean squared error between this blocky section and the original is 1129.1. Linear interpolation produces the severely blurred image in Fig. 4(c) which reduces the MSE only slightly by 14.5%. The resulting image obtained from cubic spline interpolation in Fig. 4(d) is significantly improved with a 40.9% reduction in MSE, but is still somewhat blurry. The BSA restoration produced the best image shown in Fig. 4(e) by reducing the MSE by 60.2%.

These results clearly demonstrate several advantages of this technique that was designed specifically for text images. The BSA-restored image in Fig. 4(e) is strongly bimodal and has both smooth foreground and background regions. There are sharp discontinuities at the edges which are not observed in the linear or cubic spline expansion results. The significant reduction in the amount of gray pixels produces superior character separation evident in the word "conform". This gray pixel decrease also frequently results in sharper contrast within a single character. The hole in the character "a" in the word "that" which is hardly apparent in the block-averaged image is vastly improved by the algorithm. A comparative study of the reduction in mean squared error for the various image expansion techniques is plotted in Fig. 5. Shown in Fig.5(a) are results for a group of five full-page images that were scanned at 300 dpi and degraded with block-averaging factor $q = 2$. Linear interpolation reduced the MSE by an average of 34.4% for these images. Cubic spline expansion performed much better by



(a) MSE reduction for 2x2 degradation

(b) MSE reduction for 4x4 degradation

Figure 5: Comparison of restoration techniques

reducing the MSE by an average of 70.3%. The proposed BSA restoration technique was the most accurate for these images and resulted in an average 80.9% reduction in mean squared error. Shown in Fig. 5(b) are restoration results for a second group of five images again scanned at 300 dpi and severly degraded by $q = 4$ block-averaging. Expansion using linear interpolation reduced the MSE by an average of 12.4% and cubic spline expansion resulted in a 40.2% reduction average. The best results were obtained using the BSA algorithm which reduced the mean squared error by an average of 60.7%.

6 Conclusions

In this paper, we present a new resolution expansion technique for the restoration of grayscale text images. Bimodal, smoothness, and average (BSA) scores that measure desired properties observed in text images were introduced and combined to form a single scoring function. The restored image obtained by solving this nonlinear optimization problem is one which is strongly bimodal and smooth, while satisfying the average constraint score. The proposed BSA restoration technique was shown to be both qualitatively and quantitatively superior to the existing linear interpolation and cubic spline expansion techniques.

In order to further improve the proposed BSA restoration technique, new measures may be added that make use of a priori knowledge. If the scanning resolution and font size within an image are known, the average stroke width in pixels can be computed. A new score could be created to measure how close a group of pixels is to this desired width. Additionally, different measures for the existing bimodal, smoothness, and average scores could be implemented. Another research area under exploration is optimally selecting the weights for the three scoring functions, which could potentially be of significant benefit.

Acknowledgements

The first author wishes to thank G. H. Nickel of Los Alamos National Laboratory and C.S. Cumbee of the Department of Defense for technical discussions and S. J. Dennis of the Department of Defense for his support of this research.

References

[1] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, No. 6, pp. 1153-1160, 1981.

[2] T. C. Chen and R. J. P. de Figueiredo, "Image decimation and interpolation techniques based on frequency domain analysis," *IEEE Transactions on Communications*, Vol. 32, No. 4, 1984.

[3] A. D. Kulkarni and K. Sivaraman, "Interpolation of digital imagery using hyperspace approximation," *Signal Processing*, Vol. 7, pp. 65-73, 1987.

[4] V. S. Nalwa, "Edge-detector resolution improvement by image interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 3, pp. 446-451, 1987.

[5] N. B. Karayiannis and A. N. Venetsanopoulos, "Image interpolation based on variational principles," *Signal Processing*, Vol. 25, pp. 259-288, 1991.

[6] R. R. Schultz and R. L. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Transactions on Image Processing*, Vol. 3, No. 3, pp. 233-242, 1994.

[7] J. Skilling and R. K. Bryan, "Maximum entropy image reconstruction: general algorithm," *Mon. Not. Royal Astronomy Society*, Vol. 211, 1984, pp. 111-124.

[8] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 26, No. 6, December 1978.

OCR of Degraded Documents using HMM-Based Techniques

**Issam Bazzi, Premkumar Natarajan, Richard Schwartz,
Andras Kornai, Zhidong Lu, John Makhoul**

BBN Technologies, GTE Corporation
70 Fawcett Street, Cambridge MA, 02138
ibazzi@bbn.com

Abstract

We present an OCR system for handling degraded documents, such as faxed text. The basic system utilizes the BBN BYBLOS OCR system, which uses a Hidden Markov Model (HMM) approach for training and recognition. To handle degraded documents, we present two approaches, which can be applied individually or jointly. In the first approach, we train the system on documents that exhibit the expected kind of degradation. For example, to perform OCR on fax documents, we train the system on fax data. In the second approach, the system performs unsupervised adaptation on each page to be recognized in such a way as to maximize a desired objective function. Several objective functions were attempted: Maximum Likelihood Linear Regression (MLLR), Maximum a Posteriori (MAP), and Leave-One-Out MAP. We report on results using the above approaches on fax text images generated from the University of Washington English Image Database I. Applying adaptation techniques, in addition to training on fax, we have reduced the character error rate by a factor of three from the base condition.

1 Introduction

In earlier papers [6,8], we presented an HMM-based OCR system referred to as the BBN BYBLOS OCR system in this paper, incorporating the BBN BYBLOS continuous speech recognition system. The BYBLOS OCR system uses a character model trained on a corpus of text images, a lexicon and a grammar. A brief review of the BYBLOS OCR system is provided in the following section.

While our earlier papers reported on the performance of the BYBLOS OCR system on data from the University of Washington English Image Database I (UW corpus) [9], in this paper we present techniques for dealing with degraded documents within the framework of the BYBLOS OCR system. The accuracy of OCR systems is fundamentally dependent upon the quality of the scanned text image [3]. Common document processing operations such as faxing induce significant degradations in image quality. Part of the degradation is due to the low resolution scanning devices in fax machines and part of the degradation is from the printing process. Sometimes transmission noise adds to the degradation. While existing literature contains examples of algorithms for processing the degraded image to enhance quality [2], this paper focuses on model-based techniques for handling the degradations within the OCR system. For our recognition experiments on degraded data we have used fax-degraded documents generated from *clean* documents in the UW corpus.

The paper is organized as follows. In section 2 we provide a brief review of the BYBLOS OCR system along with some background information. In section 3 we present recognition results using a model trained on degraded documents as well as the results obtained using a system trained on clean data. In Section 4 we discuss and demonstrate the use of adaptation to further improve recognition accuracy. A summary and conclusion in Section 5 follow this.

2 System Overview

This section gives a brief review of the BBN BYBLOS OCR system. For a more detailed description the reader is referred to [6]. A pictorial representation of the system is given in Fig. 1. In the figure, knowledge sources are depicted by ellipses and are dependent on the particular language or script. The OCR system components themselves are identified by rectangular boxes and are independent of the particular language or script. Thus, the same OCR system can be configured to perform recognition on any language.

At the top level, the OCR system can be subdivided into two basic functional components:

training and recognition. Both, training and recognition share a common pre-processing and feature extraction stage. The pre-processing and feature extraction stage starts off by first deskewing the scanned image and then locating the positions of the text lines on the deskewed image.

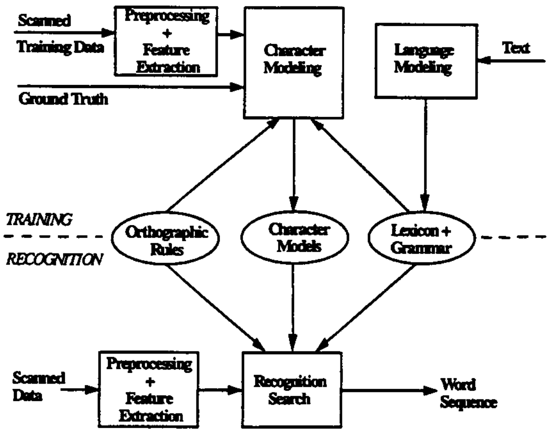


Figure 1: Block diagram of BBN BYBLOS OCR system

The feature extraction program computes a feature vector as a function of the horizontal position within a line, see Fig. 2. First, each line of text is horizontally segmented into a sequence of thin, overlapping, vertical strips called frames (one frame is shown in Fig. 2). For each frame we then compute a language-independent, feature vector that is a numerical representation of the frame.

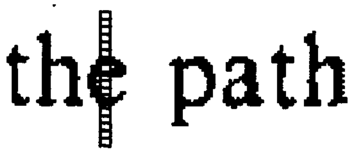


Figure 2: Feature extraction on a line of English text

The OCR system models each character with a multi-state, left-to-right HMM. Each state has an associated output probability distribution over the features. The number of states and the allowable transitions are system parameters that can be set. For our experiments we have used 14-state, left-to-right HMMs with the topology shown in Fig. 3. Training is performed using the Baum-Welch or Forward-Backward algorithm, which aligns the feature vectors with the character-models to obtain maximum likelihood estimates of HMM parameters.

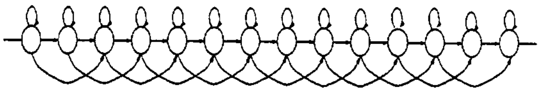


Figure 3: Figure shows a 14-state, left-to-right HMM with self-loops and skips

For our system the HMM parameters are the means and variances of the component gaussians in the gaussian mixture model of the state output probabilities, the mixture component weights and the state transition probabilities. On the other hand during recognition we search for the sequence of characters that is most likely given the feature-vector sequence and the trained character-models, in accordance with the constraints imposed by a lexicon and/or a statistical grammar. The use of a lexicon during recognition is optional but its use generally results in a lower Character Error Rate (CER). The lexicon is estimated from a suitably large text corpus. Typically the grammar (language model), which provides the probability of any character or word sequence, is also estimated from the same corpus.

A significant advantage of HMM-based systems is that they provide a language-independent framework for training and recognition. At the same time, they do not require the training data to be segmented into words or characters, i.e., they automatically train themselves on non-segmented data.

3 Training on Fax

The most straightforward way to improve recognition performance on degraded data is to train on data that has been subjected to a similar degradation process. In this section we present our results with fax-degraded data.

3.1 Parallel Fax Corpus

For our English OCR experiments we used data from the UW corpus. The UW corpus consists of 958 pages scanned from technical articles containing more than 11000 zones of text. For our experiments we randomly selected 95 zones for training and 36 zones for testing. To generate the fax-degraded documents, the selected documents from the UW corpus were first printed on paper. The printed images were then faxed from one plain paper fax machine to another and the faxed images were scanned into bitmaps on the computer. The procedure used for generating the faxed data resulted in fax-degraded training and test corpora that mirrors the clean data from the UW English database. Also, this design allows us to compare the recognition results on the degraded documents with the results on corresponding *clean* documents from the UW corpus.

3.2 Fax Training Results

In our first set of experiments we trained the system using three different training data sets: clean data alone, fax data alone and a mixture of the clean and fax data. For each training condition we tested the system on both clean and faxed data. The output of the recognizer was compared with the reference transcriptions and the average character error rate (CER) was measured by adding the number of substitutions, deletions and insertions to obtain the total number of errors, and then dividing the total number of errors by the total number of characters in the reference transcriptions. The CER's for different training conditions are listed in Table 1.

Table 1: Character Error Rates Under Different Training Conditions

Training	CER % (Clean Test)	CER % (Fax Test)
Clean Only	0.6	5.3
Clean + Fax	0.6	2.7
Fax Only	1.0	2.2

For the model trained on clean data alone, the CER on the fax data, 5.3%, is about nine times higher than the CER on clean data, 0.6%. By training the system on the fax training data we were able to bring down the error rate on the fax test data from 5.3% to 2.2%. At the same time the CER on clean data increased from 0.6% to 1.0%. With the aim of restoring the performance on clean data while maintaining the improved accuracy on the fax data we trained our system on a mix of the clean and fax training. Using this system we achieved a CER of 0.6% on clean data and 2.7% on faxed data.

The fact that a model trained on fax data alone yields a CER of 2.2% on the fax test set while the model trained on clean data alone yields a CER of 0.6% on the clean test set indicates that the recognition of fax documents is an inherently more difficult problem than the problem of recognizing clean documents. In the next section we discuss the use of powerful adaptation techniques to further improve the accuracy of recognition on fax-degraded documents.

4 Adaptation

Adaptation is the process of adjusting the parameters of an initial trained model so as to improve performance on a particular document. Adaptation techniques for HMMs have been used earlier by researchers in the speech community [4,5]. For example in speech recognition systems, a speaker-independent (SI) model is first trained on speech data from many speakers. At recognition time, the SI model is then adapted to each speaker to better model the finer variations for that particular speaker. Similarly for OCR, we can first train a Document Independent (DI) model on data from many documents. We may then adapt the parameters of this DI model using adaptation data for a particular document in such a way as to improve the recognition accuracy for that document.

Adaptation techniques can be broadly divided into two categories: supervised adaptation and unsupervised adaptation. In supervised adaptation the character transcriptions for the adaptation data are provided whereas in unsupervised adaptation we first use the DI model to recognize the document and then use the errorful, recognized text as the transcriptions for the adaptation data. Using the adapted model we can then recognize the document again, typically with higher accuracy.

In the following we present a brief description of unsupervised adaptation using two popular objective functions, the Likelihood function and the Posterior probability. The technique based on maximizing the Likelihood function of the adaptation data is referred to as the Maximum Likelihood Linear Regression (MLLR) technique while the technique based on maximizing the posterior probability of the adaptation data is referred to as the Maximum A Posteriori (MAP) technique. In both cases, the technique must deal with the fact that we do not have sufficient data to re-estimate all the parameters.

4.1 MLLR Adaptation

The MLLR technique handles the data-insufficiency problem by first segmenting the gaussians into a few distinctive sets and then inferring a shared transformation for all the gaussians in each set. In our MLLR adaptation program we only re-estimate the means of the gaussians. Thus, we do not re-estimate transition probabilities, mixture component weights or mixture component covariance's and these parameters take their values from the original model set. Mathematically the MLLR method is described as follows.

$$\lambda_{mllr} = \text{argmax}_{\lambda}\, P(X|\lambda)$$

where λ is the model parameter vector, X is the observation vector and λ_{mllr} is the model parameter vector that maximizes the likelihood function, $P(X|\lambda)$.

4.2 MAP Adaptation

Unlike the MLLR technique, the MAP approach deals with the lack of data by incorporating prior information into the adaptation process. Usually the prior knowledge is obtained from a prior training stage where prior distributions are estimated for the parameters that are to be adapted. The MAP technique infers a separate transformation for each gaussian in the model. Since a larger number of transformations are estimated, MAP adaptation typically requires more data than the MLLR technique. In addition the MAP technique also imposes the additional burden of training the prior distributions for the model parameters to be re-estimated.

Mathematically the MAP procedure is described as follows,

$$\lambda_{map} = \text{argmax}_{\lambda}\, P(\lambda|X)$$

where λ is the unknown model parameter vector, X is the observation vector and λ_{map} is the model parameter vector that maximizes the posterior probability $P(\lambda|X)$. The Posterior probability, $P(\lambda|X)$, may be computed using Bayes' Law as follows:

$$P(\lambda|X)=P(X|\lambda)\, P(\lambda)/ P(X)$$

and since the observation probability, P (X), is independent of λ, we may rewrite the MAP equation as,

$$\lambda_{map} = \text{argmax}_{\lambda}\, [P(X|\lambda)\, P(\lambda)]$$

For any document, the resulting MAP adapted model is an interpolation between a DI model trained on data from many documents and the document dependent (DD) model trained on data from that particular document. The more adaptation data available for the particular document, the closer the MAP model is to the DD model. It is typical to have more adaptation data available in OCR than in speech. For example, while using OCR on a whole book or newspaper, a large amount of adaptation data becomes available in an incremental fashion. For our experiments we implemented adaptation for each text zone separately.

4.3 Results with Adaptation

For our first set of adaptation experiments we started off by using the model trained on the mix of clean and fax data as our Document Independent (DI) model. We performed a first pass of recognition using the DI model and used the recognition results of the first pass as input to the adaptation program. The results of the experiments are listed in Table 2. As can be seen from the figures in Table 2, MLLR is seen to be better than MAP adaptation; a conclusion that is counter-intuitive. A detailed analysis of the errors showed that the MAP adapted model tended to repeat the recognition errors made in the first pass of recognition, indicating that the MAP adaptation was overfitting the models to the adaptation data thereby *memorizing* the errors. To get over this problem we devised a different strategy, called the **Leave-One Out MAP** technique, for implementing MAP adaptation. In the Leave-One Out technique adaptation is done at the sentence (just one line of text, in this context) level and the adaptation data consists of all the sentences except the one to be recognized.

Table 2: Results of Using Adaptation Techniques on Fax Data

Training Data/ AdaptationCondition	CER % (Fax)
Clean + Fax / No Adaptation	2.7
Clean + Fax / MAP	2.5
Clean + Fax / MLLR	2.2
Clean + Fax / Leave-One-Out MAP	2.1

From Table 2, it can be seen that the Leave-One Out MAP technique is indeed much better than the basic MAP technique and slightly better than the MLLR adaptation technique. The MLLR technique, on the other hand, is easier to implement and computationally much less expensive than the MAP. Based on this we have settled on the use of the MLLR technique as the algorithm of choice for our adaptation system.

For our final adaptation experiment we used the model trained on the fax data alone as the initial DI model and used the MLLR adaptation technique to adapt the models. As before, we ran one pass of recognition using the DI models and used the recognized text as the adaptation data. After adaptation the error rate decreased from 2.2% to 1.7%. The results of this experiment along are listed in Table 3.

Table 3: Summary of Results with and Without Adaptation

Training Data / Adaptation Method	CER% (Fax)
Fax only / No Adaptation	2.2
Fax only / MLLR Adaptation	1.7

5 Conclusions

In this paper we have addressed the problem of performing character recognition on fax degraded documents. The techniques presented are general and may be applied to other kinds of degradations or noise environments. We have demonstrated that HMM-based systems can easily be trained to model different degradations by using a properly chosen training data set. As indicated by experimental results, a properly chosen training set can significantly reduce the recognition error rate.

Another useful aspect of HMM-based systems is the possibility of adapting the models to a particular document. Experimental results presented in the paper indicate the substantial improvement that adaptation can offer. A comparative analysis of two basic adaptation techniques, MAP adaptation and MLLR adaptation, is provided. The Leave-One Out MAP technique is marginally superior to the MLLR method but the small gain in accuracy does not justify the added complexity and computational expense. As such, the MLLR method is the adaptation technique of choice for our system.

6 References

[1] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, Vol. 77, No. 2, pp. 257-286, Feb. 1989

[2] J.D. Hobby and H.S. Baird, "Degraded Character Image Restoration," *Fifth Annual Symposium on Document Analysis and Information Retrieval*, Alexis Park Resort, Las Vegas, Nevada, pp. 233-245, April 15-17, 1996

[3] S.V. Rice, J. Kanai, T.A. Nartker, "An Evaluation of OCR Accuracy,"In *Information Science Research Institute, 1993 Annual Research Report,* University of Nevada, Las Vegas, pp. 9-20, 1993

[4] J.L. Gauvain and C.H. Lee, "Maximum-a-posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,"In *IEEE Transactions on Speech and Audio Processing,* Vol. 2, pp. 291-298, 1994

[5] C.J. Leggetter and P.C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs," In *Computer Speech and Language,* Vol. 9, pp. 171-186, 1995

[6] R. Schwartz, C. LaPre, J. Makhoul, C. Raphael, and Y Zhao, "Language-Independent OCR Using a Continuous Speech Recognition System," *Proc. Int. Conf. on Pattern Recognition,* Vienna, Austria, pp. 99-103, August 1996

[7] I. Bazzi, C. LaPre, J. Makhoul, C. Rapahel, R. Schwartz, "Omnifont and Unlimited Vocabulary OCR for English and Arabic," *Proc. Int. Conf. Doc. Analysis and Recognition,* Ulm, Germany, pp. 842-845, Aug. 1997

[8] L. Nguyen, T. Anastasakos, F. Kubala, C. LaPre, J. Makhoul, R. Schwartz, N. Yuan, G. Zavaliagkos, and Y. Zhao, "The 1994 BBN/BYBLOS Speech Recognition System," *Proc. ARPA Spoken Language Systems Technology Workshop,* Austin, TX, Morgan Kaufmann Publishers, pp. 77-81, January 1995

[9] I.T. Phillips, S. Chen, and R.M. Haralick, "CD-ROM document database standard," *Proc. Int. Conf. Document Analysis and Recognition*, Tsukuba City, Japan, pp. 478-483, Oct. 1993

QUARC: A Remarkably Effective Method for Increasing the OCR Accuracy of Degraded Typewritten Documents

Michael Cannon, Judith Hochberg, and Patrick Kelly
Los Alamos National Laboratory

Abstract. *We present a practical method for improving the OCR accuracy of degraded typewritten document images. Our method is based on a judicious selection of a restoration algorithm for each document that is to be processed. The selection is based on a comprehensive assessment of the quality of the document. The assessment quantifies the severity of a variety of document degradations, such as background speckle, touching characters, and broken characters. A statistical classifier then uses these measures to select an optimal restoration method for the document at hand. On a 41-document corpus, our methodology improved the corpus OCR character accuracy by 24% and the word accuracy by 30%.*

1. Introduction

Commercial OCR algorithms perform well on clean laser-written documents. However, many organizations have huge archives of typewritten material, much of it of marginal quality. For example, the U.S. Department of Energy has an archive of over 300 million classified documents consisting of typewritten documents, teletypewriter output, and carbon copies on aging fibrous paper. As part of the declassification review process, almost all of these documents have been photocopied and/or photoreduced. By today's OCR standards, this archive and others like it are of marginal quality. Even though many successful document enhancement methods are known [1 - 3], they must often be applied under human guidance to avoid further image degradation. Unsupervised use of enhancement software can lead to a marked degradation in corpus OCR accuracy[4].

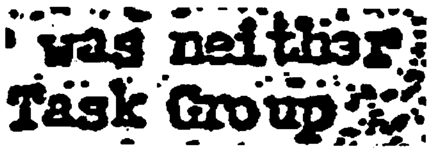
In this paper we present an effective method for *automatically* selecting the optimal restoration method for each document in a corpus. The method consists of two parts. First, we use five measures to assess the quality of a document image. Second, we use this quality assessment to automatically select an optimal restoration algorithm for each document by means of a statistical classifier. After restoration, we show a marked improvement in corpus OCR accuracy. On a 41-member document corpus, our methodology resulted in a 24% improvement in OCR character accuracy and a 30% improvement in word accuracy. We call our procedure **QUARC**: **QU**ality **A**ssessment, **R**estoration, and o**C**r.

2. Data

The Department of Energy made a 41-member 300-dpi corpus of document images available to us for this work. Its quality is representative of the archive mentioned above. Ground-truth text files for the documents were also made available. We used Caere OmniPage Pro v8.0 to perform OCR and found the character accuracy of the corpus to be 65.72% and the word accuracy to be 49.01%.

3. Quality Measures

Our document image quality measures are designed to quantify the document degradations we observed in the DOE corpus. Many of these degradations are illustrated in Figure 1.



was neither
Task Group

Figure 1. A portion of a page from the 41-member DOE corpus. This is a photocopy of a low-contrast carbon copy, which originally had neither background speckle nor broken characters. The photocopier automatically set a threshold to map subtle changes in gray tone to black or white.

We formulated five quality measures, each normalized to the range 0 to 1. The following is a preliminary description of the measures. A more technical definition will be given in the oral presentation of the paper and is also found in [5] and [6].

1. *Small Speckle Factor (SSF).* The small speckle factor measures the amount of black background speckle in the document image. The origin of the speckle varies. In our DOE corpus, much of it arises from photocopying low contrast documents (Figure 1). The background speckle can sometimes be so severe that it is interpreted as text by the OCR engine.

2. *White Speckle Factor (WSF).* Many degraded documents exhibit fattened character strokes. This problem can arise in carbon copies of documents, especially photocopies of carbon copies. The fattened stroke width can lead to OCR difficulties by creating unexpected small white connected components or by reducing or eliminating expected white components.
3. *Touching Character Factor (TCF).* The touching character factor measures the degree to which neighboring characters touch. Like white speckle, touching characters are caused by fattened strokes, as seen in the word "was" in Figure 1. Touching characters cause problems for OCR by making it difficult to differentiate between certain letters such as "ni" and "m", and by creating completely novel and uninterpretable text.
4. *Broken Character Factor (BCF).* The broken character factor measures the degree to which individual characters are broken. In our 41-document corpus, broken characters are the largest single cause of OCR errors. Broken characters often arise from photocopying low contrast documents, as seen in both occurrences of the letter "e" in Figure 1.
5. *Font Size Factor (FSF).* We find a correlation in our corpus between OCR accuracy and the size of the font. This correlation might not stem from the font size *per se*, but rather from degradations that accompany an increase or decrease in the size of the font.

As we developed the five quality measures, some of the parameters within each measure were tweaked in order to make the correlation with the OCR error as high as possible[5]. The correlation between quality measures and the OCR error was sufficiently high to motivate us to attempt to predict the OCR error rate based on the quality measures themselves. The prediction was based on a linear combination of the quality measures and was computed using a least-square method[6]. We obtained the weights for the linear combination by training on half the data; we then used the weights to predict the error rates for the other half. The correlation between the actual OCR error rates and the predicted ones was .89, an indication that the quality measures are indeed meaningful.

4. Restoration Methods

Our document image restoration methods are designed to repair the degradations reflected in the quality measures. We implemented fourteen restoration algorithms, but determined that only four were effective[5]. We applied each restoration method to the documents in our corpus, OCR'd all the resulting document images, and then computed the corresponding OCR accuracies. The restored version of a document with the highest OCR accuracy indicated the restoration algorithm that was best suited for that particular document.

- *Do Nothing.* It may be that the best enhancement for a document image is to leave it alone. Doing nothing is therefore included in our suite of restoration algorithms.
- *Cut on Typewriter Grid.* The documents in our corpus lie on a fixed-width typewriter (or teletypewriter) grid. If a document is plagued with touching characters, we should in principle be able to separate them if the typewriter grid is known. Our method for determining the typewriter grid is an extension of a method put forth by Lu [7]. We find the typewriter grid by first computing the Fourier transform of the vertical projection of lines of text. The average of the magnitude-squared of the transforms is computed. A typical average is shown in Figure 2. The prominent peak indicates the period of the typewriter grid.

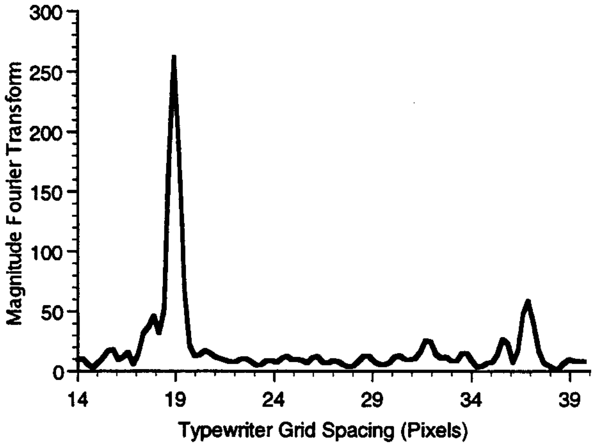


Figure 2. An average of the Fourier transform magnitudes of several lines of text. The prominent peak at 19 pixels indicates the width of the typewriter grid.

Our cutting algorithm moves along each line of text extracting two neighboring characters at a time. The location of the characters is known from the typewriter grid. We check to see if the two characters constitute the same black connected component - if they do, a white vertical line is drawn between them. If the characters do not touch, the line is not drawn, as it may destroy character detail such as serifs. In order to stay synchronized with the true character positions, the algorithm frequently computes the cross correlation between the typewriter grid and the vertical projection of the line of text and adjusts its position to the point of maximum correlation. In the Appendix we describe the special case of a *variable-width* typewritten font.

- *Global Fill Holes and Breaks.* In order to fill in breaks and fractures in characters, we employ a method described by Loce and Dougherty [8]. The filling operation consists of operating on the document image with 8 simple morphological kernels and ORing the results together.
- *Global Despeckle.* In order to suppress black background speckle while preserving character shape, we rely on another method described by Loce and Dougherty [9]. They prescribe a union of a 2-erosion basis set. Each kernel is 3x3 with two nubbins on it, which we apply globally to the document image.

5. Automatic Restoration Method Selection

We are now in a position to train the statistical classifier that will predict the best restoration method for new documents. We know each document's five quality measures that will be input to the classifier. We also know the best restoration method (out of the four-method set) that will optimally improve it. More generically, we have 41 objects, each described by five features and belonging to one of four classes, a classic pattern classification problem. We therefore trained a statistical classifier, using the Pocket algorithm [10], to assign each document, based on its five quality measures, to one of the four restoration methods. We did this two times, training first to the best category for improving OCR *character* accuracy, then to the best category for improving *word* accuracy.

We tested the statistical classifier using cross-validation. That is, we cycled through the entire corpus, on each iteration training on 40 documents and testing on the 41st. The OCR improvement resulting from the best possible restoration method for each document gave an upper bound for these results. For a lower bound, we found the outcome of choosing a restoration method randomly.

The following subsection presents the results of the cross-validation test in three ways: according to improvement in OCR character accuracy, improvement in OCR word accuracy, and selection of the optimal OCR algorithm. By all measures, the method was a success.

5.1 OCR Character Accuracy Results

As shown in Table 1, automatic selection of a restoration method substantially improved the OCR character accuracy of the corpus. The character accuracy in the 41-document subcorpus increased from 65.72% to 81.18%, a hefty 24% improvement. The improvement was not quite as good as our established upper bound (the outcome using the best restoration method for each document), but certainly better than our lower bound (from random selection of a restoration method).

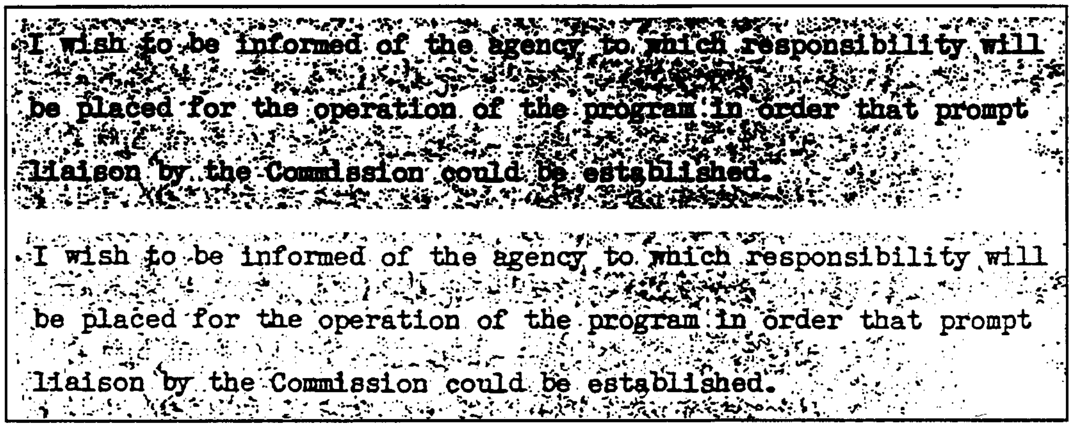
Restoration Selection Procedure	Character Accuracy	Word Accuracy
No restoration	**65.72%**	**49.01%**
Random selection of four restoration methods	**72.52%**	**53.76%**
Statistical classifier selection of four methods	**81.18%**	**63.80%**
Best of four restoration methods	**82.51%**	**65.62%**

Table 1. A compilation of OCR character accuracies resulting from a variety of restoration method selection criteria.

5.2 *Restoration Cascade*

It is tempting to couple our restoration methods in pairs. Perhaps a best restoration method would consist of background despeckle followed by a cut on the typewriter grid. We have experimented with some of these combinations and obtained spotty results. Some restoration cascades improved four or five of our documents by an additional 10% or so character accuracy. But in general, we saw little improvement in the corpus OCR accuracy as a whole.

One reason for this lackluster result may be that some of our restoration methods tend to be dual purpose already. For example, the Loce/Dougherty despeckle algorithm also tends to thin fattened strokes, as shown in Figure 3. The algorithm by itself has the effect of a cascade. Another reason the cascade



I wish to be informed of the agency to which responsibility will be placed for the operation of the program in order that prompt liaison by the Commission could be established.

I wish to be informed of the agency to which responsibility will be placed for the operation of the program in order that prompt liaison by the Commission could be established.

Figure 3. Top: a portion of an original document plagued by background speckle and fattened stroke widths. Bottom: the same portion of the document after enhancement by the Loce/Dougherty 2-erosion basis set. Note that both degradations have been addressed by the one enhancement method.

does not work is that the restoration methods also introduce artifacts into the document image. Perhaps the application of two methods in cascade introduces too many artifacts for the OCR engine to handle, and the benefit of the restoration methods is lost.

We believe that a cascade of restoration methods may still have merit; it is just too difficult to show it and train a classifier accordingly on our 139-member corpus. We will investigate the approach further when a larger 1000-member document corpus becomes available to us.

6. Software Implementation

Our entire approach to document image restoration and OCR has been implemented in three C++ software modules.

1. **TRAIN**: The user runs **TRAIN** in a directory containing many document images and their ground truth text files. **TRAIN** restores each image using four different methods and then computes the OCR accuracy resulting from each one. The quality measures from each document image as well as its best restoration method are written to a disk file. It takes several hours for **TRAIN** to run.

2. **CLASSIFIER**: The user next runs a program called **CLASSIFIER**, which reads in the disk file created by **TRAIN**. **CLASSIFIER** uses this information to create the classifier that is used by **QUARC** to automatically select an optimal restoration method based on a document's quality measures. It takes just a few seconds for **CLASSIFIER** to run; information defining the classifier is written to a disk file.

3. **QUARC**: The main production program is **QUARC**, which reads in the disk file produced by **CLASSIFIER** and then proceeds to optimally restore and **OCR** the document images that are passed to it.

7. Conclusions

We have presented a successful method for automatically improving the quality of document images in a typewritten archive, and we demonstrated a marked increase in OCR accuracy. The 24% improvement in OCR character accuracy and the 30% improvement in word accuracy on our 41-member corpus are significant. The method is easy to use - we view it as a pre-OCR cleanup operation, and it takes about one-tenth the computational effort of the OCR process itself. We like the automatic classifier because it takes into account all five quality measures when selecting an appropriate restoration method, rather than using one or two thresholds set by trial and error on a subset of the measures. Our methodology is not limited to our suite of four restoration methods. Any other restoration methods can be included, even if they are folded into the OCR process itself, as long as they are "best" for a meaningful number of documents in a training corpus.

On the other hand, the need to train a classifier for best performance on a particular corpus is a real effort, because it requires textual ground truth. Perhaps one-time training on a very large corpus would obviate the need for repeated training on smaller specialized corpora.

Acknowledgments

Professor Nathan Brener and his staff at Louisiana State University scanned the 41 documents used in this study and provided the keyed-text ground truth. Don Hush at Los Alamos provided assistance with the statistical classifier and the Pocket algorithm. Tom Curtis, Department of Energy, and Jim Campbell and Gary Craig, Federal Intelligent Document Understanding Laboratory, provided important administrative and technical support. Technical conversations with Becker

Drane and Steve Dennis, Department of Defense, were particularly valuable to us.

References

1. Victor T. Tom and Paul W. Baim, *Enhancement for Imaged Document Processing*, Proceedings 1995 Symposium on Document Image Understanding Technology, Annapolis, MD, p154.
2. P. Stubberud, et. al, *Adaptive Image Restoration of Text Images that Contain Touching or Broken Characters*, Proceedings ICDAR'95 Third International Conference on Document Analysis and Recognition, Montreal, 1995, p778.
3. TMSSequoia, "ScanFix Software," 206 West 6th Avenue, Stillwater, OK, 74074 ©1997.
4. In a joint experiment with Highland Technologies on a 140-document corpus, we found the OCR character accuracy dropped from 80% to 66% after applying ScanFix corrective techniques in an unsupervised manner.
5. T. M. Cannon, J. G. Hochberg, P. M. Kelly, *Quality Assessment and Restoration of Typewritten Document Images*, submitted to International Journal on Document Analysis and Recognition, expected publication date: Spring, 1999.
6. Michael Cannon et. al, *An Automated System for Numerically Rating Document Image Quality*, Proceedings 1997 Symposium on Document Image Understanding Technology, Annapolis, MD, p162.
7. Yi Lu, *On the Segmentation of Touching Characters*, Proceedings, ICDAR'93 Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, p440.
8. Robert P. Loce and Edward R. Dougherty, *Enhancement and Restoration of Digital Documents*, SPIE Optical Engineering Press, 1997, p192.
9. ibid., p198.
10. S. I. Gallant, *Preceptron-based Learning Algorithms*, IEEE Trans. Neural Networks, Vol. 1, No. 2, 1990, p179.

Appendix

In Section 4, we describe a method for cutting touching characters on a typewritten grid. Since not all typewritten fonts are fixed-width, it is important to determine if we are dealing with a variable-width font before cutting on a non-existent fixed-width grid. We can determine if a document has a fixed-width font by measuring the height of the peak shown in Figure 2. If the height is more than ten standard deviations above the mean of the transform, the document has a fixed-width font, otherwise the font is variable-width and no attempt is made to separate touching characters.

Information Extraction

Handwritten Document Image Analysis at Los Alamos: Script, Language, and Writer Identification

Judith Hochberg, Kevin Bowers, Michael Cannon, and Patrick Kelly

Mail Stop B265, Los Alamos National Laboratory, Los Alamos, NM 87545
{judithh, tmc, kelly}@lanl.gov
kbowers@eecs.berkeley.edu

Abstract

A system for automatically identifying the script used in a handwritten document image is described. The system was developed using a 496-document dataset representing six scripts, eight languages, and 281 writers. Documents were characterized by the mean, standard deviation, and skew of five connected component features. A linear discriminant analysis was used to classify new documents, and tested using writer-sensitive cross-validation. Classification accuracy averaged 88% across the six scripts. The same method, applied within the Roman subcorpus, discriminated English and German documents with 85% accuracy. Pilot results indicate that a variation of the method may be applicable to writer identification.

1. Introduction

Script and language identification are important parts of the automatic processing of document images in an international environment. A document's script (e.g., Cyrillic or Roman) must be known in order to choose an appropriate optical character recognition (OCR) algorithm. For scripts used by more than one language, knowing the language of a document prior to OCR is also helpful. And language identification is crucial for further processing steps such as routing, indexing, or translation.

For scripts such as Greek, which are used by only one language, script identification accomplishes language identification. For scripts such as Roman, which are used by many languages, it is normally assumed that script identification will take place first, followed by language identification within the script (e.g. [1]). Alternatively, it may be possible to skip script identification as an intermediate step, recognizing languages directly regardless of their script.

To the best of our knowledge, script identification has never been attempted for handwritten documents. Because of the dramatic individual differences in handwriting, we found a feature-based approach to be most successful, in contrast to the template matching we have previously applied to machine printed documents [2-3]. In the spirit of Wilensky et al. [4], each document was characterized by a single feature vector, containing summary statistics taken across the document's black connected components. The documents were then classified using linear discriminant analysis.

The main focus of this work was script identification: the method was 88% accurate in distinguishing among six scripts, including challenging pairs of related (and visually similar) scripts such as Roman/Cyrillic and Chinese/Japanese. We also took a first look at language identification within the Roman script: the method was 85% accurate for English versus German documents. Finally, we report promising pilot results (80% accuracy for a rough

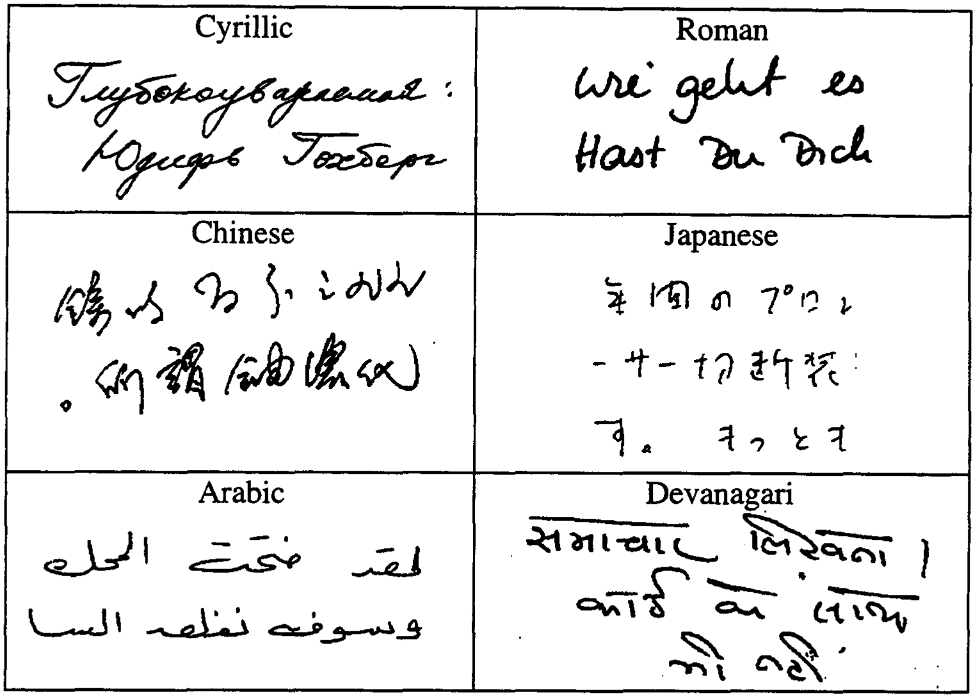


Fig 1. Examples of six handwritten scripts

implementation) on a variation of our method applied to writer identification from free text.

2. Data

We assembled a corpus of 496 handwritten documents from six scripts: Arabic, Chinese, Cyrillic, Devanagari, Japanese, and Roman. The scripts are illustrated in Figure 1. For the most part, document images were obtained from foreign language speakers we were acquainted with or whom we contacted through the Internet. Over 75% of the documents we collected were 'natural' -- letters, lecture notes, official documents, etc. The remaining documents were written on request. 281 different writers were represented in the corpus.

Around a third of the documents had at least one document quality issue such as ruling lines, line curvature, line skew, character fragmentation, or brevity (fewer than 100 connected components). Character fragmentation and ruling lines were addressed in preprocessing. We did not attempt to correct for the other phenomena, but simply included all documents in the training and testing process in order to perform a realistic test of the classification method.

3. Script identification

Connected components. The basic element of the analysis was the eight-connected black component. After finding all the components in a document image, unusually small or large components were filtered out in order to remove speckle, ruling lines, and outsize components in general. Some filtering criteria were absolute (e.g., removing components with height or width less than three pixels), and some were relative (e.g., removing components with height or width more than four standard deviations above the document mean).

Features. Once filtering was completed, several features were extracted from the remaining components. To develop the feature set we studied the document images and determined which visual features guided our human script identification. The final set of features was:

- relative Y centroid

- relative X centroid
- number of white holes
- sphericity
- aspect ratio (height/width)

For each of the five connected component features, three document summary statistics were calculated: the mean, standard deviation, and skew. This created a fifteen-element vector for each document.

Discrimination. The classification method used a collection of linear discriminant functions. A separate Fisher linear discriminant [5] was trained to separate each possible pair of scripts in the dataset (Arabic vs. Chinese, Arabic vs. Cyrillic, etc.). New documents were classified by applying each individual linear discriminant to the document's feature vector, while keeping track of the results. The document was then assigned to the class receiving the most "votes".

The classifier was tested through writer-sensitive cross-validation. For each writer, the classifier was trained on all data except that writer's documents. Then the writer's documents were classified using the trained classifier. We calculated the percentage of documents correctly classified for each script, and averaged these percentages to produce an overall accuracy figure unbiased by the scripts' sample sizes.

Results. The linear discriminant analysis was 88% accurate. Table 1 breaks down these results by script, and also presents the cross-classification matrix. The individual percentages for the different scripts were pleasingly uniform, especially since the amount and quality of data available for the different scripts varied considerably. When documents were misclassified, the errors were sensible: Roman and Cyrillic tended to be confused, and likewise Chinese and Japanese.

Character fragmentation adversely affected classification: 90% of documents with no fragmentation, or only mild fragmentation, were correctly classified, compared to 81% of documents with moderate or severe fragmentation ($F = 5.21$, $p < 0.05$). Ruling lines also appeared to affect classification -- 89% of unruled documents were correctly classified, compared to 81% of ruled documents -- although this difference was just short of statistical significance ($F = 3.18$, $p = 0.07$). Of the 366 documents in the corpus with no or mild fragmentation, and without ruling lines, 91% were classified correctly.

4. Language identification

Method. Of the Roman script documents in the corpus, 107 were in English and 58 in German. Using the same preprocessing, feature selection, and classification techniques described for script identification, we attempted to distinguish between these two groups. We also tried to identify

Table 1. Script identification results

Script	% correct	Classified as					
		Arabic	Chinese	Cyrillic	Devanagari	Japanese	Roman
Arabic	89%	51	0	0	3	2	1
Chinese	87%	0	104	0	0	8	8
Cyrillic	88%	1	0	49	2	0	4
Devanagari	88%	0	0	1	22	1	1
Japanese	86%	3	6	0	0	63	1
Roman	91%	2	1	9	0	3	150
Average	**88%**						

the languages directly, using a single discriminant analysis for the seven-way discrimination among Arabic, Chinese, Cyrillic, Devanagari, Japanese, German, and English.

Results. For the two-way (English vs. German) task, correct identification averaged 85% (84% for English, 86% for German). For the seven-way task, 80% of English and German documents were correctly identified by language. Interestingly, overall *Roman* identification improved to 93% when the two languages were split apart. It may be that the heterogeneity of the combined Roman group adversely affected the script classifier's performance, so that dividing the group into two smaller, more homogeneous groups helped. Classification of the other scripts was not affected by the Roman split.

5. Writer identification

While experimenting with a variation of the method described in sections 3 and 4, we obtained exciting pilot results for writer identification. We present them here, knowing that they are extremely preliminary, in the hopes that they may inspire further research.

Method. Connected components were identified, filtered, and features extracted as described above, producing a five-element vector per component. Then a *k*-means cluster analysis was performed across the entire training set, resulting in 256 clusters, or connected component types. Now each document could be represented as a histogram of cluster occurrences, and documents compared to each other on the basis of their histogram similarity, using a distance metric such as Kullback-Leibler entropy [6].

The pilot study analyzed the 282 documents in our corpus whose writers had contributed more than one document to the corpus. Using the histogram comparison method, we determined how many of these documents were most similar to another document by the same writer -- a one-nearest-neighbor classifier.

Related work by Wilensky et al. on Chinese writer identification from free text also used a nearest-neighbor algorithm, representing each document by a feature vector similar to the ones we used for script and language identification [4].

Results. As shown in Table 2, roughly 80% of documents by multi-document writers were closest to another document by the same writer. This was particularly impressive given that (at the time) the corpus contained 466 documents, representing 260 different writers. In other words, the odds of picking another document by the same writer by chance were extremely low.

In comparison, Wilensky et al.'s accuracy was much higher (98% or better), but their task was much easier since it involved only fifteen writers and 106 documents. In addition, their feature selection (in contrast to ours) was optimized for writer identification.

Table 2. Pilot results for writer identification

Script	# documents by multi-document writers	# correctly matched	% correctly matched
Arabic	14	12	86%
Chinese	99	79	80%
Cyrillic	37	29	78%
Devanagari	19	16	84%
Japanese	3	3	100%
Roman	110	87	79%
total	282	226	80%
total # writers			260

5. Conclusion

The feature set and classifier we developed served to discriminate scripts with 88% accuracy. While not as accurate as script identification for machine printed document images [2-3], this result exceeded our initial expectations given the variability of handwritten documents. Classification accuracy was higher for documents without fragmented characters and ruling lines.

Language identification for English versus German was 85% accurate once Roman identity was known, and 80% accurate when script and language identification were performed together. It would be worthwhile to explore the language identification task more thoroughly.

Pilot work showed 80% accuracy for writer identification. We believe that these results could be improved with a more judicious selection of features. Since the pilot study was a casual offshoot of our main line of research, the features used to represent documents were chosen for their utility in script identification, not writer identification. In fact, we were careful not to choose features that showed substantial individual variation, such as white component sphericity. A more fully-developed algorithm along these lines could be useful in the intelligence field or in forensics.

References

[1] Spitz, A. L (1977). Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19:235-245.

[2] Hochberg, J., Kelly, P., Thomas, T., Kerns, L. (1997). Automatic script identification from document images using cluster-based templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19176-181.

[3] Patent: *Script identification from images using cluster-based templates* (5,844,991).

[4] Wilensky, G., Crawford, T., Riley, R. (1997). Recognition and characterization of handwritten words. In Doermann, D. (ed.): *Proceedings of the 1997 Symposium on Document Image Understanding Technology*. College Park, MD: University of Maryland Institute for Advanced Computer Studies, pp. 87-98

[5] Duda, T., Hart, P. (1973). *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, pp. 114-118.

[6] Deco, G. & D. Obradovic (1996). *An Information-Theoretic Approach to Neural Computing*. New York: Springer, p. 10.

Acknowledgments

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36. We would like to thank Steve Dennis, Ron Riley, and Greg Wilensky for their help. Most of all, we would like to thank the many people from around the world who shared with us their writing and that of their friends and families.

Classification of Document Page Images

Christian K. Shin **David S. Doermann**
Language and Media Processing Laboratory
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275
{cshin,doermann}@cfar.umd.edu

Abstract

Searching for documents by their type or genre is a natural way to enhance the effectiveness of document image retrieval. The layout of a document contains a significant amount of information that can be used to classify a document in the absence of domain-specific models. Our classification approach is based on a "visual similarity" between the structure of a document image and a representative document image of a defined class, and is realized by building a supervised classifier. We use image features, such as the percentages of text and non-text (graphics, image, table, and ruling) zones, the presence of bold font style, font size, and density of content area. In order to obtain class labels for training samples, we conducted a user relevance test where subjects rated UW-I document images with respect to 12 representative images. We implemented our classification scheme using OC1, a decision tree classifier, and report our initial findings.

1 Introduction

Searching in a large heterogeneous collection of scanned document images often produces uncertain results in part because of the size of the collection and the lack of an ability to focus queries appropriately. We need more natural ways of organizing and searching a collection of document images based on perceived *relevance*[1]. Searching for documents by their type or genre (i.e., functional category) is a natural way to enhance the effectiveness of document retrieval in the workplace, and such systems are proposed in [1, 2].

The goal of our work is to build *classifiers* that can label the type or genre of a document image. Many different genres of documents have a predefined form or a standard set of components that depict a unique spatial arrangement while other classes are less well defined. Layout analysis extracts structure without reference to *models*[2] of particular kinds of pages, and is necessary since our input image has no structural definition that is immediately perceivable by a computer. Classification is thus based on "visual similarity" of structures.

The general premise for searching documents based on their layout structure is that the layout structure of a document often reflects its type. For example, business letters are in many ways more visually similar to one another than they are to magazine articles. Thus, a user searching for a particular document while knowing the class of documents is able to more effectively narrow the group of documents being searched. Authors typically use combinations of layout and content visual features (e.g., bold font for emphasis) to convey an intended organization, or to assign priorities to specific components, so more detailed queries may make use of functional attributes [3, 4].

1.1 Related Work

Some page classification work has been reported in the literature, but most methods require either domain-specific models [5–8] or are based on text obtained by optical character recognition (OCR) [5–7].

In Dengel et al.[5], OfficeMAID processes business letters and assumes business letters as the only input. The domain document model is restricted to business letters, and the classification of recipient is based on OCR and used for delivery of the document via electronic mail.

Taylor et al.[8] have reported on a component of the IDUS (Intelligent Document Understanding

[1] *Relevance* in document image classification is defined as a measure of relatedness between a document image and a representative document of a defined class. It is a function of their similarity (proportional to relevance) or dissimilarity/distance (inversely proportional to relevance). The terms *relevance* and *similarity* are used interchangeably in this paper.

[2] *Models* provide document type specific interpretation for a block of text. If a specific model is given (for example, business letters, memos, forms, advertisements, or technical articles), a block of text can have a semantic label with respect to the model (such as the logical date or salutation in a letter).

System) where classification uses a combination of geometric-based and content-based features to process heterogeneous batches of documents including business letters, memos, technical journals, and newspapers. The approach employs two steps. The first step sorts documents by the number of columns, and the column structure determines one or more candidate models. The second step consists of classification engines for each (predefined) class. Each document classification engine is based on the presence of key functional components or landmarks, and requires OCR. The rules to compute these functional components in the domain-specific models are constructed manually.

In Hao et al.[6], documents are represented as trees obtained as a result of nested layout segmentation. Sample documents of pre-determined types (letters, memos, and articles) are also prepared as trees for classification. The classification is realized by matching input document instances to the sample trees using approximate tree matching [9]. The matching requires logical labels that are generated manually, and also requires OCR.

In Maderlechner et al.[7], classification is both by form and by content. The software system classifies a large variety of office documents according to layout form and textual content. For form classification, more than 200 layout models are acquired manually as reference models, and these models are compared with a given input document for pattern matching. For content classification, fuzzy string matching is performed on OCR'd text.

1.2 Approach

We propose a method for using layout structures of documents (i.e., their visual appearance) to facilitate the search and retrieval of a document stored in a multi-genre database by building a supervised classifier. Ideally, we need tools to automatically generate layout features that are relevant for the specific classification task at hand. Class labels for training samples can be obtained manually or by clustering examples. Once the image features and their types are obtained from a set of training images, classifiers can be built.

Our approach uses 59 image features derived from the existing University of Washington Image Database I (UW-I) groundtruth [10] including the percentages of text and non-text (graphics, image, table, and ruling) zones, the presence of bold font style, font size, and density of content area measured by dividing the total content area by the page area. To obtain class labels for training samples, we conducted a user relevance test where subjects rated UW-I document images with respect to a set of representative images. We used the relevance rating obtained from the experiment to assign class labels with varying degrees of confidence (Section 2). We implemented our classification scheme using the OC1 decision tree classification software [11] (Section 4).

2 Relevance Judgments

Relevance (or similarity) judgments are often subjective, and it is extremely difficult (if not impossible) to devise a single metric that can be consistently used to determine relevance between two images. Perceived relevance by humans is not only different among individuals, but it also varies according to one's perspective of the specific task at hand. Such approaches are common in testing environments [12]. Since humans are the ultimate judges, it is natural to attempt to measure perceptual relevance based on the characteristics of human relevance judgments. There have been many attempts to study different aspects of the human perception of relevance in psychology [13] as well as to understand how the perception of relevance is used in office environments in social anthropology [1].

In order to obtain relevance judgments for image classification experiments, we conducted a user relevance test using the UW-I Document Image Database, a collection of 979 pages of technical article images. These page images are primarily from multi-page articles, but are treated as individual, independent images.

We first selected 12 representative pages of visually different classes from the database (Figure 1). We prepared a survey form that showed the 12 representative thumbnail images, and a separate package of all UW-I thumbnail images. For each representative image, we asked each of seven subjects to browse through the UW-I thumbnails, and to find images that are visually similar. For each similar image, they were asked to rate it with a degree of relevance ranging from 1 (minimally similar) to 5 (highly similar).

The test consists of 12 representative query images against 979 UW-I thumbnail images on 10 pages (each page containing ~100 thumbnails). Each subject spent two to four hours to complete the test; this averages to one to two minutes per query for browsing and selectively marking each thumbnail page. From each of our subjects, we thus received 12 sets of relevance judgment. The data obtained from the experiment are summarized in Figure 2. The two key factors are number of judgments (Figure 2a) and relevance rating (Figure 2b).

Due to the large number of images in the survey database, we expected that only a subset of the images would be labeled as relevant (having relevance rating ≥ 1) for each representative image. We fo-

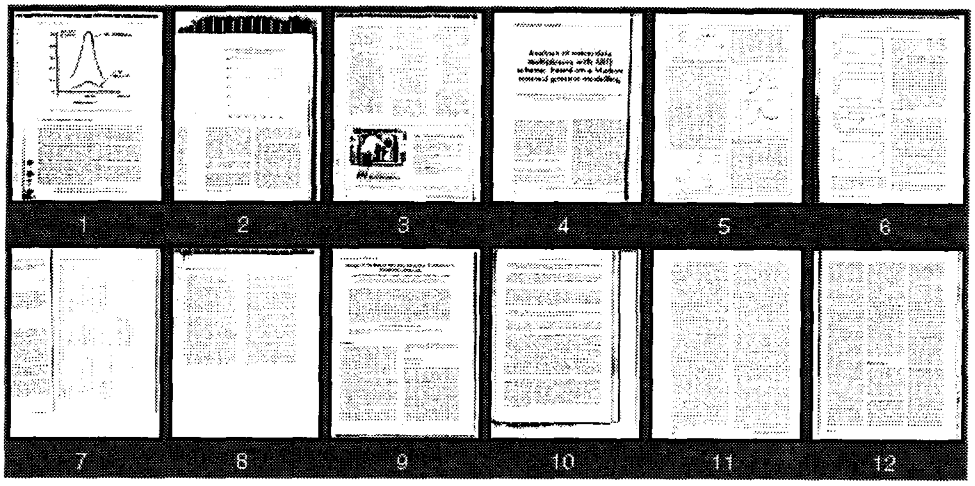


Figure 1: 12 selected representative classes of images for the similarity experiment.

Table 1: Cumulative number of images found relevant to representative image #1.

Rel.	Number of Judgments						
Rating	≥ 1	≥ 2	≥ 3	≥ 4	≥ 5	≥ 6	7
≥ 1	197	118	83	45	25	12	5
≥ 2	173	114	80	44	25	12	5
≥ 3	111	72	55	37	22	11	5
≥ 4	34	25	19	16	11	5	3
5	1	1	1	1	1	1	1

cused on the images that were labeled by at least one subject. The results were processed as follows.

First, images that are identified as relevant to some representative image(s) were grouped together based on how many of the seven subjects agreed (columns in Table 1). The first group (column) includes database images which at least one of the seven subjects labeled as relevant to at least one representative image, and the last group (column) includes images that all seven subjects agreed were relevant to some representative image(s) (representative image #1 in Table 1). The fact that more subjects labeled images as relevant seems to suggest that there is more agreement in their relevance to a representative image.

The images in each of the seven groups are further divided into five groups based on the average relevance ratings from 1 to 5 (rows in Table 1). The average relevance ratings are computed by dividing the sum of the relevance ratings by the number of subjects who labeled the image with relevance ratings ≥ 1. Each group is considered to be more relevant to one of the representative images than the preceding group. The fact that images have higher average relevance ratings seems to suggest that these images are more similar to a representative image. The numbers of images in each of the two sets of groups for the representative image #1 are shown in Table 1, and they are also graphically shown in Figure 3. In Figure 3a, each stacked bar represents a column (number of judgments) of data in Table 1, and each stacked bar in Figure 3b represents a row (relevance rating) of data in Table 1. For this example, 197 judgments were made.

Our classification approach requires that each test and/or training image have a unique class label (one of 12 representative image classes). One way of doing this for each document is to simply pick the class with the highest average relevance across all judgments. As previously mentioned, however, both the relevance assigned by the subjects and the number of judgments made are important features in determining the actual relevance of a given document to a class. Intuitively, documents with a larger number of judgments, and a higher average judgment, are more relevant. In order to compute a single value for each document/class pair, we explored several ways of combining the individual relevance ratings and the number of judgments.

Ultimately, we found that multiplying the number of judgments by the avenge relevance rating was the best solution. For each image, we compute the total relevance scores by multiplying the average relevance rating for a given class, by the number of judgments made for that class. We then select the class having the maximum total relevance score as an assigned label for that image.

3 Implementation

In order to facilitate our classification task based on layout structures, we generated layout features that capture aspects of visual appearance. We used 59 features derived from UW-I groundtruth [10]. They are 30 content features: content ratios of graphics, image, table, ruling, text, and combined non-text for the entire page and each of four quadrants; page density (dividing sum of all content zone areas by page area); content occupancy (dividing total content bounding box area by page area); maximum column numbers; 5 zone features: number of all, text, image, and graphics zones, and average zone area; 5 total content bounding box features: area, density, aspect ratio, and coordinates of centroid; and 16 font features: presence of bold font and large font size on 8 horizontal page slices.

We have trained classifiers that determine class memberships among the 12 classes of layout structure. The relevance judgments obtained in Section 2 are used for determining class memberships of training and test examples. For each representative image, candidate image pair we compute a total score by multiplying the number of judgments made by the average relevance rating for that image. We then ordered the test and training images by their *total scores.*[3]

For building and testing classifiers, we used the decision tree classifier software package, OC1 [11]. We provided 59 image features (discussed above) together with class labels (described in Section 2) as input to OC1. For determining quality of decision trees, we used OC1's default *impurity measure*[4], the Twoing Criterion [14].

4 Experiments

We evaluated the performance of the classifiers on various training sets based on the strength of the training set as determined by the user relevance test. The UW-I images in the user relevance test are sorted (in descending order) by their total relevance scores, and are divided into 7 subsets, each consisting of about 135 images. The first subset contains the training images that have the highest 1/7th of the total scores, and the 7th subset contains the images that have the lowest 1/7th of the total scores.

Figure 4a shows classification accuracy for each 7th-tile of the test set using the "leave-one-out" resampling evaluation method. The left-most point shows around 83% classification accuracy for the top 7th-tile of the test set, and the rightmost point is the classification accuracy for the bottom 7th-tile. This confirms the fact that the classifier performance increases as quality of the training set increases. Figure 4b shows overall classification accuracy for each of the 12 document types for the first 7th-tile.

We also evaluated the performance of the classifiers on title pages. We first grouped the training and test images in the user relevance test (described in Section 2) into two classes, title page and non-title page. We combined page classes #4 and #9 in Figure 1 into a title page class, and the rest were combined into a non-title page class. We then assigned class labels to training sample images accordingly, and obtained the classification accuracy using the "leave-one-out" resampling evaluation method. The accuracy results for each 7th-tile are presented in Figure 5a. There are 57 title pages and 891 non-title pages in the experiment. We obtained an overall classification accuracy of 95.57% (906 correct out of 948). The classifier correctly classified 37 out of 57 title pages, and we had 22 false positives and 20 false negatives; examples are shown in Figure 5b.

4.1 Discussion

We investigated the page classes that obtained the four lowest classification accuracies (i.e., page classes 6, 4, 5, and 9 shown in Figure 4b). The user relevance test shows that page class pairs 5 & 6 and 4 & 9 are perceived as similar to each other, hence have reduced accuracy. The images belonging to page classes 4 & 9 and 5 & 6 have one of the highest co-identified ratios. 91% of the page images in page classes 4 & 9 and 48% of the page images in page classes 5 & 6 are identified together by at least one subject, and are ranked first and third among 66 possible class pairs, respectively. Many subjects, for example, identified images in page class 4 as being in page class 9, or vice versa. We intuitively can see that page classes 4 & 9 both represent two-column title pages, and page classes 5 & 6 are similar in that they both are two-column pages with some graphics.

From the observation above, we combined the top four most similar page class pairs. The top four co-identified page class pairs are 4 & 9, 1 & 2, 5 & 6, and 10 & 11, and their co-identification ratios are 91%, 52%, 48%, and 40% respectively. The classification accuracy for the first 7th-tile improved to ~91% (8-page classification) from ~83% in 12-page classification. Figure 6a shows the classification accuracy for each 7th-tile of the test set using the "leave-one-out" resampling evaluation method. This again confirms the fact that classifier performance increases as quality of training set increases.

5 Conclusions and Future Work

In this work, we have proposed and implemented a supervised classification module for building classifiers that are capable of classifying user-defined types based on layout features in the absence of

[3] Here, the *total score* is used as a measure of strength of each training image. Having a higher total score indicates that the image is more relevant to one of the representative query images than one with a lower total score.

[4] The *impurity measure* or metric is used to determine the "goodness" of a decision boundary location.

domain-specific models. The classification accuracy we obtained is very promising. We are currently identifying more relevant features, and building an automatic feature extraction module. As discussed in Section 4, classification accuracy is dependent on the quality of training samples. We have plans to develop an effective methodology or mechanism to build efficient document image classifiers by better understanding similarity/distance relationships among the training examples. We also have plans to develop indexing methods for effective genre-based document image search and retrieval for large heterogeneous document image collections.

Acknowledgments

The support of this effort by the Department of Defense under contract MDA 9049-6C-1250 is gratefully acknowledged.

References

[1] J. Blomberg, L. Suchman, and R. Trigg. Reflections on a work-oriented design project. *PCD '94: Proceedings of the Participatory Design Conference*, pages 99–109, 1994.

[2] D. Doermann, C. Shin, A. Rosenfeld, H. Kauniskangas, J. Sauvola, and M. Pietikainen. The development of a general framework for intelligent document image retrieval. In *International Workshop on Document Analysis Systems*, pages 605–632, 1996.

[3] D. Doermann, E. Rivlin, and A. Rosenfeld. The function of documents. *International Journal of Computer Vision*, 16:799–814, 1998.

[4] M. Lipshutz and S.L. Taylor. Functional decomposition of business letters. In *Symposium on Document Analysis and Information Retrieval*, pages 435–448, 1995.

[5] A. Dengel, R. Bleisinger, F. Fein, R. Hoch, F. Hones, and M. Malburg. OfficeMAID - a system for office mail analysis, interpretation and delivery. In *International Workshop on Document Analysis Systems*, pages 253 – 276, 1994.

[6] X. Hao, J.T.L. Wang, M.P. Bieber, and P.A. Ng. Heuristic classification of office documents. *International Journal on Artificial Intelligence Tools*, 7:233–265, 1995.

[7] G. Maderlechner, P. Suda, and T. Bruckner. Classification of documents by form and content. *Pattern Recognition Letters*, 18:1225–1231, 1997.

[8] S.L. Taylor, M. Lipshutz, and R.W. Nilson. Classification and functional decomposition of business documents. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 563–566, 1995.

[9] J.T.L. Wang, K. Zhang, K. Jeong, and D. Shasha. A system for approximate tree matching. *IEEE Transactions on Knowledge and Data Engineering*, 16:559–571, 1994.

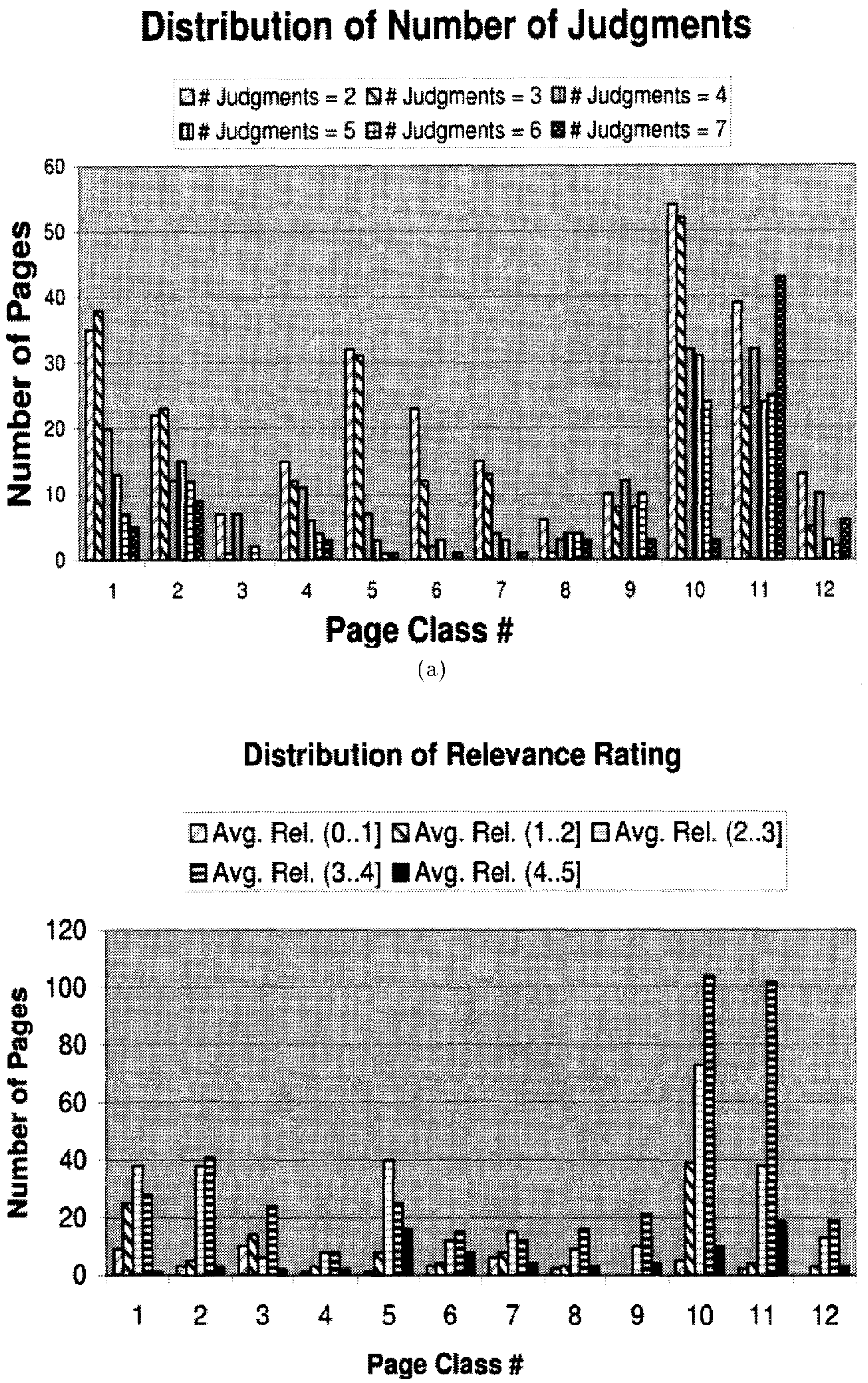
[10] University of Washington. University of Washington Document Images CD-I.

[11] S. Murty, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[12] E. M. Voorhees and D. K. Harman (Eds.). *The Seventh Text REtrieval Conference (TREC-7).* Department of Commerce, National Institute of Standards and Technology, 1998.

[13] S. Santini and R. Jain. Similarity matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Submitted).*

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.



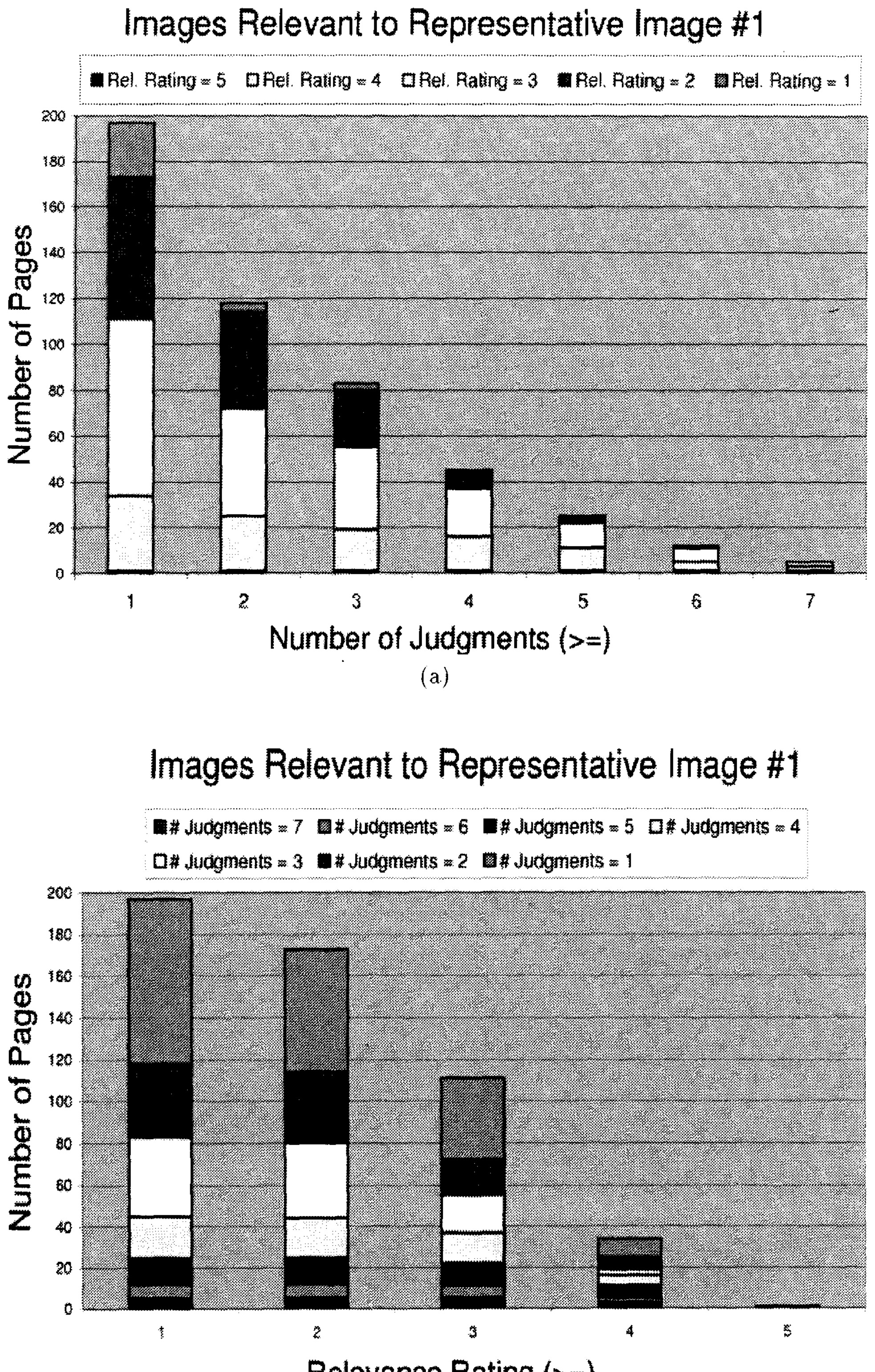
Distribution of Number of Judgments

(a)

Distribution of Relevance Rating

(b)

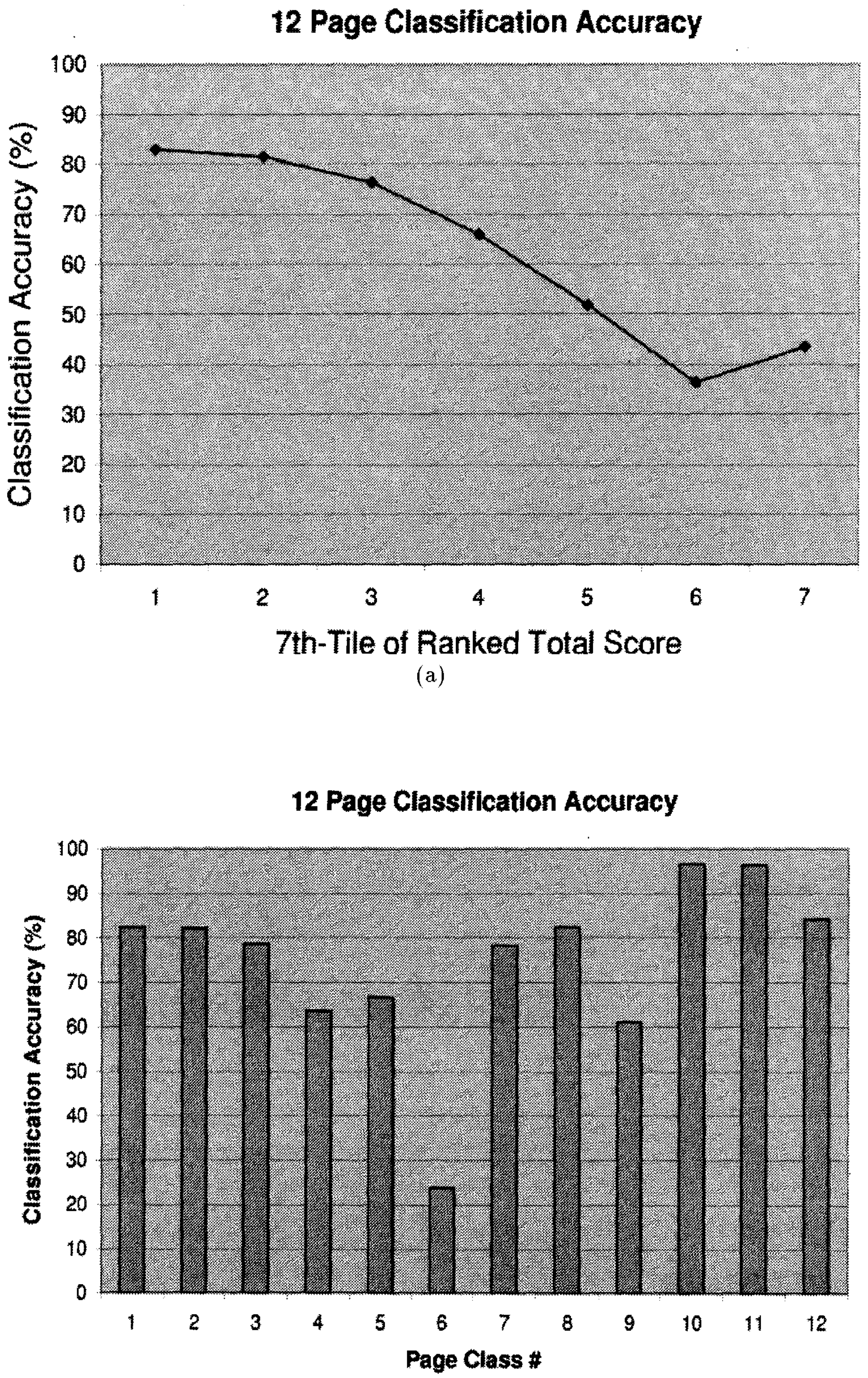
Figure 2: (a) Distribution of number of judgments, and (b) distribution of relevance rating.



(a)

(b)

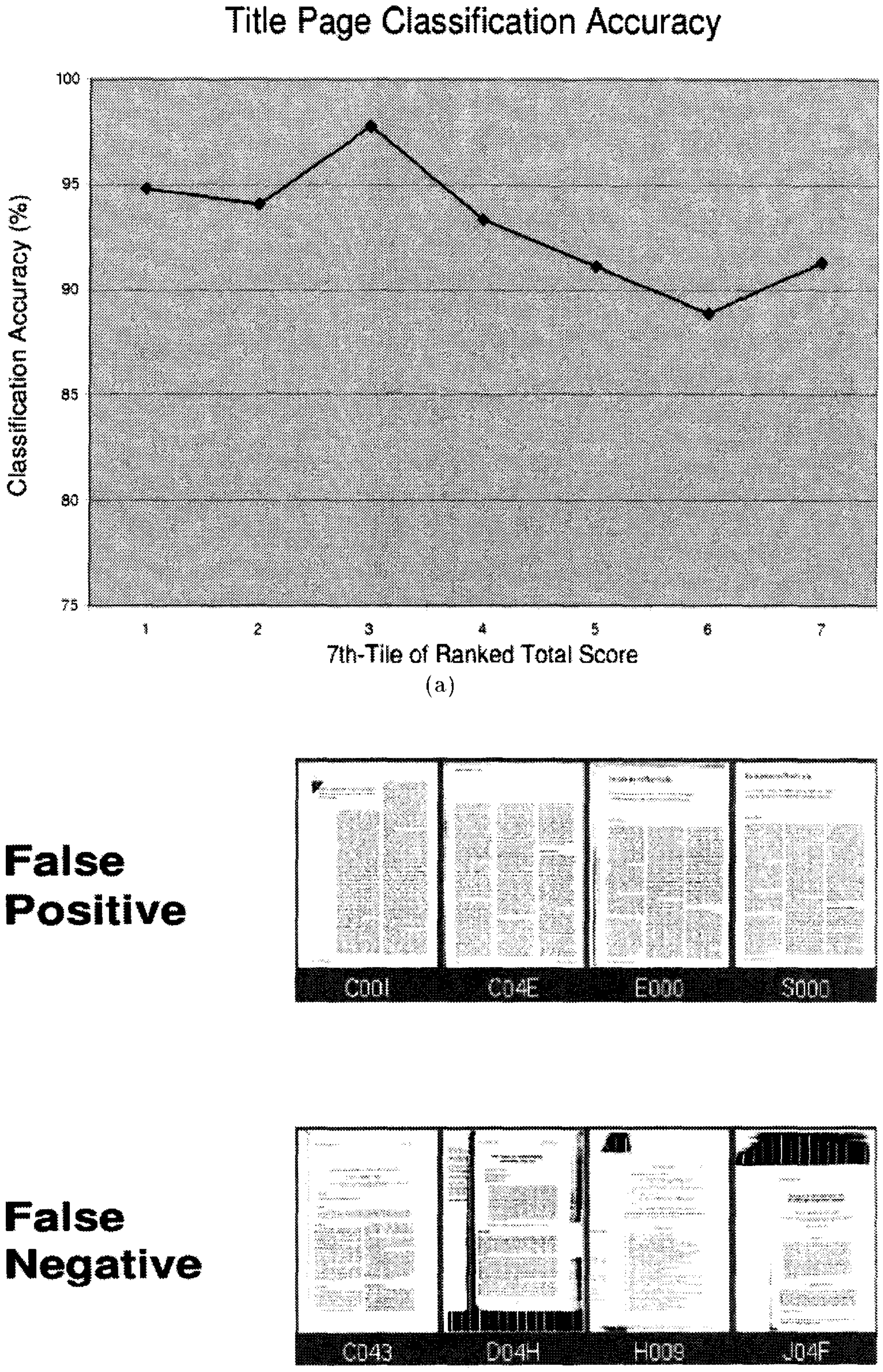
Figure 3: (a) Number of images relevant to representative image #1 with respect to number of judgments, and (b) number of images relevant to representative image #1 with respect to relevance rating.



(a)

(b)

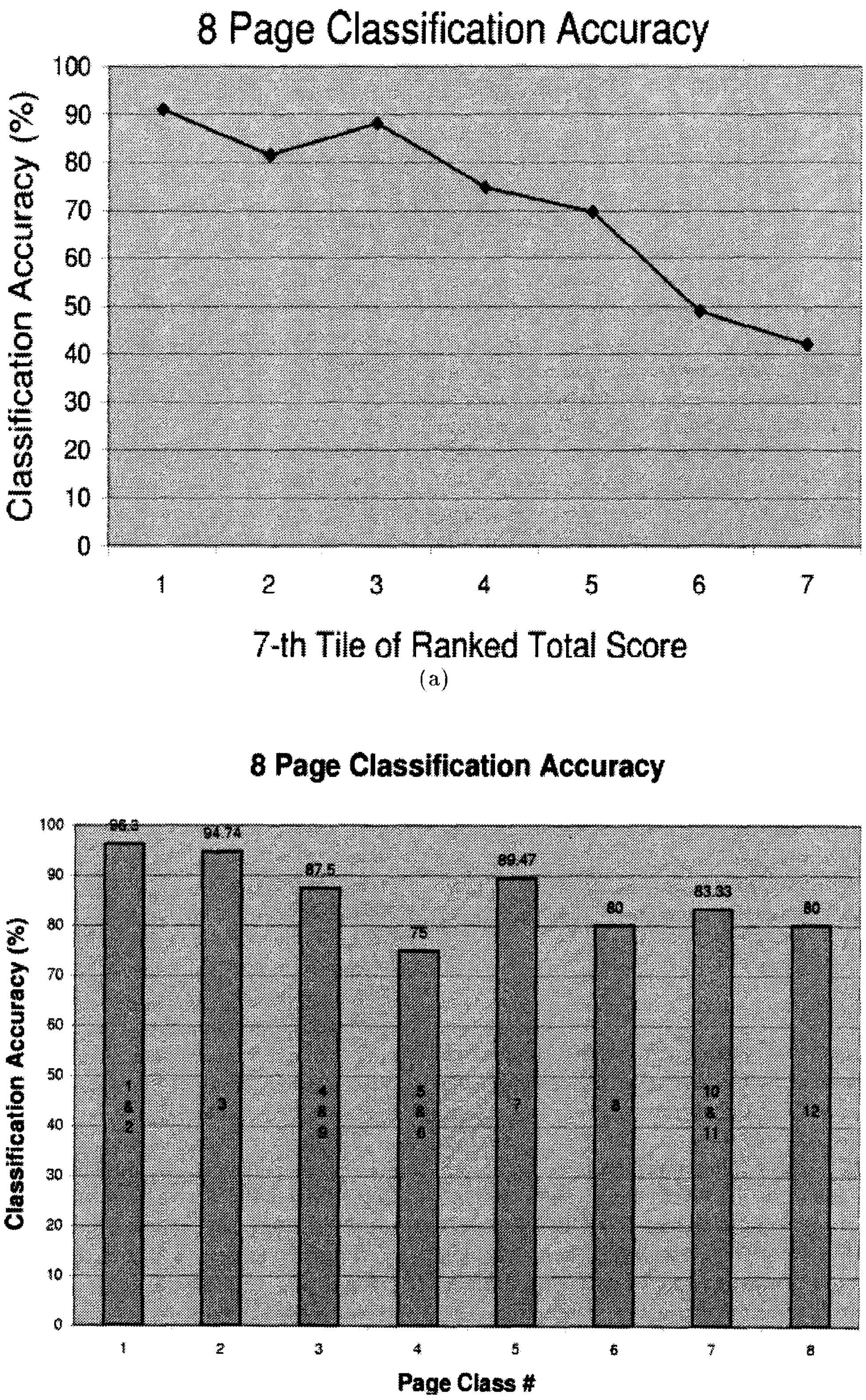
Figure 4: (a) Page classification accuracy for ordered 7th-tiles, and (b) page classification accuracy for each document types for the first 7th-tile.



(a)

(b)

Figure 5: (a) Title page classification accuracy for ordered 7th-tiles, and (b) title page classification errors.



(a)

(b)

Figure 6: (a) 8 Page classification accuracy for ordered 7th-tiles, and (b) 8 page classification accuracy for each document types for the first 7th-tile.

Information Extraction from Symbolically Compressed Document Images

Dar-Shyang Lee, Jonathan J. Hull

Ricoh Silicon Valley, Inc.
2882 Sand Hill Road, Suite 115
Menlo Park, CA 94025
email: {dsl,hull}@rsv.ricoh.com

Abstract

The extraction of information from symbolically compressed document images is an increasingly important problem as the related standard (JBIG2) and commercial products become available. Symbolic compression techniques work by clustering individual connected connected components (blobs) in a document image and storing the sequence of occurrence of blobs and representative blob templates, hence the name symbolic compression. These techniques are specifically targeted to improving the compression ratio in binary document images. This paper proposes methods for extracting information from symbolically compressed document images by deciphering the sequence of occurrence of blobs. We propose a new deciphering algorithm that uses a hidden Markov model. Applications of this method to language identification, multilingual OCR, and duplicate detection are discussed. Experiments in duplicate detection are performed using the MG software package and the University of Washington database. The OCR-free and language independent nature of the algorithm suggests possible applications in a multilingual document database.

1. Introduction

The extraction of information from compressed document images is useful since the compression algorithm not only reduces the size of the image, providing less data to process, but also represents characteristics of the original image in the compressed data stream that can be used directly to compute information about original document. CCITT groups 3 and 4 compression are one example. These methods include pass codes in the compressed data stream, which are attached to bottoms of strokes and concavities. The configuration of pass codes in CCITT-compressed document images has been used for skew detection [26] and duplicate detection [12, 17].

Symbolic compression has recently been proposed for inclusion in the JBIG2 standard [10]. Symbolic compression methods were first discussed by Ascher and Nagy [1]. More recent works include [9, 17, 27, 29]. In symbolic compression, images are coded with respect to a library of pattern templates. Templates in the library are typically derived by grouping (clustering) together connected components that have similar shapes. One template is chosen or generated to represent each cluster. The connected components in the image are then stored as a sequence of template identifiers and their offsets from the previous component. In this way, an approximation of the original document is obtained without duplicating storage for similarly shaped connected components. Minor differences between individual components and their representative templates, as well as all other components which are not encoded in this manner, are optionally coded as residuals.

An example of symbolic compression is shown in Figure 1. After connected component clustering, the original document image is represented as a set of bitmap templates, (A a h i s t) in this example, their sequence of occurrence in the original image (0 1 2 1 5 3 4 1 2 1 5 3 4 1 5 1 5), as well as information about the relative geometric offset between adjacent connected components (e.g., (+2, 0) means the beginning of the second component in the sequence is 2 pixels to the right of the end of the first component), and a compressed residual image. The residual is the difference between the original image and the pattern templates. This data can be compressed with arithmetic coding or another technique. A lossy representation for a symbolically compressed image could be obtained by not storing the residual image.

Symbolic compression techniques improve compression efficiency by 50% to 100% in comparison to the commonly used Group 4 standard [18, 28]. A lossy version can achieve 4 to 10 times better compression efficiency than Group 4 [28]. Symbolic compression techniques are also used in some multilayered compression formats for color documents [8].

The symbolic compression format is especially useful for information extraction. Clusters of connected components that are approximately the size of characters can be assumed to be characters. Also, the sequence of cluster identifiers is a substitution cipher. This allows us to apply a deciphering algorithm to extract character interpretations. The use of a

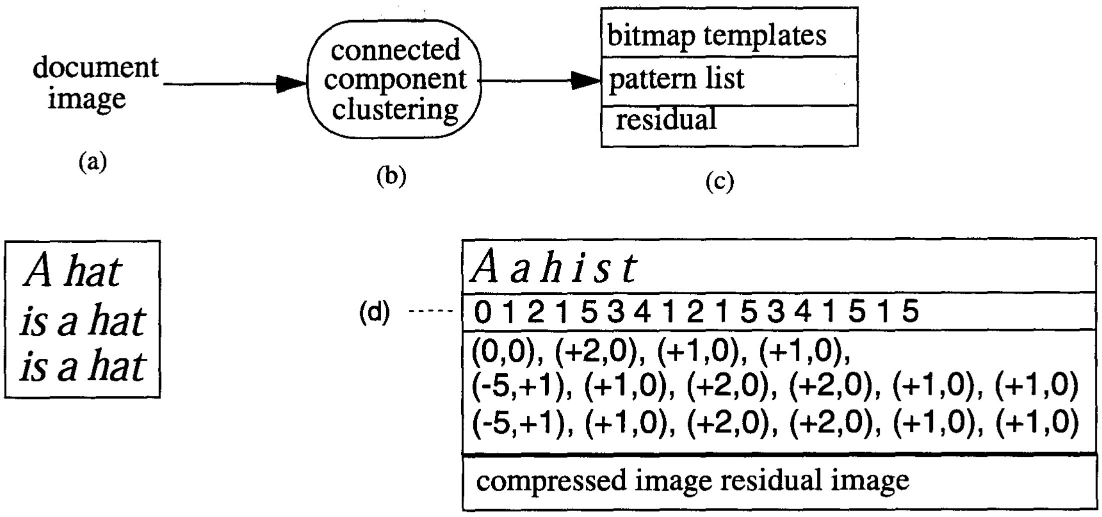


Figure 1 - Example of symbolic compression. The connected components in an original document image (a) are grouped into clusters (b). A bitmap template is chosen to represent each cluster and stored in the compressed file (c) together with their sequence of occurrence in the original image (d). Information about the geometric offset between adjacent components as well as image residual data are also stored in the compressed format.

deciphering algorithm for character recognition was proposed by Nagy and Casey [3]. Both character-level [2, 19] and word level [6] deciphering techniques have been proposed for text recognition.

This paper describes the application of a novel deciphering algorithm to the extraction of information from symbolically compressed document images. The deciphering algorithm can be configured to perform language identification and OCR in multiple languages. The accuracy of the OCR results may not be as high as that of a commercial OCR process. However, the accuracy is often high enough to be useful for various applications. The use of such character recognition results for document duplicate detection is described in this paper. An n-gram method for document matching is proposed. Experimental results demonstrate the utility of the character recognition technique and the accuracy of the document matching method.

2. Character Interpretation Deciphering

In the example shown in Figure 1, with the exception of the capital "A", there is a one-to-one correspondence between bitmap templates and English alphabetic characters. This is an ideal case known as a *simple substitution cipher.* If we were to replace each template identifier by its corresponding alphabetic character, "a" for 1, "h" for 2, and so on, the original message could be recovered from the sequence of component identifiers.

In practice, however, multiple templates can be formed for a single alphabetic symbol, as in the case of upper and lower case "a". This results in a many-to-one *homophonic substitution cipher.* In an even more realistic scenario, a single pattern could correspond to a partial symbol or multiple symbols due to image fragmentation and segmentation errors.

A deciphering algorithm is available for simple substitution ciphers. By exploiting the redundancy in a language, the plain text message can be recovered from a sequence of cipher symbols of sufficient length [22]. Numerous algorithmic solutions have been proposed for simple substitution ciphers, including relaxation techniques [14, 16, 20] dictionary-based pattern matching [19, 24] and optimization techniques [7, 25]. We propose a deciphering algorithm that uses a Hidden Markov Model (HMM) [23].

Considering the Markov process of state traversal as a language source from which a particular plain text message can be generated with some probability, then the added symbol production at the traversed states in an HMM describes the enciphering process of a substitution cipher, where each letter in plain text is replaced with a cipher symbol one at a time. This analogy between the source language modeling as a Markov process and the representation of the enciphering function by symbol probabilities is the basis for our solution. The state probabilities are initialized with language statistics, and the symbol probabilities are estimated with the EM algorithm.

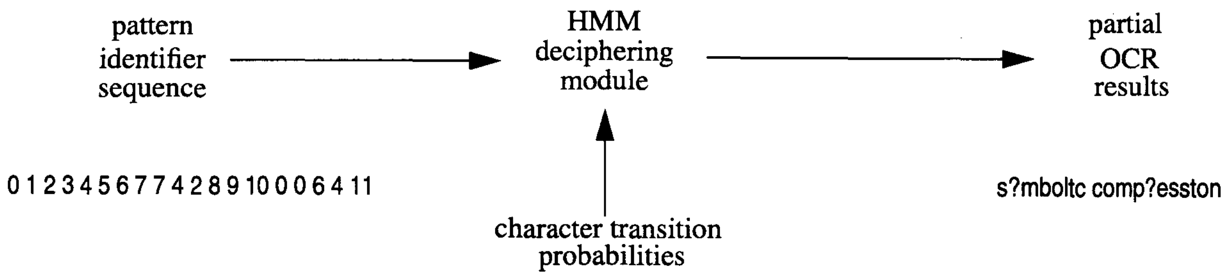


Figure 2 - Deciphering a symbolic compressed image produces partial OCR results.

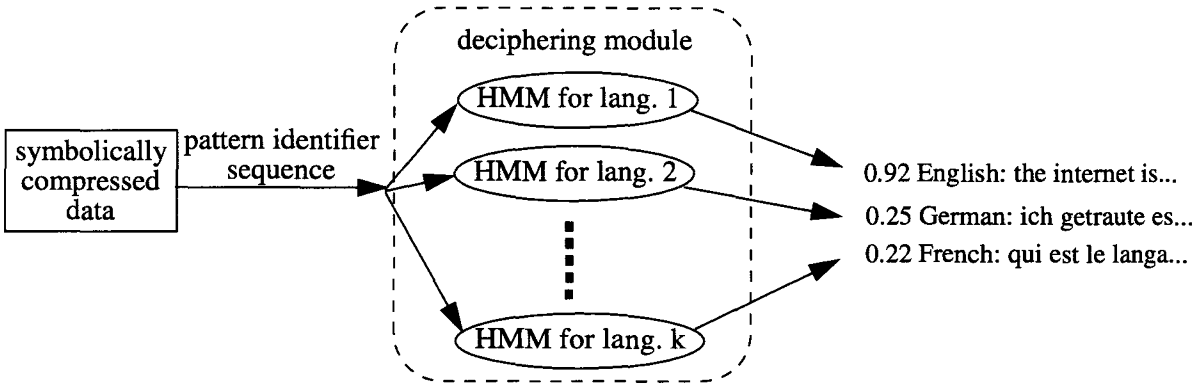

deciphering module
symbolically compressed data
pattern identifier sequence
HMM for lang. 1
HMM for lang. 2
HMM for lang. k
0.92 English: the internet is...
0.25 German: ich getraute es...
0.22 French: qui est le langa...

Figure 3 - Simultaneous language identification and deciphering. The deciphering module is given a compressed document of unknown language. It produces a set of possible interpretations in various languages.

Information extraction from symbolically compressed documents can be viewed as a deciphering problem. The objective is the recovery of the association between character interpretations and pattern templates from a sequence of template identifiers. In symbolic compression schemes, image components are grouped to improve clustering and they are also roughly sorted in reading order to the reduce entropy in their relative offsets. The objective of both measures is to improve compression performance. However, they also facilitate the application of deciphering techniques for information extraction.

Figure 2 shows an outline of the proposed HMM deciphering algorithm. It reads the pattern identifier sequence from a symbolically compressed document image and uses character transition probabilities to produce partial OCR results. These results may not be completely correct. However, they are often adequate for various tasks which will be described in the next section.

There are several reasons why the deciphered results will be less than perfect. First of all, it is obvious that the problem is never truly a simple substitution. The use of upper and lower case letters and multiple typefaces always lead to more than one template per alphabetic symbol. Imaging defects and segmentation problems further complicate the template-to-symbol mapping. In addition, short sequences and rare patterns do not posses sufficient statistics for deciphering. Even with ample exemplars, certain contents such as numeric strings can not be deciphered due to lack of context. Nevertheless, we believe sufficient information can be recovered for language identification, duplicate detection or document classification.

Identification of the language of the text in the original image can also be performed by using a collection of HMM's. The pattern sequence extracted from a compressed document is simultaneously deciphered with various language models. Each result includes the partial OCR results as well as a score that measures the confidence of the model. The language can be identified by selecting the model that produced the maximum score. This process is depicted in Figure 3.

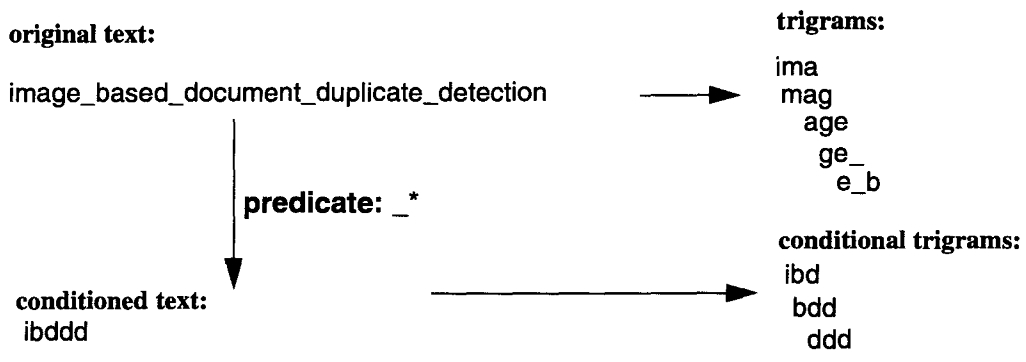


Figure 4 - Conditional n-grams are generated from consecutive characters satisifying a predicate.

3. Information Extraction from Partial OCR Results

We use an n-gram method to extract information from the partial OCR results output by the deciphering algorithm. N-gram based methods have been used for various information extraction tasks [11, 13]. Their error-tolerant and language-independent characteristics are particularly suitable for information extraction from partially deciphered character interpretations. Cavnar and Trenkle [4] and Damashek [5] showed that an n-gram based document categorization algorithm is resistant to garbled input text.

An n-gram method for measuring the similarity of two documents typically extracts all sequences of n consecutive characters from each document. Their similarity is represented by a function of the number of n-grams they have in common.

While regular n-grams provide a robust solution to information retrieval, they do not present an effective indexing scheme for applications like document matching. The redundancy results in a large number of indexing terms and the converging behavior of n-gram statistics blurs distinctions between individual document. Densely clustered documents of similar contents decrease the error tolerance of the indexing method for finding any particular document.

We use a modified n-gram method to detect whether two documents are duplicates of each other. We use *conditional* n-grams that are generated from consecutive characters that satisfy a predicate. For example, a predicate of "the character following the space character" would form n-grams from the first character of consecutive words, as illustrated in Figure 4.

Since conditional n-grams are formed on a string of filtered text, they generate fewer terms per document than the more typically used non-conditional n-grams described above. It attempts to eliminate some of the redundancies in regular n-grams to obtain effective indexing terms and a more uniformly distributed document space using appropriately defined conditions.

The similarity of two documents is measured by calculating the dot product of their n-gram frequency vectors. Each entry in such a vector is the number of occurrences of an -gram in the document. Documents that have dot products above a threshold are duplicates.

4. Experimental Results

The experimental performance of the HMM deciphering algorithm and the conditional n-gram method for text string comparison were investigated. The HMM deciphering algorithm was trained with character transition probabilities calculated from a corpus of over 100,000 words of English.

The character deciphering rate (number of characters deciphered in a test document) as a function of the amount of text was first investigated for a perfect simple substitution cipher problem. The results in Table 1 show that a 99% deciphering rate is achieved with only 1200 characters of test data, using character bigram statistics. Similar performance is achieved with 800 characters of test data and trigram statistics. This illustrates the value of the additional contextual information present in trigrams.

The deciphering algorithm was also tested on sequences of cluster identifiers extracted from a few synthetic images and three all-text images in the University of Washington database [21]. The mgtic algorithm [28] was used as the symbolic compression algorithm. Between 80% to 95% of the characters in the testing documents were correctly deciphered.

The performance of the conditional n-gram method for document matching was tested on the 979 documents in the University of Washington (UW) database. This database contains 146 pairs of duplicate documents. Each member of a pair had been scanned from a different generation photocopy of the same document. Approximately 10% of the characters in the ground truth files for the UW database were corrupted to simulate a 90% correct decode rate by the HMM.

Conditional trigrams, as well as conventional trigrams and 5-grams were extracted from each of the 979 UW documents. Each document was compared to the other 978 documents by calculating a similarity score using a weighted sum of the frequencies of the n-grams they have in common. A sorted list of the 10 documents with the highest similarity scores was output. The most similar document is at top of the list. Ideally, this is a duplicate for the original document, if it exists in the database.

Table 2 compares the performance of conditional and non-conditional n-grams in duplicate detection. The Top 1 correct rate is the percentage of the 292 test documents with the highest similarity scores that are duplicates. This shows how often the correct match is the first choice output by the comparison algorithm. The Top 10 correct rate is the percentage of documents with duplicates for which the duplicate was contained in the 10 documents with the highest similarity scores. The storage space for this technique is indicated by the total number of n-grams indexed.

The results in Table 2 show that conditional trigrams provide a 100% correct rate in duplicate detection. This compares to the 81.85% correct rate achieved by non-conditional trigrams, in the first choice, and 97.95% in the top 10 choices. Non-conditional 5-grams also produced a 100% correct duplicate detection rate. However, this was at the cost of almost a 40:1 increase in storage requirement in comparison to conditional trigrams.

Table 1. Character deciphering rates for various lengths of text in perfect simple substitution ciphers.

# of chars	100	200	400	800	1200	1600	2000
bigram	57.55	72.73	93.19	96.74	99.13	99.13	99.56
trigram	66.47	90.17	98.80	99.01	99.44	99.54	99.76

Table 2. Comparison of duplicate detection rates and storage required for various conditional and non-conditional n-grams.

criterion	**non-conditional trigrams**	**non-conditional 5-grams**	**conditional trigrams**
Top 1 correct rate	81.85%	100%	100%
Top 10 correct rate	97.95%	100%	100%
Total number of n-grams indexed	19,098	712,460	16,180

5. Conclusions

A method for information extraction from symbolically compressed document images was presented. The technique is based on a novel deciphering approach that uses Hidden Markov Models. Although the error rate in the text recovered by deciphering is normally higher than that by a conventional OCR system, we demonstrated that there is sufficient information for certain document processing tasks. An n-gram based method for duplicate detection was proposed here.

The deciphering algorithm is limited to documents composed mostly of text. Also, the success of deciphering depends on redundancy in the language and the original document. Therefore, it would be difficult to adapt it to ideographic languages such as Chinese or to apply it to very short documents.

Experimental results showed that the HMM can successfully decipher over 98% of the text in English language document images that contain as little as 400 characters. The proposed technique for duplicate detection was also investigated experimentally. Duplicates were successfully detected in a database of about 979 images.

Future work includes investigation of adaptation to new languages. Only gathering of statistics for the HMM should be required.

References

[1] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Transactions on Computers*, Vol. C-23, No. 11, pp. 1174-1179, Nov. 1974.

[2] L. R. Bähl and J. Cocke, "Font-independent character recognition by cryptanalysis," *IBM Technical Disclosure Bulletin*, Vol. 23, No. 3, pp. 1588-1589, August 1981.

[3] R. Casey and G. Nagy, "Autonomous reading machine," *IEEE Transactions on Computers*, vol. C-17, No. 5, May 1968.

[4] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," *Proceedings of the Symposium on Document Analysis and Information Retrieval*, pp. 161-175, Las Vegas, Nevada, 1994.

[5] M. Damashek, "Gauging similarity with n-grams: language-independent categorization of text", *Science*, pp. 843-848, February, 1995.

[6] C. Fang and J. J. Hull, "A word-level deciphering algorithm for degraded document recognition," *Fourth Symposium on Document Analysis and Information Retrieval*, University of Nevada at Las Vegas, Las Vegas, Nevada, April 24-26, 1995, pp. 191-202.

[7] W. S. Forsyth and R. Safavi-Naini, "Automated cryptanalysis of substitution ciphers," *Cryptologia*, vol. 17, no. 4, pp. 407-418, 1993.

[8] P. Haffner, L. Bottou, P. G. Howard, P. Simard, Y. Bengio and Y. Le Cun, "Browsing through high quality document images with DjVu," *Proceedings of IEEE Advances in Digital Libraries*, Santa Barbara, California, April, 1998.

[9] P. Howard, "Lossless and lossy compression of text images by soft pattern matching," *Proceedings of the IEEE Data Compression Conference (DCC'96)*, Snowbird, pp. 210-219, 1996.

[10] P. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The Emerging JBIG2 Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838-848, November 1998.

[11] S. Huffman, "Acquaintance: language-independent document categorization by n-grams," *Proceedings of the 4th Text REtrieval Conference*, 1996,

[12] J. J. Hull, "Document image similarity and equivalence detection," *International Journal on Document Analysis and Recognition*, vol. 1 no. 1, February, 1998, 37-42.

[13] J. J. Hull and S. N. Srihari, "Experiments in text recognition with binary n-gram and viterbi algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 5, pp. 520-530, September 1982.

[14] D. G. N. Hunter and A. R. McKenzie, "Experiments with relaxation algorithms for breaking simple substitution ciphers," *The Computer Journal*, vol. 26, no. 1, pp. 68-71, 1983.

[15] O. Kia and D. Doermann, "Symbolic Compression for document analysis," *Proceedings of International Conference on Pattern Recognition*, Volume III, 1996, pp. 664-668.

[16] J. King and D. Bahler, "An implementation of probabilistic relaxation in the cryptanalysis of simple substitution ciphers," *Cryptologia*, vol. 16, no. 3, pp. 215-225, 1992.

[17] D. S. Lee and J. J. Hull, "Group 4 Compressed Document Matching," *Proceedings the the Third IAPR Symposium on Document Analysis Systems*, Nagano, Japan, Nov. 4-6, 1998, pp. 29-38.

[18] K. Mohiuddin, J. Rissanen and R. Arps, "Lossless binary image compression based on pattern matching," *Proceedings of International Conference on Computers, Systems & Signal Processing*, December, 1984.

[19] G. Nagy, S. Seth and K. Einspahr, "Decoding sub-

stitution ciphers by means of word matching with application to OCR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 710-715, 1987.

[20] S. Peleg and A. Rosenfeld, "Breaking substitution ciphers using a relaxation algorithm," *Communications of the ACM*, vol.22, no.11, pp. 598-605, November 1979.

[21] I. T. Phillips, S. Chen, R. M. Haralick, "CD-ROM document database standard," *Proceedings of the 2nd ICDAR*, pp. 478-483, 1993.

[22] F. Pratt, *Secret and Urgent: the story of codes and ciphers*, Blue Ribbon Books, Garden City, New York, 1942.

[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[24] R. S. Ramesh, G. Athithan and K. Thiruvengadam, "An automated approach to solve simple substitution ciphers," *Cryptologia*, vol. 17, no. 2, pp. 202-218, 1993.

[25] R. Spillman, M. Janssen, B. Nelson and M. Kepner, "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers," *Cryptologia*, vol. 17, no. 1, pp. 31-44, 1993.

[26] A. Lawrence Spitz, "Skew determination in CCITT group 4 compressed document images," *Proceedings of the Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, March 16-18, 1992, pp. 11-25.

[27] I. Witten, T. Bell, H. Emberson, S. Inglis, and A. Moffat, "Textual Image Compression: two stage lossy/lossless encoding of textual images," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 878-888, June 1994.

[28] I. Witten, A. Moffat and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold, New York, 1994.

[29] Q. Zhang and J. Danskin, "Entropy-based pattern matching for document image compression," *Proceedings of the International Conference on Image Processing*, pp. 221-224, Lausanne, Switzerland, Sept. 1996.

Multimedia Information Retrieval at the Center for Intelligent Information Retrieval

R. Manmatha *
Multimedia Indexing and Retrieval Group
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts, Amherst, MA 01003
manmatha@cs.umass.edu

Abstract

Abstract: Building the digital libraries of the future will require a number of different component technologies including the ability to retrieve multi-media information. This paper will describe progress in this area at the Center for Intelligent Information Retrieval (CIIR). This includes:

1. *Multi-modal retrieval using appearance based image retrieval and text retrieval. This work has been applied to a large database of trademarks containing image and text data from the US Patent and Trademark Office. 68,000 trademarks may be searched using either image retrieval or image and text retrieval while 615,000 trademarks may be searched using text retrieval.*

2. *Indexing handwritten manuscripts. Recently we have developed a scale-space technique for word segmentation in handwritten manuscripts.*

3. *Other projects including color based image retrieval and the extraction of text from images.*

1 Introduction

The Center for Intelligent Information Retrieval (CIIR) has a number of projects to index and retrieve multimedia information. We will describe some of the progress made in these areas since the last SDIUT meeting [20]. The projects include:

1. Image Retrieval: Work on indexing images using their content continues using both appearance based and color based retrieval. In previous work on appearance based retrieval [20, 26] we focussed on part image retrieval - whether a part of a query image is similar to a part of a database image. Recent work at the Center has focussed on "whole image retrieval' ' i.e. whether two images are similar in their entirety. In this work [27], two images are considered similar if their distributions of local curvature and phase at multiple scales are similar.

   The Center is also doing work on retrieving images, using color, from homogeneous databases. [6]. Color retrieval ssytems are inappropriate for heterogeneous databases. For example, a query image of a red flower will retrieve not only red flowers but also other red objects like cars and dresses. Most users do not find this meaningful. If, however, the database consisted only of flowers then a query on the color red would only retrieve red flowers and this is more meaningful to most users. This work has been applied to indexing a small database of flower patents. Another salient feature of this work is that using domain constraints, we are able to segment flowers from the background and index only the color of the flower rather than all the colors of the entire image.

2. Multi-modal retrieval: A multi-modal retrieval combining appearance based image retrieval and text retrieval is being applied to retrieve trademark images from a database provided by the US Patent and Trademark Office. 68000 trademarks may be searched using either image retrieval or image and text retrieval while 615,000 trademarks may be searched using text retrieval. A multi-modal provides many constraints so that the image search may be constrained. In addition, our multi-modal system solves the problem of how a query image is to be obtained. An initial search is done using text and the result is a list of trademarks with their associated text and images which may then be used for image or text retrieval.

*This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademarks Office and the Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by NSF IRI-9619117 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

3. Finding Text in Images: The conversion of scanned documents into ASCII so that they can be indexed using INQUERY (CIIR's text retrieval engine). Current Optical Character Recognition Technology (OCR) can convert scanned text to ASCII but is limited to good clean machine printed fonts against clean backgrounds. Handwritten text, text printed against shaded or textured backgrounds and text embedded in images cannot be recognized well (if it can be recognized at all) with existing OCR technology. Many financial documents, for example, print text against shaded backgrounds to prevent copying.

   The Center has developed techniques to detect text in images. The detected text is then cleaned up and binarized and run through a commercial OCR. Such techniques can be applied to zoning text found against general backgrounds as well as for indexing and retrieving images using the associated text.

   The Center is continuing work in this area. Most of this work has involved speeding up some of our techniques in this area

4. Word Spotting: The indexing of hand-written and poorly printed documents using image matching techniques. Libraries hold vast collections of original handwritten manuscripts, many of which have never been published. Word Spotting can be used to create indices for such handwritten manuscript archives.

   Our recent work in this area has involved developing new techniques for word segmentation based on scale space methods. Old handwritten manuscripts are challenging for word segmentation algorithms for many reasons; ascenders and descenders from adjacent lines touch, noise and ink bleeding are present, the manuscripts show shine through and xeroxing and scanning have introduced additional artifacts. Our technique for word segmentation first involves segmenting the lines out using a new projection profile technique and then detecting words in each line by creating scale space blobs.

A discussion of the Center's work on word segmentation of handwritten mansucripts and its work on appearance based image retrieval and multi-modal retrieval now follows.

2 Appearance Based Image Retrieval and Multi-Modal Retrieval

The image intensity surface is robustly characterized using features obtained from responses to multi-scale Gaussian derivative filters. Koenderink [16] and others [11] have argued that the local structure of an image can be represented by the outputs of a set of Gaussian derivative filters applied to an image. That is, images are filtered with Gaussian derivatives at several scales and the resulting response vector locally describes the structure of the intensity surface. By computing features derived from the local response vector and accumulating them over the image, robust representations appropriate to querying images as a whole (global similarity) can be generated. One such representation uses histograms of features derived from the multi-scale Gaussian derivatives. Histograms form a global representation because they capture the distribution of local features (A histogram is one of the simplest ways of estimating a non parametric distribution). This global representation can be efficiently used for global similarity retrieval by appearance and retrieval is very fast.

The choice of features often determines how well the image retrieval system performs. Here the task is to robustly characterize the 3-dimensional intensity surface. A 3-dimensional surface is uniquely determined if the local curvatures everywhere are known. Thus, it is appropriate that one of the features be local curvature. The principal curvatures of the intensity surface are invariant to image plane rotations, monotonic intensity variations and further, their ratios are in principle insensitive to scale variations of the entire image. However, spatial orientation information is lost when constructing histograms of curvature (or ratios thereof) alone. Therefore we augment the local curvature with local phase, and the representation uses histograms of local curvature and phase.

Local principal curvatures and phase are computed at several scales from responses to multi-scale Gaussian derivative filters. Then histograms of the curvature ratios [15, 7] and phase are generated. Thus, the image is represented by a single vector (multi-scale histograms). During run-time the user presents an example image as a query and the query histograms are compared with the ones stored, and the images are then ranked and displayed in order to the user.

2.1 The choice of domain

There are two issues in building a content based image retrieval system. The first issue is technological, that is, the development of new techniques for searching images based on their content. The second issue is user or task related, in the sense of whether the system satisfies a user need. While a number of content based retrieval systems have been built ([10, 9]), it is unclear what the purpose of such systems is and whether people would actually search in the fashion described.

Here, we describe how the techniques described here may be scaled to retrieve images from a database of about 63000 trademark images provided by the US Patent and Trademark Office. This database consists of all (at the time the database was provided) the registered trademarks in the United States which consist only of designs (i.e. there are no words in them). Trademark images are a good domain with which to test image retrieval. First, there is an existing user need: trademark examiners do have to check for trademark conflicts based on visual appearance. That is, at some stage they are required to look at the images and check whether the trademark is similar to an existing one. Second, trademark images may consist of simple geometric designs, pictures of animals or even complicated designs. Thus, they provide a test-bed for image retrieval algorithms. Third, there is text associated with every trademark and the associated text maybe used in a number of ways. One of the problems with many image retrieval systems is that it is unclear where the example or query image will come from. In this paper, the associated text is used to provide an example or query image. In future papers, we will explore how text and image searches may be combined to build more sophisticated systems. Using trademark images does have some limitations. First, we are restricted to binary images (albeit large ones). As shown later in the paper, this does not create any problems for the algorithms described here. Second, in some cases the use of abstract images makes the task more difficult. Others have attempted to get around it by restricting the trademark images to geometric designs [13].

2.2 Global representation of appearance

Three steps are involved in order to computing global similarity. First, local derivatives are computed at several scales. Second, derivative responses are combined to generate local features, namely, the principal curvatures and phase and, their histograms are generated. Third, the 1D curvature and phase histograms generated at several scales are matched. These steps are described next.

A. Computing local derivatives: Computing derivatives using finite differences does not guarantee stability of derivatives. In order to compute derivatives stably, the image must be regularized, or smoothed or band-limited. A Gaussian filtered image $I_\sigma = I * G$ obtained by convolving the image I with a normalized Gaussian $G(\mathbf{r}, \sigma)$ is a band-limited function. Its high frequency components are eliminated and derivatives will be stable. In fact, it has been argued by Koenderink and van Doorn [16] and others [11] that the local structure of an image I at a given scale can be represented by filtering it with Gaussian derivative filters (in the sense of a Taylor expansion), and they term it the N-jet.

However, the shape of the smoothed intensity surface depends on the scale at which it is observed. For example, at a small scale the texture of an ape's coat will be visible. At a large enough scale, the ape's coat will appear homogeneous. A description at just one scale is likely to give rise to many accidental mis-matches. Thus it is desirable to provide a description of the image over a number of scales, that is, a scale space description of the image. It has been shown by several authors [18, 14, 32, 30, 11], that under certain general constraints, the Gaussian filter forms a unique choice for generating scale-space. Thus local spatial derivatives are computed at several scales.

B. Feature Histograms: The normal and tangential curvatures of a 3-D surface (X,Y,Intensity) are defined as [11]:

$$N(\mathbf{p}, \sigma) = \left[\frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2 I_x I_y I_{xy}}{\left(I_x^2 + I_y^2\right)^{\frac{3}{2}}} \right] (\mathbf{p}, \sigma)$$

$$T(\mathbf{p}, \sigma) = \left[\frac{\left(I_x^2 - I_y^2\right) I_{xy} + \left(I_{xx} - I_{yy}\right) I_x I_y}{\left(I_x^2 + I_y^2\right)^{\frac{3}{2}}} \right] (\mathbf{p}, \sigma)$$

Where $I_x(\mathbf{p}, \sigma)$ and $I_y(\mathbf{p}, \sigma)$ are the local derivatives of Image I around point p using Gaussian derivative at scale σ. Similarly $I_{xx}(\cdot, \cdot)$, $I_{xy}(\cdot, \cdot)$, and $I_{yy}(\cdot, \cdot)$ are the corresponding second derivatives. The normal curvature N and tangential curvature T are then combined [15] to generate a shape index as follows:

$$C(\mathbf{p}, \sigma) = atan \left[\frac{N + T}{N - T} \right] (\mathbf{p}, \sigma)$$

The index value C is $\frac{\pi}{2}$ when $N = T$ and is undefined when either N and T are both zero, and is, therefore, not computed. This is interesting because very flat portions of an image (or ones with constant ramp) are eliminated. For example in Figure 2(middle-row), the background in most of these face images does not contribute to the curvature histogram. The curvature index or shape index is rescaled and shifted to the range $[0, 1]$ as is done in [7]. A histogram is then computed of the valid index values over an entire image.

The second feature used is phase. The phase is simply defined as $P(\mathbf{p}, \sigma) = atan2\left(I_y(\mathbf{p}, \sigma), I_x(\mathbf{p}, \sigma)\right)$. Note that P is defined only at those locations where C is and ignored elsewhere. As with the curvature index P is rescaled and shifted to lie between the interval $[0, 1]$.

At different scales different local structures are observed and, therefore, multi-scale histograms are a more robust representation. Consequently, a feature vector is defined for an image I as the vector $V_i = \langle H_c(\sigma_1) \ldots H_c(\sigma_n), H_p(\sigma_1) \ldots H_p(\sigma_n) \rangle$ where H_p and H_c are the curvature and phase histograms respectively. We found that using 5 scales gives good results and the scales are $1 \cdots 4$ in steps of half an octave.

C. Matching feature histograms: Two feature vectors are compared using normalized cross-covariance defined as

$$d_{ij} = \frac{V_i^{(m)} \cdot V_j^{(m)}}{\left\|V_i^{(m)}\right\| \left\|V_j^{(m)}\right\|}$$

where $V_i^{(m)} = V_i - mean(V_i)$.

Retrieval is carried out as follows. A query image is selected and the query histogram vector V_q is correlated with the database histogram vectors V_i using the above formula. Then the images are ranked by their correlation score and displayed to the user. In this implementation, and for evaluation purposes, the ranks are computed in advance, since every query image is also a database image.

2.2.1 Experiments

The curvature-phase method is tested using two databases. The first is a trademark database of 2048 images obtained from the US Patent and Trademark Office (PTO). The images obtained from the PTO are large, binary and are converted to gray-level and reduced for the experiments. The second database is a collection of 1561 assorted gray-level images. This database has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different background(s) and a small number of other miscellaneous objects such as houses. These images were obtained from the Internet and the Corel photo-cd collection and were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry.

In the following experiments an image is selected and submitted as a query. The objective of this query is stated and the relevant images are decided in advance. Then the retrieval instances are gauged against the stated objective. In general, objectives of the form 'extract images similar in appearance to the query' will be posed to the retrieval algorithm. A measure of the performance of the retrieval engine can be obtained by examining the recall/precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved and precision is the proportion of retrieved material that is relevant [31]. It is a standard widely used in the information retrieval community and is one that is adopted here.

Queries were submitted each to the trademark and assorted image collection for the purpose of computing recall/precision. The judgment of relevance is qualitative. For each query in both databases the relevant images were decided in advance. These were restricted to 48. The top 48 ranks were then examined to check the proportion of retrieved images that were relevant. All images not retrieved within 48 were assigned a rank equal to the size of the database. That is, they are not considered retrieved. These ranks were used to interpolate and extrapolate precision at all recall points. In the case of assorted images relevance is easier to determine and more similar for different people. However in the trademark case it can be quite difficult and therefore the recall-precision can be subject to some error. The recall/precision results are summarized in Table 1 and both databases are individually discussed below.

Figure 1 shows the performance of the algorithm on the trademark images. Each strip depicts the top 8 retrievals, given the leftmost as the query. Most of the shapes have roughly the same structure as the query. Note that, outline and solid figures are treated similarly (see rows one and two in Figure 1). Six queries were submitted for the purpose of computing recall-precision in Table 1.

Experiments are also carried out with assorted gray level images. Six queries submitted for recall-precision are shown in Figure 2. The left most image in each row is the query and is also the first retrieved. The rest from-left to right are seven retrievals depicted in rank order. Note that, flat portions of the background are never considered because the principal curvatures are very close to zero and therefore do not contribute to the final score. Thus, for example, the flat background in Figure 2(second row) is not used. Notice that visually similar images are retrieved even when there is some change in the background (row 1). This is because the dominant object contributes most to the histograms. In using a single scale poorer results are achieved and background affects the results more significantly.

The results of these examples are discussed below, with the precision over all recall points depicted in parentheses. For comparison the best text retrieval engines have an average precision of 50%:

1. Find similar cars(65%). Pictures of cars viewed from similar orientations appear in the top ranks because of the contribution of the phase histogram. This result also shows that some background variation can be tolerated. The eighth retrieval although a car is a mismatch and is not considered.

2. Find same face(87.4%) and find similar faces: In the face query the objective is to find the same face. In experiments with a University of Bern face database of 300 faces with a 10 relevant faces each, the average precision over all recall points for all 300 queries was 78%. It should be noted that the system presented here works well for faces with the same representation and parameters used for all the other databases. There is no specific "tuning" or learning involved to retrieve faces. The query "find similar faces" resulted in a 100% precision at 48 ranks because there are far more faces than 48. Therefore, it was not used in the final precision computation.

3. Find dark textured apes (64.2%). The ape query results in several other light textured apes and country scenes with similar texture. Although these are not mis-matches they are not consistent with the intent of the query which is to find dark textured apes.

4. Find other patas monkeys. (47.1%) Here there are 16 patas monkeys in all and 9 within a small view variation. However, here the whole image is being matched so the number of relevant patas monkeys is 16. The precision is low because the method cannot

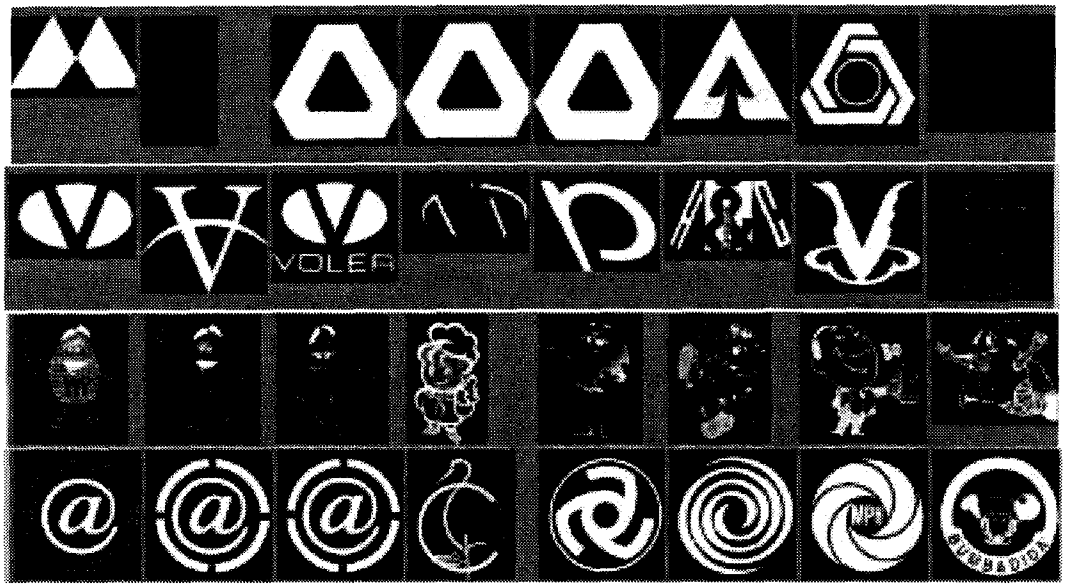


Figure 1: Trademark retrieval using Curvature and Phase

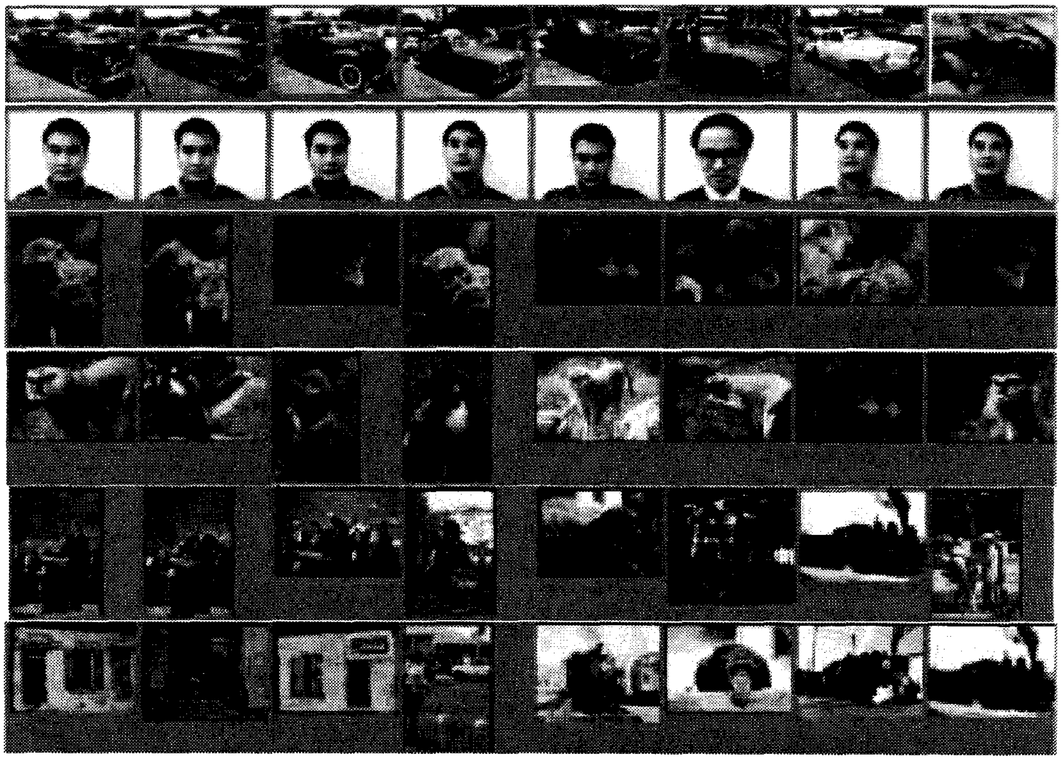


Figure 2: Image retrieval using Curvature and Phase

distinguish between light and dark textures, leading to irrelevant images. Note, that it finds other apes, dark textured ones, but those are deemed irrelevant with respect to the query.

5. Given a wall with a Coca Cola logo find other Coca Cola images (63.8%). This query clearly depicts the limitation of global matching. Although all three database images that had a certain texture of the wall (also had Coca Cola logos) were retrieved (100% precision), two other very dissimilar images with coca-cola logos were not.

6. Scenes with Bill Clinton (72.8%). The retrieval in this case results in several mismatches. However, three of the four are retrieved in succession at the top and the scenes appear visually similar.

While the queries presented here are not "optimal"

Table 1: Precision at standard recall points for six Queries

Recall	0	10	20	30	40	50	60	70	80	90	100
Precision(trademark) %	100	93.2	93.2	85.2	76.3	74.5	59.5	45.5	27.2	9.0	9.0
Precision(assorted) %	100	92.6	90.0	88.3	87.0	86.8	83.8	65.9	21.3	12.0	1.4

average(trademark)	61.1%
average(assorted)	66.3%

with respect to the design constraints of global similarity retrieval, they are however, realistic queries that can be posed to the system. Mismatches can and do occur. The first is the case where the global appearance is very different. The Coca Cola retrieval is a good example of this. Second, mismatches can occur at the algorithmic level. Histograms coarsely represent spatial information and therefore will admit images with non-trivial deformations. The recall/precision presented here compares well with text retrieval. The time per retrieval is of the order of milli-seconds. In the next section we discuss the application of the presented technique to a database of 63000 images.

2.3 Trademark Retrieval

The system indexes about 68,000 trademarks from the US Patent and Trademark office in the design only category. These trademarks are binary images. In addition, associated text consists of a design code that designates the type of trademark, the goods and services associated with the trademark, a serial number and a short descriptive text.

The system for browsing and retrieving trademarks is illustrated in Figure 3. The netscape/Java user interface has two search-able parts. On the left a panel is included to initiate search using text. Any or all of the fields can be used to enter a query. In this example, the text "Merriam Webster' is entered and all images associated with it are retrieved using the Inquery [4] text search engine. The user can then use any of the example pictures to search for images that are similar. In the specific example shown, The second image is selected and retrieved results are displayed on the right panel. The user can then continue to search using any of the displayed pictures as the query.

In this section we adapt the curvature/phase histograms to retrieve visually similar trademarks. The following steps are performed to retrieve images.

Preprocessing: Each binary image in the database is first size normalized, by clipping. Then they are converted to gray-scale and reduced in size.

Computation of Histograms: Each processed image is divided into four equal rectangular regions. This is different than constructing a histogram based on pixels of the entire image. This is because in scaling the images to a large collection, we found that the added degree of spatial resolution significantly improves the retrieval performance. The curvature and phase histograms are computed for each tile at three scale. A histogram descriptor of the image is obtained by concatenating all the individual histograms across scales and regions.

These two steps are conducted off-line.

Execution: The image search server begins by loading all the histograms into memory. Then it waits on a port for a query. A CGI client transmits the query to the server. Its histograms are matched with the ones in the database. The match scores are ranked and the top N requested retrievals are returned.

2.3.1 Examples

In Figure 3, the user typed in Merriam Webster in the text window. The system searches for trademarks which have either Merriam or Webster in th associated text and displays them. Here, the first two trademarks (first two images in the left window) belong to Merriam Webster. In this example, the user has chosen to 'click' the second image and search for images of similar trademarks. This search is based entirely on the image and the results are displayed in the right window in rank order. Retrieval takes a few seconds and is done by comparing histograms of all 63,718 trademarks on the fly.

The original image is returned as the first result (as it should be). The images in positions 2,3 and 5 in the second window all contain circles inside squares and this configuration is similar to that of the query. Most of the other images are of objects contained inside a roughly square box and this is reasonable considering that similarity is defined on the basis of the entire image rather than a part of the image.

The second example is shown in Figure 4. Here the user has typed in the word Apple. The system returns trademarks associated with the word Apple. The user queries using Apple computer's logo (the image in the second row, first column of the first window). Images retrieved in response to this query are shown in the right window. The first eight retrievals are all copies of Apple Computer's trademark (Apple used the same trademark for a number of other goods and so there are multiple copies of the trademark in the database). Trademarks number 9 and 10 look remarkably similar to Apple's trademark. They are considered valid trademarks because they are used for goods and services in areas other than computers. Trademark 13 is another version of Apple Computer's logo but with lines in the middle. Although somewhat visually different it is still retrieved

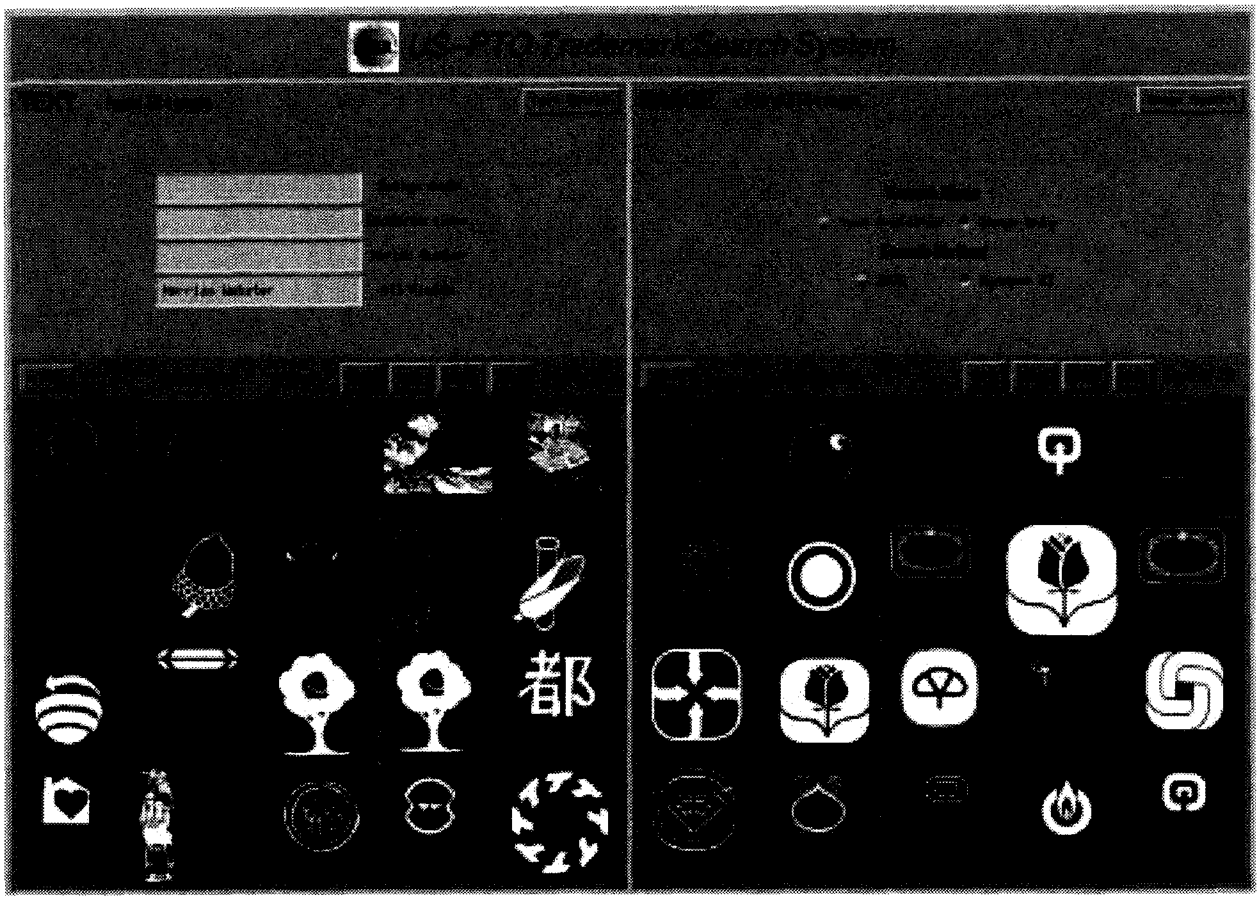


Figure 3: Retrieval in response to a "Merriam Webster" query

in the high ranks. Image 14 is an interesting example of a mistake made by the system. Although the image is not of an apple, the image has similar distributions of curvature and phase as is clear by looking at it.

The third example demonstrates combining text and visual appearance for searching. We use the same apple image obtained in the previous image as the image query. However, in the text box we now type "computer" and turn the text combination mode on. We now search for trademarks which are visually similar to the apple query image but also have the words computer associated with them. The results are shown in Figure 5 on the right-hand side. Notice that the first image is the same as the query image. The second image is an actual conflict. The image is a logo which belongs to the Atlanta Macintosh User's Group. The text describes the image as a peach but visually one can see how the two images may be confused with each other (which is the basis on which trademark conflicts are adjudicated). This example shows that it does not suffice to go by the text descriptions alone and image search is useful for trademarks. Notice that the fourth image which some people describe as an apple and others as a tomato is also described in the text as an apple.

The system has been tried on a variety of different examples of both two dimensional and three dimensional pictures of trademarks and had worked quite well. Clearly, there are issues of how quantitative results can be obtained for such large image databases (it is not feasible for a person to look at every image in the database to determine whether it is similar). In future work, we hope to evolve a mechanism for quantitative testing on such large databases. It will also be important to use more of the textual information to determine trademark conflicts.

3 Word Segmentation in Handwritten Archival Manuscripts

There are many single author historical handwritten manuscripts which would be useful to index and search. Examples of these large archives are the papers of George Washington, Margaret Sanger and W. E. B Dubois. Currently, much of this work is done manually. For example, 50,000 pages of Margaret Sanger's work were recently indexed and placed on a CDROM. A page by page index was created manually. It would be useful to automatically create an index for an historical archive similar to the index at the back of a printed book. To achieve this objective a semi-automatic scheme for indexing such documents have been proposed in [23, 22, 21]. In this scheme known as *Word Spotting* the document page is segmented into words. Lists

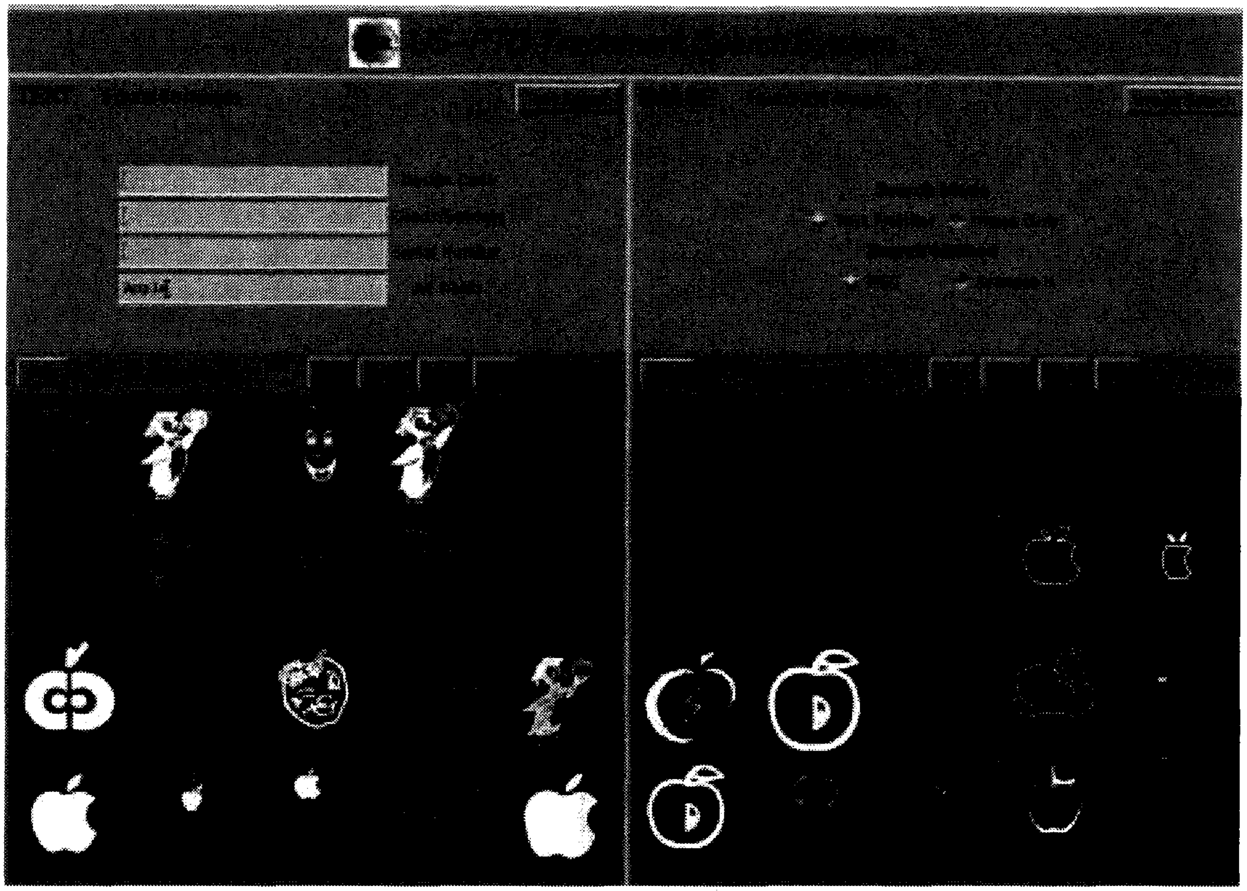


Figure 4: Retrieval in response to the query "Apple"

of words containing multiple instances of the same word are then created by matching word images against each other. A user then provides the ASCII equivalent to a representative word image from each list and the links to the original documents are automatically generated. The earlier work in [23, 22, 21] concentrated on the matching strategies and did not address full page segmentation issues in handwritten documents. In this paper, we propose a new algorithm for word segmentation in document images by considering the scale space behavior of blobs in line images.

Most existing document analysis systems have been developed for machine printed text. There has been little work on word segmentation for handwritten documents. Most of this work has been applied to special kinds of pages - for example, addresses or "clean" pages which have been written specifically for testing the document analysis systems. Historical manuscripts suffer from many problems including noise, shine through and other artifacts due to aging and degradation. No good techniques exist to segment words from such handwritten manuscripts. Further, scale space techniques have not been applied to this problem before.

We outline the various steps in the segmentation algorithm below.

The input to the system is a grey level document image. The image is processed to remove horizontal and vertical line segments likely to interfere with later operations. The page is then dissected into lines using projection analysis techniques modified for gray scale image. The projection function is smoothed with a Gaussian filter (low pass filtering) to eliminate false alarms and the positions of the local maxima (i.e., white space between the lines) is detected. Line segmentation, though not essential is useful in breaking up connected ascenders and descenders and also in deriving an automatic scale selection mechanism. The line images are smoothed and then convolved with second order anisotropic Gaussian derivative filters to create a scale space and the *blob* like features which arise from this representation give us the focus of attention regions (i.e., words in the original document image). The problem of automatic scale selection for filtering the document is also addressed. We have come up with an efficient heuristic for scale selection whereby the correct scale for blob extraction is obtained by finding the scale maxima of the blob extent. A connected component analysis of the blob image followed by a reverse mapping of the bounding boxes allows us to extract the words. The box is then extended vertically to include the ascenders and descenders. Our approach to word segmentation is novel as it is the first algorithm which utilizes the inherent scale space behavior of words

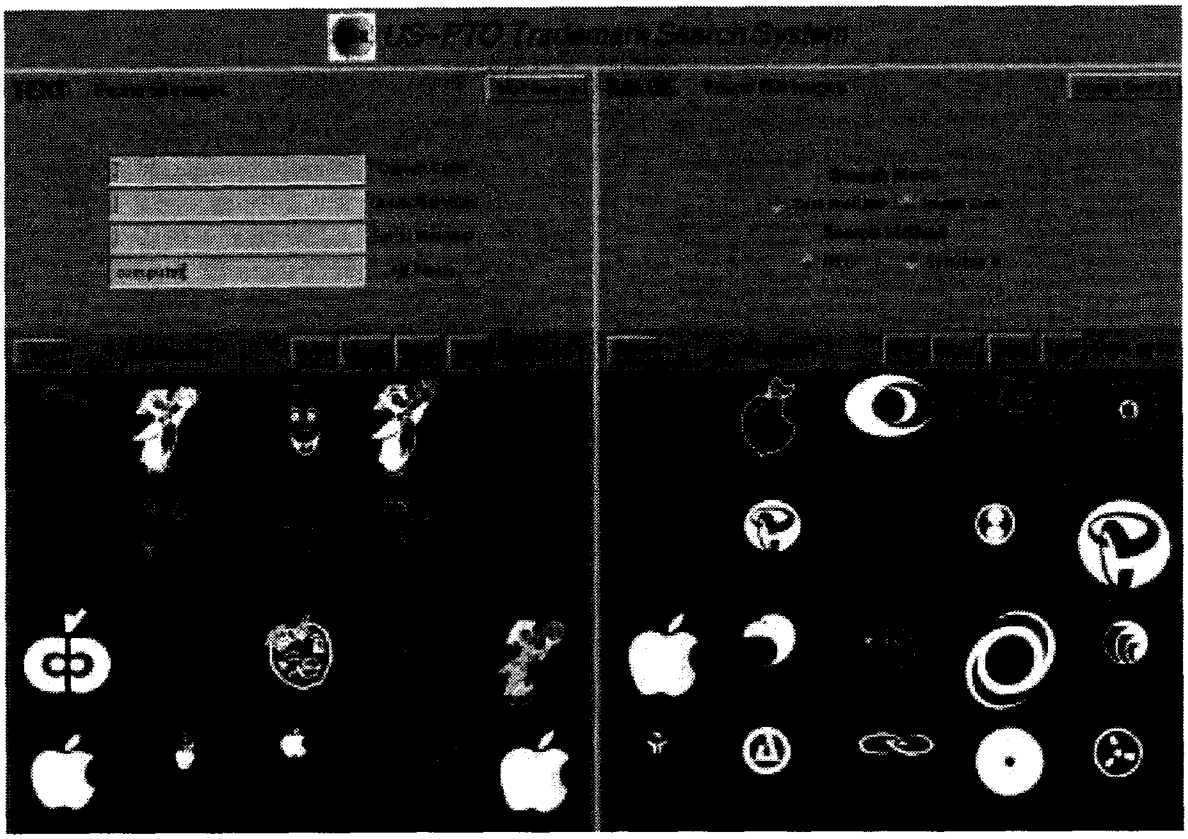


Figure 5: Retrieval in response to the query "Apple" limited to text searches

in grey level document images.

3.1 Related Work

Most recognition systems mask the issue of segmentation by considering well segmented patterns [2] or using words written in boxes whose location is known [8]. However, correct segmentation is crucial in full page document analysis and directly relates to the performance of the entire system. We present some of the work in word and character segmentation.

3.1.1 Word and character segmentation

Character segmentation schemes proposed in the literature have mostly been developed for machine printed characters and work poorly when extended to handwritten text. An excellent survey of the various schemes has been presented in [5].

Very few papers have dealt exclusively with issues of word segmentation in handwritten documents and even they have focussed on identifying gaps using geometric distance metrics between connected components. Seni and Cohen [28] evaluate eight different distance measures between pairs of connected component for word segmentation in handwritten text. In [19] the distance between the convex hulls is used. Srihari et all [29] present techniques for line separation and then word segmentation using a neural network. However, the existing word segmentation strategies have certain limitations

- Almost all the above methods require binary images. Also, they have been tried only on clean white self-written pages and not manuscripts.
- Most of the techniques have been developed for machine printed characters and not handwritten words. The difficulty faced in word segmentation is in combining discrete characters into words.
- Most researchers focus only on word recognition algorithms and considered a database of clean images with well segmented words, [1] is one such example.

Only a few [29] have performed full, handwritten page segmentation. However, we feel that schemes such as [29] are not applicable for page segmentation in manuscript images for the reasons mentioned below.

- Efficient image binarization is difficult on manuscript images containing noise and shine through.
- Connected ascenders and descenders have to be separated.
- Prior character segmentation was required to perform word segmentation and accurate character segmentation in cursive writing is a difficult problem. Also the examples shown are contrived (self written) and do not handle problems in naturally written documents.

3.2 Word Segmentation

Modeling the human cognitive processes to derive a computational methodology for handwritten word segmentation with performance close to the human visual system is quite complex due to the following characteristics of handwritten text.

- The handwriting style may be cursive or discrete. In case of discrete handwriting characters have to be combined to form words.
- Unlike machine printed text, handwritten text is not uniformly spaced.
- Scale problem. For example, the size of characters in a header is generally larger than the average size of the characters in the body of the document.
- Ascenders and descenders are frequently connceted and words may be present at different orientations.
- Noise, artifacts, aging and other degradation of the document. Another problem is the presence of background handwriting or shine through.

We now present a brief background to scale space and how we have applied it to document analysis.

3.3 Scale space and document analysis

Scale space theory deals with the notion and importance of scale in any physical observation i.e. objects or features are relevant only at particular scales and meaningless at other scales [14, 11, 18]. In scale space, starting from an original image, successively smoothed images are generated along the scale dimension. It has been shown by several researchers [14, 11, 18] that the Gaussian uniquely generates the linear scale space of the image when certain conditions are imposed.

We feel that *scale space* also provides an ideal framework for document analysis. We may regard a document to be formed of features at multiple scales. Intuitively, at a finer scale we have characters and at larger scales we have words, phrases, lines and other structures. Hence, we may also say that there exists a scale at which we may derive words from a document image. We would, therefore, like to have an image representation which makes the features at that scale (words in this case) explicit : i.e. no further processing should be required to locate the words.

3.3.1 Formal definition

The linear scale space representation of a continuous signal with arbitrary dimensions consists of building a one parameter family of signals derived from the original one in which the details are progressively removed. Let f: $\Re^N \rightarrow \Re$ represent any given signal. Then, the scale space representation $I: \Re^N \times \Re_+ \rightarrow \Re$ is defined by letting the scale space representation at zero scale be equal to the original signal $I(\cdot;\,0) = f$ and for $\sigma > 0$,

$$I(\cdot;\sigma) = G(\cdot;\sigma) \star f, \qquad (1)$$

$$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{(2\sigma^2)}} \qquad (2)$$

where G is the Gaussian kernel in two dimensions and σ is the scale parameter. We now describe the various stages in our algorithm.

3.4 Preprocessing

These handwritten manuscripts have been subjected to degradation such as fading and introduction of artifacts. The images provided to us are scanned versions of the photocopies of the original manuscripts. In the process of photocopying, horizontal and vertical black line segments/margins were introduced. Horizontal lines are also present within the text. The purpose of the preprocessing step is to remove some of these margins and lines so that they will not interfere with the blob analysis stage. The details of the pre-processing step are omitted here.

3.5 Line segmentation

Line segmentation allows the ascenders and descenders of consecutive lines to be separated. In the manuscripts it is observed that the lines consist of a series of horizontal components from left to right. Projection profile techniques have been widely used in line and word segmentation for machine printed documents [12]. In this technique a 1D function of the pixel values is obtained by projecting the binary image onto the horizontal or vertical axis. We use a modified version of the same algorithm extended to gray scale images. Let $f(x,y)$ be the intensity value of a pixel (x,y) in a gray scale image. Then, we define the vertical projection profile as

$$P(y) = \sum_{x=0}^{W} f(x,y) \qquad (3)$$

where W is the width of the image. Figure 6 shows a section of an image (rotated by 90 deg.) in (a) and its projection profile in (b). The distinct local peaks in the profile corresponds to the white space between the lines and distinct local minima corresponds to the text (black ink). Line segmentation, therefore, involves detecting the position of the local maxima. However, the projection profile has a number of false local maxima and minima. The projection function $P(y)$ is therefore, smoothed with a Gaussian (low pass) filter to eliminate false alarms and reduce sensitivity to noise. A smoothed profile is shown in (c). The local maxima is then obtained from the first derivative of the projection function by solving for y such that :

$$P'(y) = P(y) \star G_y = 0 \qquad (4)$$

The line segmentation technique is robust to variations in the size of the lines and has been tested on a wide range of handwritten pages. The next step after line segmentation is to create a scale space of the line images for blob analysis.

3.6 Blob analysis

Now we examine each line image individually to extract the words. A word image is composed of discrete char-

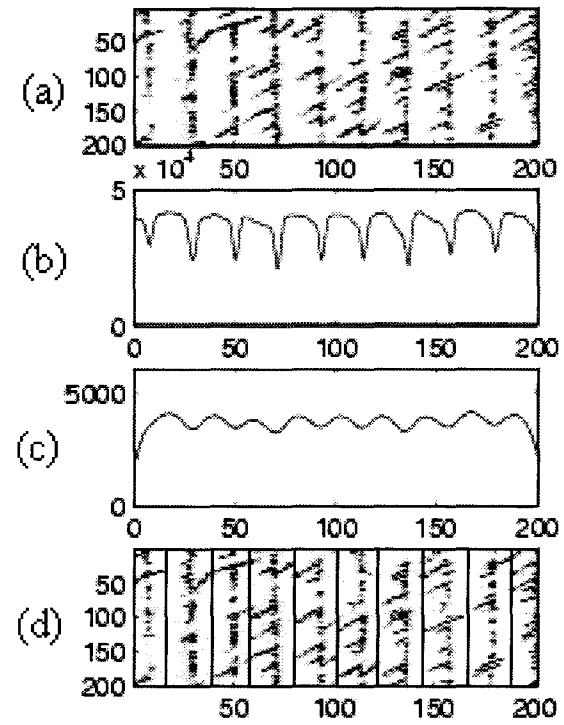


Figure 6: (a) A section of an image, (b) projection profile, (c) smoothed projection profile (d) line segmented image

acters, connected characters or a combination of the two. We would like to merge these sub-units into a single meaningful entity which is a word. This may be achieved by forming a blob-like representation of the image. A blob can be regarded as a connected region in space. The traditional way of forming a blob is to use a Laplacian of a Gaussian (LOG) [17] as the LOG is a popular operator and frequently used in blob detection and a variety of multi-scale image analysis tasks [3, 25, 17]. We have used a differential expression similar to a LOG for creating a multi-scale representation for blob detection. However, our differential expression differs in that we combine second order partial Gaussian derivatives along the two orientations at different scales. In the next section we present the motivation for using an anisotropic derivative operator.

3.6.1 Non uniform Gaussian filters

In this section some properties which characterize writing are used to formulate an approach to filtering words. In [17] Lindeberg observes that maxima in scale-space occur at a scale proportional to the spatial dimensions of the blob. If we observe a word we may see that the spatial extent of the word is determined by the following :

1. The individual characters determine the height (y dimension) of the word and
2. The length (x dimension) is determined by the number of characters in it.

A word generally contains more than one character and has an aspect ratio greater than one. As the x dimension of the word is larger than the y dimension, the spatial filtering frequency should also be higher in the y dimension as compared to the x dimension. This domain specific knowledge allows us to move from isotropic (same scale in both directions) to anisotropic operators. We choose the x dimension scale to be larger than the y dimension to correspond to the spatial structure of the word. Therefore, our approach for word segmentation, is based on the idea of a directional scale (i.e. generating an image representation by using Gaussian derivative operators at different scales for each of the two Cartesian coordinate axes) is in agreement with Lindeberg's observation that spatial dimensions are related to the scale. We define our anisotropic Gaussian filter as

$$G(x,y;\sigma_x,\sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y}e^{-(\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2})} \quad (5)$$

We may also define the multiplication factor η as

$$\eta = \frac{\sigma_x}{\sigma_y} \quad (6)$$

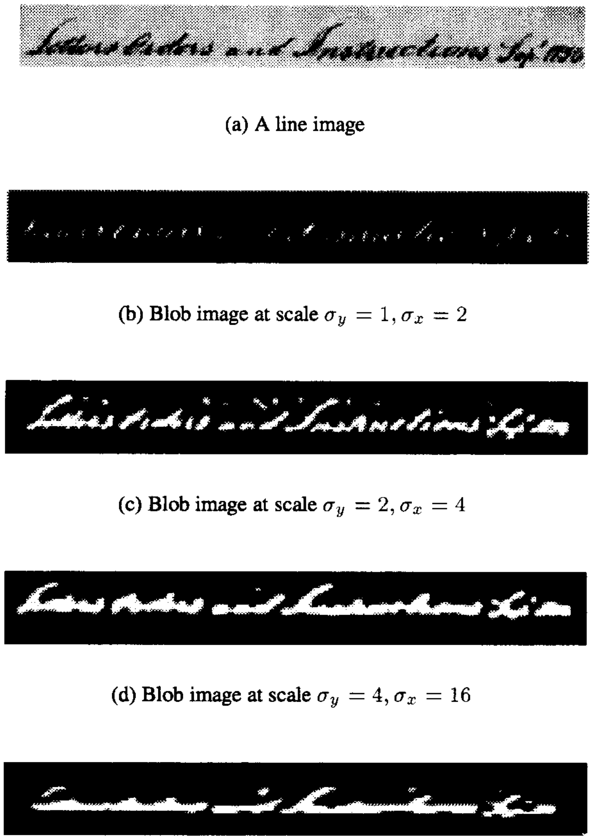
In the scale selection section we will show that the average aspect ratio or the multiplication factor η lies between three and five for most of the handwritten documents available to us. Also the response of the anisotropic Gaussian filter (measured as the spatial extent of the *blobs* formed) is maximum in this range. For the above Gaussian, the second order anisotropic Gaussian differential operator $L(x,y;\sigma_x,\sigma_y)$ is defined as

$$L(x,y;\sigma_x,\sigma_y) = G_{xx}(x,y;\sigma_x) + G_{yy}(x,y;\sigma_y) \quad (7)$$

A scale space representation of the line images is constructed by convolving the image with equation 7 Consider a two dimensional image $f(x,y)$, then the corresponding output image is

$$\begin{aligned} I(x,y;\sigma_x,\sigma_y) &= G_{xx}(\cdot;\sigma_x) \star f(x,y) \\ &\quad + G_{yy}(\cdot;\sigma_y) \star f(x,y) \quad (8) \\ &= G_{xx}(\cdot;\sigma_x) \star f(x,y) \\ &\quad + G_{yy}(\cdot;\eta\sigma_x) \star f(x,y) \quad (9) \end{aligned}$$

The main features which arise from a scale space representation are blob-like (i.e., connected regions either brighter or darker than the background). The sign of I may then be used to make a classification of the 3-D intensity surface into foreground and background. For example consider the line image in Figure 7(a). The figures show the blob images $I(x,y;\sigma_x,\sigma_y)$ at increasing scale values. Figure 7(b) shows that at a lower scale the blob image consists of character blobs. As we increase the scale, character blobs give rise to word blobs (Figure 7(c) and Figure 7(d)). This is indicative of the phenomenon of merging in blobs. It is seen that for certain scale values the blobs and hence the words are correctly delineated (Figure 7(d)). A further increase in the scale value may not necessarily cause word blobs to merge together and other phenomenon such as splitting is also observed. These figures show that their exists a scale at which it is possible to delineate words. In the next section we present an approach to automatic scale selection for blob extraction.



(a) A line image

(b) Blob image at scale $\sigma_y = 1, \sigma_x = 2$

(c) Blob image at scale $\sigma_y = 2, \sigma_x = 4$

(d) Blob image at scale $\sigma_y = 4, \sigma_x = 16$

(e) Blob image at scale $\sigma_y = 6, \sigma_x = 36$

Figure 7: A line image and the output at different scales

3.7 Choice of scale

Scale space analysis does not address the problem of scale selection. The solution to this problem depends on the particular application and requires the use of prior information to guide the scale selection procedure. Some of our work in scale selection draws motivation from Lindeberg's observation [17] that the maximum response in both scale and space is obtained at a scale proportional to the dimension of the object. A document image consists of structures such as characters, words and lines at different scales. However, as compared to other types of images, document images have this unique property that a large variation in scale is not required to extract a particular type of structure. For example, all the words are essentially close together in terms of their scale and therefore, can be extracted without a large variation in the scale parameter. Hence, there exists a scale where each of the individual word forms a distinct blob. The output (blob) is then maximum at this value of the scale parameter. We show elsewhere [24] that this scale is a function of the vertical dimension of the word if the aspect ratio is fixed.

Our algorithm requires selecting σ_y and the multiplication factor η for blob extraction.

A base scale is obtained by using the height of the line: i.e., an estimate of σ_y is obtained as a fraction of the line height.

$$\sigma_y = k \times \text{Line height} \qquad (10)$$

where $0 < k < 1$, the nearby scales are then examined to determine the maximum over scales. For our specific implementation we have used $k = 0.1$ and sampled σ_y at intervals of 0.3. The two values were determined experimentally and worked well over a wide range of images. The scales are then picked.

The details of the scale selection process are given elsewhere [24]. Breifly, by plotting a graph which shows the extent of the blobs versus the eta for a constant σ_y, we have shown that the maximum usually occurs for values of η between 3 and 5 (see [24]) for a large number of images. Thus, we choose $\eta = 4$.

3.8 Blob extraction and post processing

After the word blobs have been obtained at the correct scale they define the focus of attention regions which correspond to the actual words. Hence, these blobs have to be mapped back to the original image to locate the words. A widely used procedure is to enclose the blob in a bounding box which can be obtained through connected component analysis. In a blob representation of the word, localization is not maintained. Also parts of the words, especially the ascenders and descenders, are lost due to the earlier operations of line segmentation and smoothing (blurring). Therefore, the above bounding box is extended in the vertical direction to include these ascenders and descenders. At this stage an area/ratio filter is used to remove small structures due to noise.

3.9 Results

The technique was tried on 30 randomly picked images from different sections of the George Washington corpus of $6,400$ images and a few images from the archive of papers of Erasmus Hudson. This allowed us to test on algorithm on wide range of handwritten documents such as letters, notebook pages etc. To reduce the runtime, the images have been smoothed and sub-sampled to a quarter of their original size. The algorithm takes 120 seconds to segment a document page of size 800 x

600 pixels on a PC with a 200 MHz pentium processor running LINUX. A segmentation accuracy ranging from 77 – 96 percent with an average accuracy around 87.6 percent was observed. Figure 8 shows a segmented image with bounding boxes drawn on the extracted words. The method worked well even on faded, noisy images and Table 1 shows the results averaged over a set of 30 images.

The first column indicates the average no. of distinct words in a page as seen by a human observer. The second column indicates the % of words detected by the algorithm i.e, words with a bounding box around them, this includes words correctly segmented, fragmented and combined together. This measure is required as some of the words may be sufficiently small or faint to be mistaken for noise or an artifact. The next column indicate the % of words fragmented. Word fragmentation occurs if a character or characters in a word have separate bounding boxes or if 50 percent or greater of a character in a word is not detected. Line fragmentation occurs due to the dissection of the image into lines. A word is line fragmented if 50 percent or greater of a character lies outside the top or bottom edges of the bounding box. The sixth column indicates the words which are combined together. These are multiple words in the same bounding box and occur due to the choice of a larger scale in segmentation. The last column gives the percentage of correctly segmented words.

Avg. words per image	% words dete-cted	% frag-mented words +line	% words comb-ined	% words correctly segmen-ted
220	99.12	1.75 +0.86	8.9	87.6

Table 2: Table of segmentation results

4 Conclusion

This paper has described the multimedia indexing and retrieval work being done at the Center for Intelligent Information Retrieval. We have described work on a system for multi-modal retrieval combining text and image retrieval as well as word segmentation for handwritten archives. The research described is part of an on-going research effort focused on indexing and retrieving multimedia information in as many ways as possible. The work described here has many applications, principally in the creation of the digital libraries of the future.

5 Acknowledgements

A number of students, staff and faculty have contributed to the work described in this paper. The appearance based image retrieval work was done by S. Chandu Ravela while the color based retrieval work was done by Madirakshi Das. Victor Wu worked on extracting text from images while Nitin Srimal worked on word segmentation using scale space blobs. Tom Michel and Kamal Souccar worked on the text retrieval as well the interfaces for the multi-modal retrieval. Joseph Daverin, David Hirvonen and Adam Jenkins provided programming support for different parts of this work. Bruce Croft contributed to the text and multi-modal retrieval and James Allan provided suggestions on text retrieval.

References

[1] A.J. Robinson A.W. Senior. An off-line cursive handwriting recognition system. *IEEE transactions on PAMI*, 3:309–321, 1998.

[2] H. S. Baird and K. Thompson. Reading Chess. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(6):552–559, 1990.

[3] D. Blostein and N. Ahuja. A multiscale region detector. *Computer Vision Graphics and Image Processing*, 45:22–41, 1989.

[4] J. P. Callan, W. B. Croft, and S. M. Harding. The inquery retrieval system. In *Proceedings of the 3^{rd} International Conference on Database and Expert System Applications*, pages 78–83, 1992.

[5] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on PAMI*, 18:690–706, July 1996.

[6] M. Das, R. Manmatha, and E. M. Riseman. Indexing flowers by color names using domain knowledge-driven segmentation. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 94–99, Princeton, NJ, Oct. 1998.

[7] Chitra Dorai and Anil Jain. Cosmos - a representation scheme for free form surfaces. In *Proc. 5th Intl. Conf. on Computer Vision*, pages 1024–1029, 1995.

[8] R. O. Duda and P. E. Hart. Experiments in recognition of hand-printed text. In *AFIPS Conference Proceedings*, pages 1139–1149, 1968.

[9] J.R. Bach et al. The virage image search engine: An open framework for image management. In *SPIE conf. on Storage and Retrieval for Still Image and Video Databases IV*, pages 133–156, 1996.

[10] Myron Flickner et al. Query by image and video content: The qbic system. *IEEE Computer Magazine*, pages 23–30, Sept. 1995.

[11] L. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Utrecht, Holland, 1993.

[12] J. Ha, R. M. Haralick, and I. T. Phillips. Document page decomposition by the bounding-box projection technique. In *ICDAR*, pages 1119–1122, 1995.

[13] K. Shields J. P. Eakins and J. M. Boardman. Artisan - a shape retrieval system based on boundary family indexing. In *In Proc. SPIE conf. on Storage and Retrieval for Image and Video Databases IV, vol. 2670, San Jose*, pages 17–28, Feb 1996.

[14] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[15] J. J. Koenderink and A. J. Van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.

[16] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[17] T. Lindeberg. On scale selection for differential operators. In *Eighth Scandinavian Conference on Image Analysis*, pages 857–866, 1993.

[18] Tony Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[19] U. Mahadevan and R. C. Nagabushnam. Gap metrics for word separation in handwritten lines. In *ICDAR*, pages 124–127, 1995.

[20] R. Manmatha. Multimedia indexing and retrieval at the center for intelligent information retrieval. In *Symposium on Document Image Understanting Technology, SDIUT'97*, Cambridge, U.K., April 1997. 4th European Conf. Computer Vision, Institute for Advanced Computer Studies, University of Maryland.

[21] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten manuscripts. In Mark Maybury, editor, *Intelligent Multi-media Information Retrieval.* AAAI/MIT Press, April 1998.

[22] R. Manmatha, Chengfeng Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 631–637, 1996.

[23] R. Manmatha, Chengfeng Han, E. M. Riseman, and W. B. Croft. Indexing handwriting using word matching. In *Digital Libraries '96: 1st ACM International Conference on Digital Libraries*, pages 151–159, 1996.

[24] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten manuscripts. In *submitted to the IEEE International Conference on Computer Vision (ICCV'99)*, Sep. 1999.

[25] D. Marr. *Vision.* W.H. Freeman: San Francisco, 1982.

[26] S. Ravela and R. Manmatha. Image retrieval by appearance. In *In the Proc. of the 20th Intl. Conf. on Research and Development in Information Retrieval (SIGIR'97)*, pages 278–285, July 1997.

[27] S. Ravela and R. Manmatha. On computing global similarity in images. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 82–87, Princeton, NJ, Oct. 1998.

[28] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27:41–52, 1994.

[29] S. Srihari and G. Kim. Penman : A system for reading unconstrained handwritten page images. In *Symposium on document image understanding technology (SDIUT 97)*, pages 142–153, April 1997.

[30] Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[31] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[32] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.

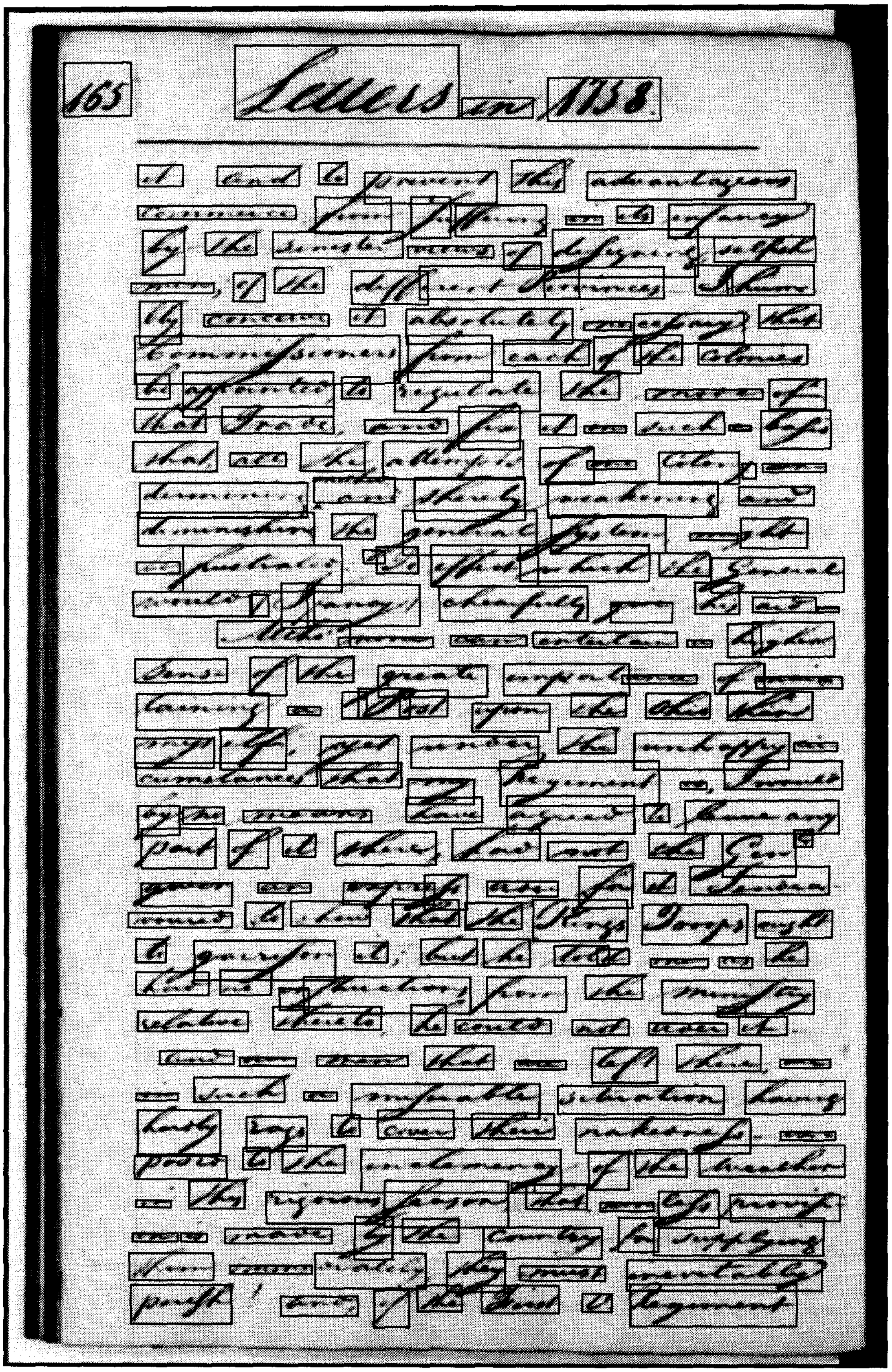


Figure 8: Segmentation result on a image 1670165.tif from the George Washington collection

[14] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984.

[15] J. J. Koenderink and A. J. Van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.

[16] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[17] T. Lindeberg. On scale selection for differential operators. In *Eighth Scandinavian Conference on Image Analysis*, pages 857–866, 1993.

[18] Tony Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[19] U. Mahadevan and R. C. Nagabushnam. Gap metrics for word separation in handwritten lines. In *ICDAR*, pages 124–127, 1995.

[20] R. Manmatha. Multimedia indexing and retrieval at the center for intelligent information retrieval. In *Symposium on Document Image Understanting Technology, SDIUT'97*, Cambridge, U.K., April 1997. 4th European Conf. Computer Vision, Institute for Advanced Computer Studies, University of Maryland.

[21] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten manuscripts. In Mark Maybury, editor, *Intelligent Multi-media Information Retrieval.* AAAI/MIT Press, April 1998.

[22] R. Manmatha, Chengfeng Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 631–637, 1996.

[23] R. Manmatha, Chengfeng Han, E. M. Riseman, and W. B. Croft. Indexing handwriting using word matching. In *Digital Libraries '96: 1st ACM International Conference on Digital Libraries*, pages 151–159, 1996.

[24] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten manuscripts. In *submitted to the IEEE International Conference on Computer Vision (ICCV'99)*, Sep. 1999.

[25] D. Marr. *Vision.* W.H. Freeman: San Francisco, 1982.

[26] S. Ravela and R. Manmatha. Image retrieval by appearance. In *In the Proc. of the 20th Intl. Conf. on Research and Development in Information Retrieval (SIGIR'97)*, pages 278–285, July 1997.

[27] S. Ravela and R. Manmatha. On computing global similarity in images. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 82–87, Princeton, NJ, Oct. 1998.

[28] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27:41–52, 1994.

[29] S. Srihari and G. Kim. Penman : A system for reading unconstrained handwritten page images. In *Symposium on document image understanding technology (SDIUT 97)*, pages 142–153, April 1997.

[30] Bart M. ter Har Romeny. *Geometry Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[31] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[32] A. P. Witkin. Scale-space filtering. In *Proc. Intl. Joint Conf. Art. Intell.*, pages 1019–1023, 1983.

Applications and Systems

Languages for Document Conversion: An Experiment with Programmable XML for Dynamic Documents

Robert Thibadeau, Ph.D., Jorge Balderas, Andrew Snyder
Universal Library Project, School of Computer Science,
Carnegie Mellon University
Pittsburgh, PA

John Nestor
XML for All, Inc.
Pittsburgh, PA

Abstract

This paper introduces the notion of a strongly dynamic document that may be useful in electronic commerce catalogs, advanced textbooks, document conversion languages, and other applications. It augments the existing XML web standard to include tagging for conditional interpretation. With this inclusion, a document can perform computation as well as simply feeding classed data to external computation.

Document conversion has been plagued for years with the lack of a reference language that is universally accepted. With a reference language it is possible to design pattern recognition that can yield unambiguous decisions. During the time that document conversion was strictly reserved to conversion to ASCII characters, and now UNICODE, this was not too much of a problem. But as it becomes clearer that other aspects of what gets placed on a page makes a difference, the absence of a language for universal page expression becomes a problem. Adobe Systems, Inc., has been able to make strong progress in this area because it has a proprietary, but complete, page description language (PDF which is generally just a subset of Postscript) along with strong technical expertise in such pertinent areas as fonts, graphics, page layout, and images. Adobe's Capture product represents state of the art in what can be done, but the proprietary, single source, nature of the underlying representations is a strong inducement to look elsewhere. The topic of this paper is an extension to XML that incorporates the programmability features of Postscript and PDF in order to yield a basis for general document representation.

In the web, the conventional way of implementing dynamic documents is to have external scripts that interact in prescribed manners with a marked-up page. This is particularly the case for a document that is composed from dynamic sources, such as databases. By extending markup to support conditional interpretation, dynamic documents can be composed and preserved without the need for writing special scripts for each document application. We show an operational E-commerce catalog that employs a version of this augmentation that we have called XML For All, or XFA

Introduction

Web sites generally divide into three components:

1. HTML to prescribe how pages will appear in a browser,

2. the HTTP server that prescribes some simple interactions between the web site and

actions on a page, and

3. CGI scripts and plugins, or computer programs with specific input and output format requirements, that prescribe application-specific interactions possible between actions on a page and the web site.

This paper is about a strategy that greatly reduces the need for the third, application specific, coding component through the introduction of strong dynamic documents. We also provide one in-depth analysis of such a document that is a self-contained electronic catalog directly applicable to E-commerce.

The HTTP server implements a handful of interactions, such as following hyperlinks, but the computer programs effect much of the action, such as database posting and retrieval. Computer programs cannot be written in standard HTML, because the HyperText Markup Language is a collection of simple declarative predicates, such as "title" and "href" applied over specific text strings in a document. There is no defined notion of "a variable," or of "conditional interpretation" in HTML scripting. All conditional interpretation is carried out by the computer programs that are either intrinsic to the HTTP server or to the specific application.

Recently, the World Wide Web consortium has recommended an eXtended Markup Language, XML, to integrate the original parent of HTML, the Standard Graphics Markup Language, SGML, into the web. Where HTML prescribed a small repertoire of functions suitable to small documents that are web pages, SGML prescribes a very large and open-ended repertoire of functions suitable to marking up large documents for multiple simultaneous purposes. So, for example, in SGML a single set of tags may be employed to mark up a document both as a database of information and for display as a Postscript document. Another example is when a document has two languages for presentation, such as English and French, but shares common graphics and pictures. XML, a common ground for SGML and HTML, now provides the means of manipulating such multipurposed documents on the web.

XML, like SGML and unlike HTML, incorporates explicit notions of application specific variables in its DTDs (Document Type Declarations). But, also like SGML, XML does not incorporate conditional interpretation. XML, therefore, does not have the expressive power of a computer programming language. Without this expressive power the kinds of documents that can be represented in XML is highly limited - typically to static, non-interactive, forms such as printing on paper. Advocates of XML and SGML suggest that the XML parsing engine and the interpreter, as might be provided in CGI scripts and plugins, would provide the conditional interpretation. But, experience in computer science suggests that multipurposing a document often requires conditional interpretation on the contents. This is particularly the case when one desires to compose a document from dynamic sources such as databases.

We have recently explored a dramatic example of a dynamic document composed of tens of thousands of other dynamic documents. This is an electronic collection of weekly bulletins from tens of thousands of Churches. The collection, `http://www.hows.net`, itself, is the collection of all weekly bulletins, created and modified as the editors of the bulletins see fit. Motivated by this specific application, we augmented XML to include conditional interpretation, and therefore the expressive power of a programming language. This augmented XML is called XFA (XML For All). The Church collection document on the web at the time of this writing does not use XFA, but a large XFA beta testing site is currently available to volunteers who edit bulletins for their churches. Dynamic pages are the rule, not the exception, in the beta site. One page of the Church bulletin document may need to fetch all churches within a mile of a selected location. This is clearly a dynamic page that requires conditional interpretation of some of its components.

With XFA, dynamic documents can be composed without the need for writing special programs that interpret specific markup. Often, perhaps all too often, those specific programs have little meaning or use outside of the particular document. Furthermore, in the spirit of the intentions behind SGML, the dynamic XFA document is completely self-contained in that all conditional interpretation reliant on web server resources or user input is made specific.

So, for example, with XFA, one can author an algebra textbook that contains exercises where a reader can actively test his knowledge of algebra. The document itself can interact with the user and tell whether the user is right or wrong. It would be hard to argue, as some XML advocates might, that these algebra exercises are not part of the document itself.

The programmable augmentation of XML that XFA provides is, itself, outside the scope of XML. While it is true that XFA subsumes XML, and any XFA document through a simple quoting filter can be parsed by an XML parser, the full specificity of the XFA document cannot be realized by a conventional XML system. This lack of interoperability should not be of concern because it would be the case in comparing any two XML systems that include more than just a syntactic parser (and perhaps style sheets). In effect, making a commitment to XFA requires a commitment to certain fixed ways to augment XML for programmability.

For this reason we have undertaken a series of studies of dynamic documents that may benefit from the augmentation provided in XFA. Certainly the instance of the dynamic church bulletin of bulletins and the instance of the algebra textbook are two such instances. We are in the process of investigating many others.

Principal among these are electronic commerce catalogs. The E-commerce catalog is particularly interesting because it is much more broadly applicable than church bulletins or textbooks. Furthermore, a useful electronic catalog is clearly a highly dynamic document even its most simplified form. A reasonable dynamic catalog must be able to take orders for the products it lists and provide powerful search methods on those products. We now turn to the experiment with an electronic commerce catalog.

The Experiment

The remainder of this paper reports in detail the experiment to clearly understand the work involved in a dynamic catalog document using XFA. We asked a 20 year old undergraduate from ITESM the Monterrey Institute of Technology) in Mexico, the second author of the paper. His task was to take a simple design for a catalog, learn XFA, and write the dynamic document that implements the catalog. This paper reports on the time and effort involved, and also explains and shows the entire dynamic document markup for the electronic catalog.

The electronic catalog is at `http://www.ecom.cmu.edu/xfa`. It allows for an exploration of computer hardware catalog items by item name, product category, and manufacturer. A person coming to the catalog can register with the catalog. Another catalog page implements a shopping basket associated with the registered user. This page also dynamically totals the costs of all the items in a shopping basket and figures out shipping costs.

While this is a fairly simple electronic catalog, it was deemed sufficiently challenging to represent a good report on augmenting XML for dynamic document markup.

Methodology

The electronic catalog document incorporates procedures written in XFA to query and update data in a database. The XFA catalog pages are interpreted by the XFA interpreter.

We have implemented a general purpose XFA interpreter that can be used with any XFA page. This implementation is a CGI script on our HTTP server, but this particular implementation is arbitrary since XFA could also be realized as a plugin in a server or browser. So as to be compatible with any existing HTML browser, the interpreter generates an HTML page that incorporates responses to user input and the data retrieved from the database. HTML Forms are read directly by form-specified XFA documents in order to allow the users to input the information for query or to enter information into the database. For compatibility with virtually all database systems, XFA provides a standard database interface using markup that queries and updates data through the widely-used ODBC protocol.

Detailed Catalog Description

The catalog allows the user to perform searches of a product in the catalog in three different ways: by the *name of the product;* by the *name of the manufacturer;* or by *category of the product.* If the user chooses to search a product by name a form is displayed in which the user can input the product name. In the other two options a list of categories or manufacturers is shown in the form of hyperlinks.

The user also has the option to *browse the whole list of products in the catalog.* In all cases the user will end up with a table that contains a list of products with price, manufacturer, quantity in stock and a link that allows the customer to add the product to his/her shopping cart. If the customer clicks on the link to *add a product to cart,* the customer's ID and password will be requested, and, if the password matches a valid customer ID, the product will be added to the customer's cart.

One more essential page lets the user *register* as a customer in the database. A form prompts the customer for name, address, telephone number, credit-card number, and password. This information is entered in the database and the customer is issued a customer ID number.

Every time the customer adds a product to his or her shopping cart, its contents are displayed. This page also includes an option to place the order. The catalog page also provides the option to browse the contents of the customer's cart from the main page.

Catalog Architecture

The architecture of the main page of the document consists on a pair of frames: the main and the menu frame. Both frames have the same option links, but the menu frame remains always present while the main frame is used to display the page requested.

The frames that compose the welcome page and the "About" page are the only static HTML files of the catalog. Everything else that is presented to the user is a rendering of dynamic XFA pages. These are expressed in HTML for transmission to the user's browser.

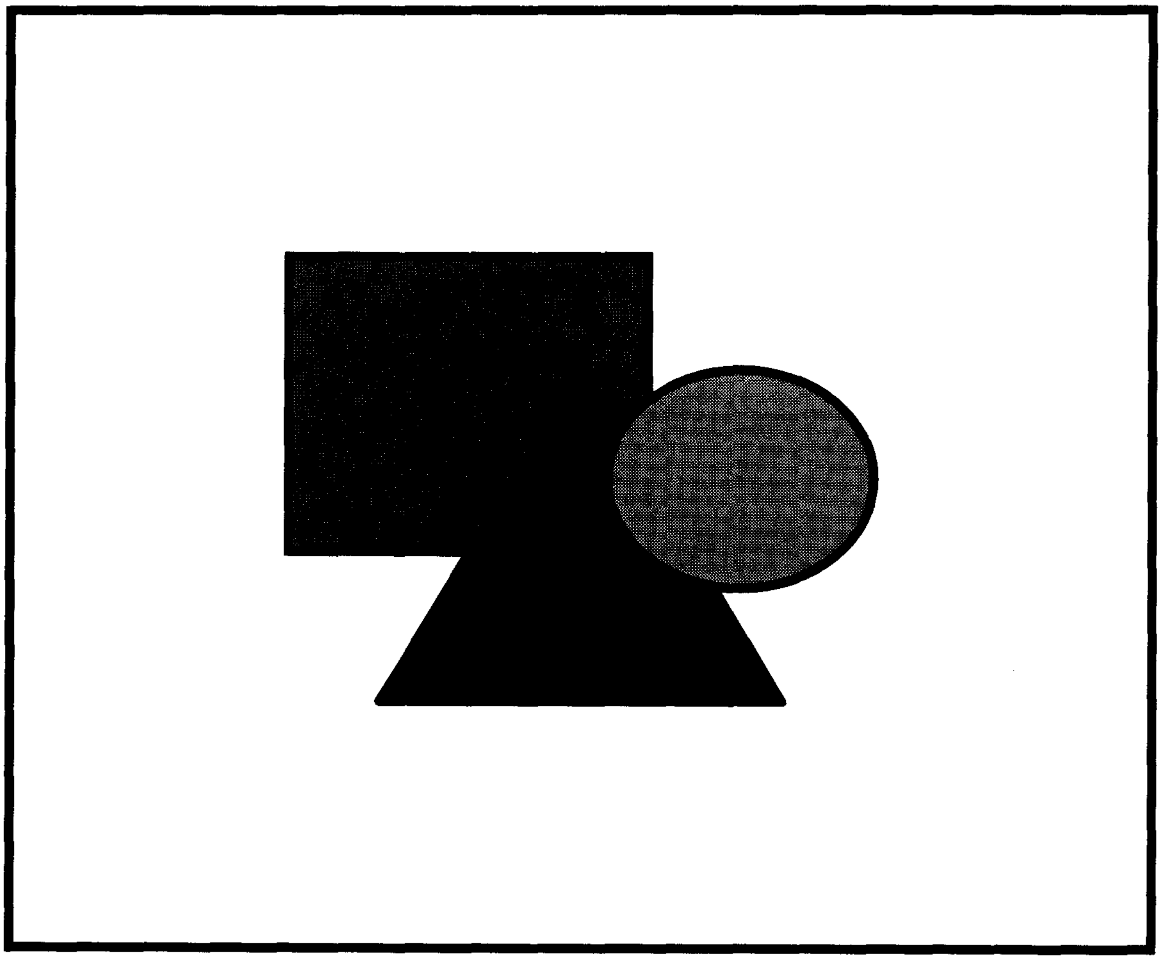


Figure 2
Look and feel of the catalog

Database

All of the product and customer information is stored in a relational database implemented using Microsoft Access. Five tables are used: *customer, product, cart details, category* and *make.* The customer table contains all the information (i.e. name, address, etc.) about the customer. The primary key, the *Customer ID,* is assigned at the time of registration.

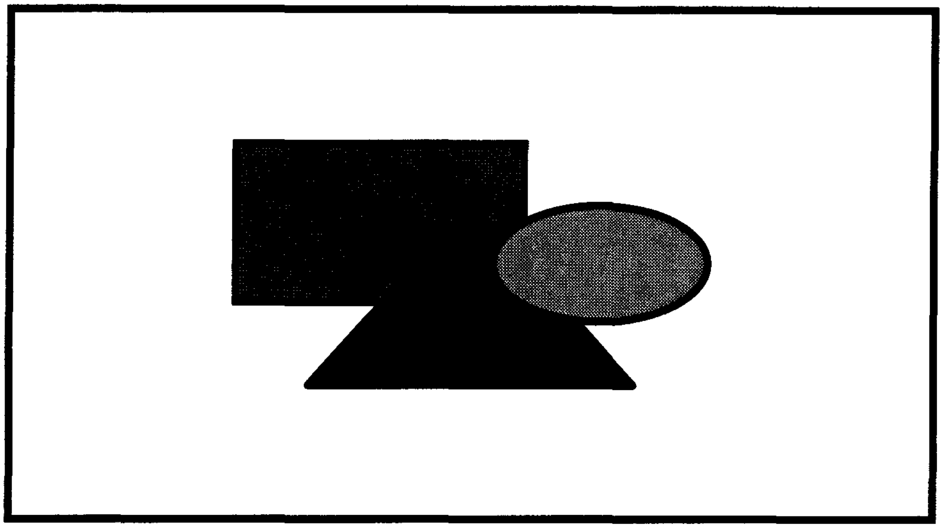


Figure 3
ERD of the database

The *product table* contains the attributes: *ID (its primary key), name, price, quantity, category* and *manufacturer name.* Both *category* and *make tables* contain two attributes: the *ID* and the *name* of the category or manufacturer accordingly. There is fifth table, which contains the details for all shopping carts. This entity contains as attributes: the *customer* and *product ID,* which are references to their corresponding tables, and the *quantity to order.*

Effort

The estimated time involved in building the electronic catalog, including the time spent in learning the markup language with no prior knowledge of XML but working knowledge of HTML, was about 60 hours.

The distribution of time spent in building the E-commerce application including time learning XFA programming, coding and debugging time, and database and catalog construction is shown in the chart in Figure 1.

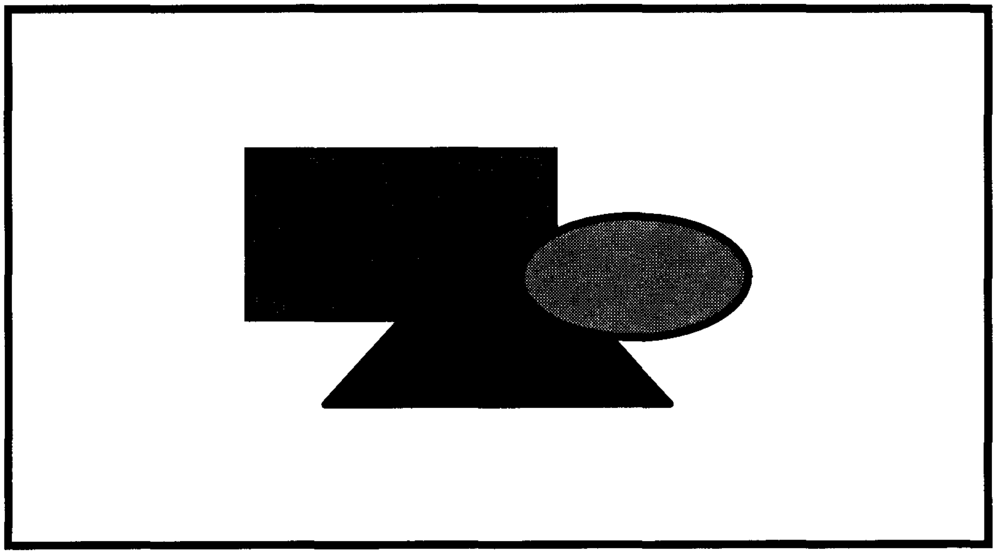


Figure 1
Distribution of time spent building the catalog.

Thirty percent of the total time (i.e. about 18 hours) was spent to becoming familiar with the language itself. That involved learning the syntax, data types, and providing methods and conditional interpretation statements (e.g. for, if, etc.) in XFA.

Coding and debugging the XFA document involved about half of the time invested in completing the demo. All coding was done using a simple text editor. The second author is of the strong opinion that coding time can be reduced with the use of a text editor that simply highlights the reserved keywords of the language.

A significant 15 percent of the time was devoted to replacing changes in the syntax of XFA. Since XFA is undergoing active development at this time, we might expect this additional time would be eliminated when the language is fully developed. On the other hand, this 15 percent is also a sign that markup does require periodic updating, if only to keep up with new features in standards.

Building the catalog and the database in Microsoft Access required around 10 percent of the overall time. The author was already familiar with Access.

XFA

The XFA markup language is processed by an interpreter which is written in C. The interpreter incorporates a nearly complete XML parser augmented for computer programmability and for certain standard I/O interfaces. These interfaces include an ODBC interface to databases such as Microsoft Access, Sybase, ADABAS, and Oracle, and an interface for handling HTML form input.

All of the XFA code that implements the E-commerce catalog is available on the catalog at `http://www.ecom.cmu.edu/xfa` in the "About" area, and, in fact, uses additional XFA markup pages to fetch and pretty print the actual code for viewing. This is not a copy of the code, it is the real code as the interpreter sees it. Again, this makes a case for strongly dynamic documents because you are not left wondering if the code you are seeing is in fact capable of producing what you see.

As a programming language, XFA supports the following data types: *strings, trees, objects* and *object sets. Strings* are further explicitly specialized if necessary as *integers, booleans, currencies, dates.*

All of XFA is expressed in the form of tags. In keeping with XML name space proposals, all XFA tags start with xfa (e.g. `<xfa:val row^attribute`). XFA procedures are known as functions and they must begin with the tag: **`<xfa:function`** `name`, and, at the time of this writing, must be saved in the .xfa file that matches the function name. More generally, a function is always defined as a path in a hierarchy of objects.

An XFA document can include any valid HTML and XML. A typical XFA document will include XFA tags mixed with HTML tags. A sample XFA file is shown in Figure 4. This example shows the syntax of a loop in XFA, delimited by the tags <xfa: for .. and </xfa: for.

Within this loop there are several XFA tags, such as <xfa: val s1^product_name which provides the value of a field in the database.

XFA provides support for HTML form entry. XFA forms work as HTML forms. For instance, the XFA interpreter supports the HTTP method *post*, which will call an XFA function when the submission button is pressed and will pass the text input in the form fields to an XFA function.

```
<xfa:function list_all
<xfa:note This macro allows to list all the products in the database/
  <HTML
    <HEAD
      <TITLE
        List of all existing products.
      </TITLE
    </HEAD
    <BODY BACKGROUND="../fondo.gif"
      <FONT FACE="Tahoma" COLOR="Navy"
        <H3
          List of products.
        </H3
      </FONT
    <HR
    <P
    <CENTER
      <TABLE BORDER=1
        <TR
          <TD<STRONG
          <TD<STRONGManufacturer
          <TD<STRONGName
          <TD<STRONGPrice
          <TD<STRONGQuantity<BR in stock
          <xfa: for s1=Sort(objects^product,"product_category_id")
        <TR
          <TD<xfa:ref input_cust_id (s1.product_id)
                <IAdd to cart!
              </xfa:ref
          <TD<xfa:val s1^product_make.make_name/
          <TD<xfa:val s1^product_name/
          <TD ALIGN=RIGHT
            <xfa:val s1^product_price/
          <TD ALIGN=RIGHT
            <xfa:val s1^product_quantity/
          </xfa: for
      </TABLE
    </BODY
  </HTML
</xfa:function
```

Figure 4
Sample of a XFA file: list_all.xfa

A relational database in XFA is referenced as `objects`; a table from a database has type `object set`; and a table row has type `object`. A database table in XFA should be referenced as `objects^table_name` and an attribute from a table is referenced as `obj^table_name^attribute_name`.

XFA provides the following database access methods: *filter, sort, new* and *delete,* which allow performing equivalent operations to the "SELECT FROM WHERE" and "UPDATE" statements in SQL.

A database configuration file is needed to map the tables and fields that are to be accessed through XFA functions. This file is shown in Figure 5.

```
<xdata:database dsn="catalog"
user="Admin" password=""
  <xdata:db_table
   XFA_name="product" SQL_name="product"
   SQL_key="product_id"
    <xdata:table_attr
     XFA_name="product_name"
     SQL_name="product_name"
     type="string" size="50"
     default="*name*"/
    <xdata:table_attr
     XFA_name="product_price"
     SQL_name="product_price"
     type="currency" default="$0.0"/
    <xdata:table_attr
     XFA_name="product_quantity"
     SQL_name="product_quantity"
     type="integer" default="0"/
    <xdata:table_attr
     XFA_name="product_category_id"
     SQL_name="product_category_id"
     type="ref" ref="category" default="1"/
    <xdata:table_attr
     XFA_name="product_make"
     SQL_name="product_make"
     type="ref" ref="make" default="1"/
  </xdata:db_table
</xdata:database
```

Figure 5
data.xml - database configuration file

Conclusions

The original intent of SGML was to provide a means for creating multipurposed digital documents. With HTML this intent was partially lost because HTML was itself only a single instantiation of purpose. The purpose was solely the viewing of mini-documents on the web that were tied together through hyperlinks and simple actions such as form requests. However, most of the intelligence behind the documents has had to be specially programmed using languages like PERL, C, Visual Basic, and JAVA, as CGI scripts and plugins. With the introduction of XML as a recommended method for reinstating the multipurposing power of SGML, we can also entertain broadening the purposes to which documents can be authored. This paper has considered broadening the purpose to the largest known possible set by the inclusion of conditional interpretation and general programmability.

To be as clear and definite as is possible, we have explored a complete E-commerce catalog using an augmented version of XML termed XFA and we have provided all the source for the XFA catalog document for viewing. This experiment cannot be judged a success or failure since we knew *a priori* that general programmability could guarantee that the XFA language could be employed to express whatever document actions were desired. Rather, this experiment constitutes a set of observations, and this paper constitutes a clear elucidation of these observations. It is for the practitioners of the art of authoring dynamic documents to judge whether conditional interpretation is to be preferred over the use of specialized CGI scripts and plugins.

The example of a textbook that contains its own interactive exercises and, perhaps, tests, would be an interesting further experiment for XFA. But there are many other, perhaps less obvious, uses. Another book that would be much better as a dynamic book would be an atlas. Plausably a complete atlas is several billion bytes, but an intelligent atlas might overcome this problem b providing its own methods for interacting with the desires of the person trying to use it. In past experiments, we developed an extreme science fiction novel called "Metafire," that contains such active experiential elements as an ultrafast web spider and a live geiger counter. Use of a language like XFA would permit a closer tie to reading the novel and physically experiencing its plot components.

We also developed a metadata language for document storage and display where the code for the interpretation of many of the metatags could be better incorporated into the document itself. For example, many figures and graphics have no general solutions for alternative display. There could be a 'conditional skip chain' that queries environmental resources for storage and presentation according to the author's own judgement.

Finally, we believe our work with color presentation of antique books would be enhanced with built-in internal search, or page hopping, associated with the book content itself. The same words meant something quite different two hundred years ago, and important topics and events were important in their day, not ours. When you are 'in' one of these books, you are in a different world, and that world demands its own unique interactive solutions to electronic presentation.

More generally, self-contained dynamic documents allow authors, publishers, and people who index documents, much greater flexibility as the individual characteristics of the individual documents become more important to accessing document content in a preservable fashion.

The introduction of the XML recommendations from the World Wide Web consortium promised the "contextualization" of document information. In effect, by employing the DTDs to type or classify tags, it is hoped that new tags could be introduced and their proper parsing known. While it is readily possible to produce a DTD for XFA, the DTD does not contextualize a document as strongly as, for example, ANSI C function declarations contextualize ANSI C functions. This is a problem for meeting the full spirit of XML, if, indeed, people find it advantageous to employ strongly dynamic documents for textbooks, collections of dynamic documents like bulletins, extreme books, old books, and for E commerce catalogs. We have to be concerned about dynamic documents of the future as well as the static documents of the past.

But, there is a deeper problem of contextualization that SGML, XML, and XFA do not address. This is contextualization to the semantic levels needed to disambiguate the intent of content across the Internet. For example, how can I know that the XFA catalog site is a demo site that illustrates the use of augmented XML in strong dynamic documents, and, not, for example, a real ecommerce site that is selling goods from Hewlett-Packard? While a person can readily detect that this is a demo site, a machine may have difficulty. A newer recommendation from the World Wide Web consortium, termed RDF, or the Resource Description Framework. RDF permits the predication of document parts by other documents and document parts. Standardizing predications would be a logical next step past RDF. RDF, like XFA, is written in XML but includes certain assumptions about conditional interpretation that XFA may make explicit. An interesting experiment would be to write RDF predications in XFA and make these sources available as dynamic predicating of new documents and document parts. For example, a document that can answer questions about itself, even though it, itself, may be dynamic. So a view of all this is a view of digital libraries of dynamic documents utilized as we use reference documents or textbooks today.

We believe that XML can serve the pattern recognition community as a jumping off place to create general well-formed representations for document representation that can be universally accepted. A major side benefit is that scanning documents to the web would then be a "given."

References

For additional reference, the following links are provided:

http://www.ecom.cmu.edu/xfa

URL for the E-commerce catalog mentioned in this paper.

http://www.xmlforall.com

Site for the extended markup language (XFA).

http://www.w3c.org/xml

XML Site from the World Wide Web Consortium.

http://www.w3c.org/rdf

RDF Site from the World Wide Web Consortium.

Glossary of terms

ANSI C (American National Standards Institute "C" Programming Language Conventions)

CGI (Common Gateway Interface)

DTD (Document Type Declarations)

ERD (Entity Relationship Diagram)

HTML (HyperText Markup Language)

HTT (HyperText Transfer Protocol)

ODBC (Open Data Base Connectivity)

RDF (Resource Description Framework)

SGML (Standard Generalized Markup Language)

SQL (Structured Query Language)

XFA (XML For All): XML augmented with Programmability.

XML (eXtended Markup Language)

Issues in Optical Character Recognition for Army Machine Translation

Jeff DeHart Christian Schlesiger
V. Melissa Holland
U.S. Army Research Laboratory (ARL)

Abstract

This paper discusses ARL's R&D program in optical character recognition (OCR). ARL's language technology research has focused in two areas, basic natural language processing (NLP) and applications of NLP to machine translation (MT). Basic NLP targets human computer interfaces in command and control systems. MT targets soldiers in the field. Their need for a simple method of determining the importance of recovered documents is a main concern. However, we lack robust, accurate OCR to provide input to MT. ARL's strategy for improving OCR is to leverage industry while at the same time providing incentive to build algorithms that meet Army needs. Simultaneously, we participate in a multi-agency program to evaluate available OCR on comprehensive test corpora.

1 Why We're Interested in OCR

The Army Research Laboratory began work in language technologies a decade ago, developing natural language processing (NLP) tools to allow easy interaction with systems like the Combat Information Processor (CIP), with its maps and complex simulation functions. More recently, we turned to NLP applications like machine translation to address the need of soldiers in the field to assess information in languages they do not understand. We expected the focus of our funding and our in-house research to be on machine translation and on translingual information retrieval and extraction—which we know to be hard problems. We have found in the course of developing and trying prototypes in the field that the real bottleneck in processing foreign language documents is OCR. That is, we have not yet obtained OCR good enough to even show the benefits of our work in MT for most kinds of documents encountered in the field.

2 The Need for Automatic Translation in the Field

Military personnel placed in forward positions during peacekeeping and other operations typically encounter printed documents in a language they cannot understand. When vast numbers of these documents are found on a mission, they are all sent back to the linguists and analysts supporting the mission from afar. For example, during the Gulf War mountains of Arabic documents captured in forward areas went back to linguists for review—too many (to this day) for all of them to be even briefly assessed. The non-linguists on the front could not be expected to cull critical documents from the rest. This situation is typical of military operations, whether combat or non-combat.

This situation means that a valuable military resource, i.e., literate and informed troops, goes underutilized. More important, since the value of information decreases rapidly with time, especially in a battlefield environment, critical information may be missed that could save lives and shorten conflict. While fully automated translation may be in the far future, just providing troops on the ground with the capability to access the worth of recovered documents is potentially life saving. Moreover, for the linguist, who can be inundated in actual operations, an automated filtering process could greatly increase the probability of finding critical information in time.

3 The FALCON Portable Translator

Motivated by these factors, the Army developed the prototype portable translator called FALCON (Forward Area Language CONverter), Figure 1, which ARL took over in 1996. FALCON was designed to bring non-linguists into the loop of the analysis process by giving them a portable document analysis support tool. With FALCON they can scan a printed page, recognize individual characters within the scanned image, and produce a rough English translation to evaluate with key word searches. They can then identify captured documents that match a profile of keywords defined by analysts for the mission. Documents that pass this relevance filter can be transmitted electronically, along with the translation produced, for further processing by linguists. Through alternative military secure and nonsecure networks (including MSE and SINCGARS), an enhanced link now exists between information collectors in the field and translators or analysts at successively higher echelons.

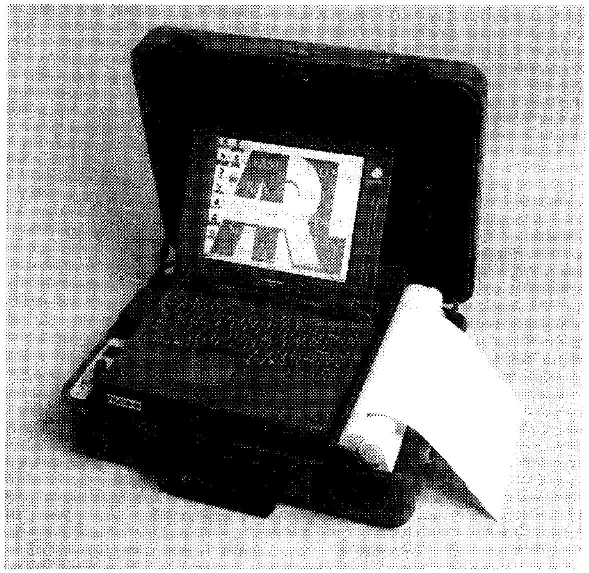


Figure 1: FALCON portable translator.

FALCON combines a laptop computer, scanner, and alternative power sources with software for optical character recognition, text-to-text translation, and tailorable key word search. These components are protected in a padded, rugged metal case. Case and components weigh about 30 lbs in the current version of FALCON, and 20 lbs in a new version being readied by ARL engineers.

Documents are processed by FALCON in three phases, schematized in Figure 2: (a) *scanning* generates a bitmap image of pages fed into the scanner, (b) *optical character recognition* recognizes characters in the bitmap, and (c) *translation* converts the OCR'd source text to English text and supports keyword searches on source or target text files.

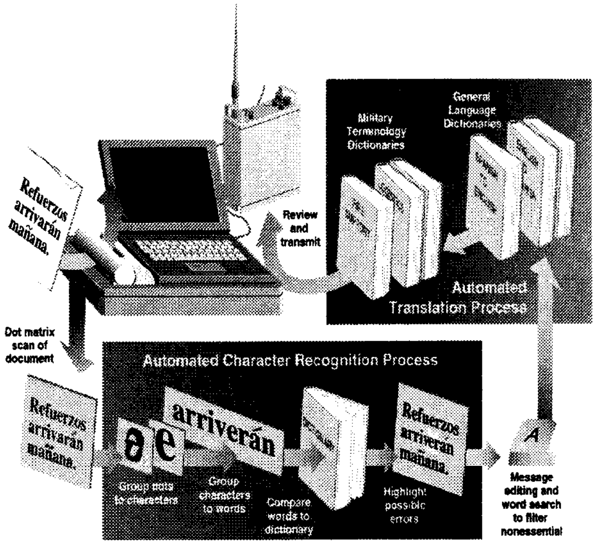

Refuerzos arrivarán mañana.
Review and transmit
Military Terminology Dictionaries
General Language Dictionaries
Automated Translation Process
Dot matrix scan of document
Automated Character Recognition Process
Refuerzos arrivarán mañana.
arriverán
Group dots to characters
Group characters to words
Compare words to dictionary
Refuerzos arriverán mañana.
Highlight possible errors
A
Message editing and word search to filter nonessential

Figure 2: OCR to translation in FALCON.

The aim of ARL has been to use off-the-shelf packages for scanning and OCR and to integrate a range of translation software, from commercial products for common languages to experimental or prototype software for rare languages. OCR and MT packages have been selected and refined in collaboration with the Air Force and the intelligence community. The primary translation software comes from a program at the National Air Intelligence Center (NAIC) and is developed for government and commercial use by SYSTRAN Inc. We have also ported software from experimental programs, such as Carnegie Mellon University's example-based machine translation project, which features software for translating between Haitian Creole and English. Whatever the source of the MT, the translation into English at best approximates the meaning of the original text and requires substantial post-editing for full comprehensibility and fluency.

Software to translate from Serbian and Croatian into English was developed by SYSTRAN specifically for use in Bosnia and has a supporting lexicon of nearly 100,000 words. It has been successively improved over the past two years and has reached a state of development characterized by NAIC as "prototype" (still beneath the pre-production and production levels of its stronger language pairs). The OCR to process the Cyrillic and latinic fonts used in these languages was adapted from Russian with U.S. government funding to Cognitive Technology Corp.

FALCON prototypes have been tried in Bosnia since May of 1997 by selected units of the U.S. Army's V Corps intelligence troops and Special Operations Forces to evaluate documents written in Serbian or Croatian. FALCON also participated in a January 1998 Joint Field Exercise at Ft Bragg, NC, as a tool for evaluating documents in multiple languages and communicating the files over secure networks to military intelligence analysts operating behind the front. FALCON was adapted for Haitian Creole and sent to Haiti in summer, 1998, for trial use in peacekeeping operations.

4 Lessons Learned

After cycles of field trials, FALCON has yielded formative feedback. The Bosnian experience is instructive. Non-linguists who used the system there found that, when the source text was accurately represented (e.g., an electronic document), the MT was indeed sufficient for document screening. They found in one trial that it saved processing time for an acquired document and reduced the human work needed to get a quick read on the document. The reported ratio of time spent translating in this trial was 15 : 9 : 2 for an army linguist : native speaker : FALCON. The translation, in users' estimation, achieved "an 80% solution" much of the time, which was felt to be good enough for evaluating a document's intelligence value.

However, for documents that had to be scanned and OCR'd, the reported utility of FALCON was much lower. The OCR package was not capable of achieving minimally useful accuracy on many documents found in the field: faxes, mechanically typed pages, multi-generation copies, or dirty, smeared paper. This limitation is typical of commercial OCR, which is geared toward small business and office uses, focusing on laser-printed rather than low-quality documents. FALCON users reported that it was faster to type documents in by hand than to OCR and correct errors.

5 OCR Program at ARL

The lessons from Bosnia and elsewhere stimulated us to invest in OCR development and evaluation. This investment, unlike our work in MT and NLP, is largely external for two reasons. First, there is a pre-existing academic research community working in this area with deep experience in document understanding issues (including research programs in the intelligence community). Second, there is a growing industrial base in multilingual OCR technology. The market for multilingual OCR is expected to increase greatly in the near future due to the globalization of information, the growth of the world wide web, and due to ubiquitous penetration of scanners, computers, digital cameras etc. into the home and business environments around the world. We therefore expect that our need for multilingual OCR will be met by concentrated, cooperative efforts with industry and academia, together with other government agencies.

Yet there are two areas in which we need OCR improvement which are not likely to be tackled by commercial developers without some government incentive. First is better OCR performance on degraded documents representative of those typically found in the field. We expect that army missions will continue to yield low-quality documents, including open publications from developing or third-world presses with poor paper and print, to multiply faxed reports, to crumpled or torn personal papers. Second, we need focus on languages not yet of high enough commercial value to foster competitive commercial development. These languages often present special problems for OCR. For example, Arabic and Farsi are printed in script, with ligatures between characters that impede character separation during the recognition process.

These issues are not unique to the army but have long concerned government labs in the intelligence community, NIST, DOE, and at Los Alamos. As a relative newcomer in this field, we have tried to take advantage of previous and ongoing work in these agencies wherever we can.

To focus this three-pronged approach—leveraging commercial progress, collaborating with other agencies, and stimulating army-relevant commercial development—we have applied a dual use strategy. That is, we have joined with other agencies and with industry to seek funds from the Dual Use Science & Technology (DUST) program, which applies to technologies in which there is ongoing commercial investment but in which commercial development either is not adequate or is not fast enough for the military. DUST provides two incentives to private industry. First is the use of new, commercial-like contracting methods known as Other Transactions; traditional government contracting requirements place so much overhead on a company that market share becomes an issue. Second, the government willingly provides half of the cost of a research project (matching industry) without asking for intellectual property rights. In return, industry focuses on government-unique problems, which otherwise might not be addressed, and government users have the advantage of technology that keeps pace with products as they are upgraded. So far, ARL has used DUST funds for development of enhanced OCR for Arabic and Farsi and is beginning DUST projects for developing improved machine translation in Arabic, Farsi, Chinese, and Korean.

6 OCR Evaluation

The state of the art in computer vision is advancing. Document understanding and its OCR component will surely take advantage of this trend. Thus, it is in the army's interest to use the highest performance OCR it can. Because we see that the future of this market will be driven by commercial industry and that upgrades to products arrive on a frequent basis, we desire evaluation techniques and test suites that are relevant to the military. The test corpus should be typical of what is found in the field. We face the same problems encountered with all benchmarking efforts: (a) developing and updating a corpus of relevant ground-truthed test data, (b) playing off the integrity of the test suite against the advantages of providing developers with typical input data and information on where they are fall short, (c) determining the best criteria for evaluation, and (d) making it as inexpensive and accessible as possible to the user.

To help us with these issues, ARL is working with the LAMP Lab at the University of Maryland to test and compare commercial products for Arabic and Farsi OCR. Development of an army representative data set has proved to be difficult, and results showing degradation in performance in newer software versions is confusing [1]. Segmentation of the data set by document type may help clarify the results. Ongoing research in the synthetic degradation of document images is also of interest to army users [2].

7 Future Research Focus

ARL has interest in several research areas for OCR: improving performance on degraded documents, developing reliable algorithms for additional languages from the Middle East and Asia, automatically identifying script and language, and recognizing document state or quality prior to OCR. The discovery of an OCR algorithm that has accurate, consistent multilingual capability is the "Golden Chalice" of the community. However, research into other "object recognition" technologies indicates that this may be far in the future. The overall long-term goal is a system that could acquire a document, recognize the quality and source language, and apply appropriate preprocessing and recognition algorithms. The target platform is a wearable computer that minimally impacts the user's mobility and provides OCR, translation, and translingual information extraction. Replacement of the scanner by a digital camera would increase flexibility of what can be OCRed as well as reducing size and weight of what the soldier has to carry.

References

[1] T. Kanungo, G.A. Marton, and O. Bulbul, OmniPage vs. Sakhr: Paired model evaluation of two Arabic OCR products, *Proc. of SPIE Conference on Document Recognition and Retrieval (VI),* 3651 (San Jose, CA, 1999).

[2] T. Kanungo, R.M. Haralick, and I. Phillips, Nonlinear local and global document degradation models, *Int'l. Journal of Imaging Systems and Technology* 5 (1994) 220–233.

Automating data entry into MEDLINE

George R. Thoma
National Library of Medicine
Bethesda, Maryland 20894
thoma@nlm.nih.gov

Abstract

Data entry for the thousands of bibliographic databases around the world from information in journal articles continues to be heavily manual. At the National Library of Medicine (NLM) we are automating the production of bibliographic records for MEDLINE^{R}, NLM's premier database used by clinicians and researchers worldwide. As a first step, the Lister Hill National Center for Biomedical Communications, an R&D division of the library, has developed a system called MARS (for *Medical Article Record System*) that involves scanning and converting (by OCR) the abstracts that appear in journal articles, while keyboarding the remaining fields (e.g., article title, authors, affiliations, etc). We focus on the abstract first because this is the largest field in a typical record, amounting to a maximum of 4000 characters. While this system is in production, we are designing a second generation system to automatically extract these other fields as well. This future system will employ scanning and OCR as well, in addition to modules that automatically zone the scanned pages, identify the zones as particular fields, and reformat the field syntax to adhere to MEDLINE conventions. This talk describes the first generation system currently used for production, and the ongoing work toward the design of the second generation system.

The initial system consists of multiple workstations of three types: scanners, workstations for manual entry (keyboarding) and workstations for reconciling (proofing and correcting), in addition to three unattended servers: a network file server, an OCR server and one to perform various file matches.

At each scan workstation, the operator first barcode-scans an ID number that appears uniquely on each issue. The operator then scans the first page of each article on which the abstract appears, and manually zones the title and abstract whose bitmapped TIFF files are sent to the network server.The OCR server retrieves these TIFF files from the network server, and produces text files of the abstract and title. The network server maintains directories in which the scanned TIFF images, the abstract text files and the citation files are all kept until they are acted upon. The barcoded ID number scanned initially serves as a directory name and all TIFF images and OCR data for all the articles in that issue are linked to that number.

Concurrently or at any time, the keyboarder keys in the fields (other than the abstract) for each article, and a second operator repeats this process for the same articles. Double keying is found to substantially improve accuracy, thereby reducing the burden on the reconcile operators. The two manual entries are compared automatically to produce a "citation difference" file highlighting inconsistencies. Then the title field from this citation difference file is automatically matched with the OCR'ed title, thereby linking the keyed data with the scanned abstract.

Meanwhile, the abstract text from the OCR is checked by a spellcheck module based on medical lexicons and heuristic rules to reduce the number of correct words that were highlighted, to reduce the burden on the reconcile operator. At this point, all the fields entered by keyboard and out of the OCR system are available for validation and proofing by the reconcile operators. Following this step, the completed record file is FTP'ed to the NLM mainframe computer, and later accessed by indexers who add appropriate descriptive information such as Medical Subject Headings, thereby completing the bibliographic record to be added to MEDLINE.

The ongoing work in developing the second generation system consists of developing algorithms to detect page zones (page segmentation), automatically label these zones by field name (article title, author, affiliation, abstract), and then automatically reformat the zone text syntax. The system relies on a database to keep track of the workflow as well as serve as a repository for data extracted from the scanned page to be used by subsequent processes.

Automated Labeling of Zones from Scanned Documents

Daniel X. Le, Jongwoo Kim, Glenn Pearson, and George R. Thom

National Library of Medicine
Bethesda, Maryland 20894

Abstract

The Lister Hill National Center for Biomedical Communications, a research and development division of the National Library of Medicine (NLM), is developing an automated system, the Medical Article Record System (MARS), to identify and convert bibliographic information from printed biomedical journals to electronic format for inclusion in the MEDLINE database. This paper describes one aspect of this ongoing effort: the automated labeling of zones from scanned images with labels such as titles, authors, affiliations, and abstracts. This labeling is based on features calculated from optical character recognition (OCR) output, neural network models, machine learning methods, and a set of rules that is derived from an analysis of the page layout for each journal and from generic typesetting knowledge for English text.

Several learning systems are considered including back-propagation neural networks, decision trees, and rule-based systems. Experiments are carried out on a variety of medical journals, and the performance of these techniques are analyzed and compared in terms of development times, training times, and classification accuracy.

1 Introduction and Background

Automated document conversion systems are being developed for a variety of document related applications to convert paper-based document information to electronic format. Paper documents usually consist of text zones, or a mixture of text and non-text zones, and each text zone has its own label such as titles, authors, affiliations, abstracts, etc. In order to support automated document searching, automated document delivery, and electronic publishing (converting papers from one format to another or modifying manuals and references, etc.), document labeling techniques are then required to extract the meanings of text zone contents.

Most document labeling techniques proposed so far in the literature [1, 2, 3, and 4] are based on the layout (geometric) structure and/or the logical structure of a document. Hones et al. [1] described an algorithm for layout extraction of mixed-mode documents. Taylor et al. [2] described a prototype system using 'feature extraction and model-based' approach. Tsujimoto et al. [3] presented a technique based on the transformation from a geometric structure to a logical structure. Tateisi et al. [4] proposed a method based on stochastic syntactic analysis to extract the logical structure of a printed document. Other techniques [5, 6] used the outputs of OCR to further improve labeling accuracy. In this paper, we propose an automated technique to label text zones with labels such as titles, authors, affiliations, and abstracts using integrated image and OCR processing, rule-based technology and back-propagation neural network. Preliminary evaluation results show that the system is capable of labeling text zones at a classification accuracy of 99.6% for the rule-based system and of 97.0% for the back-propagation neural network.

The rest of this paper is divided into six sections. Section 2 provides a system overview. Section 3 presents zone features. Sections 4, 5 and 6 describe in detail the labeling techniques and experimental results. Section 7 contains conclusions and future work.

2 System Overview

The automated labeling technique described here is one prototype component of our second-generation MARS system under development. The process consists of three steps as follows:

- Scan journal images.
- Perform optical character recognition (OCR). This includes detecting zones around paragraphs.
- Apply automated labeling. This associates a label, such as "Title", with each zone of interest.

Additionally, since verification and correction steps are needed to collect ground truth data for training, a Zone Checker system was also implemented to serve this purpose and the interested reader might refer to reference [7] for more information about this system.

Using a commercial 5-engine OCR system developed by Prime Recognition Inc. (PR) [8], scanned binary document images are first segmented into rectangular text zones. Each zone is then processed to deliver an OCR output (including zone coordinates, text line information, characters and their bounding boxes, confidence levels, font sizes, and certain style attributes). From this output, features for each zone are calculated and input to several learning systems for label classification. Finally, it is planned that the label classification outcomes from these learning systems for each zone are combined by voting to reach the final decision on the zone's label.

The calculated features include geometric ones, such as the zone's height/width ratio, the zone area or its position in a page, as well as those based on character statistics or substring recognition against word lists. These features are extracted from the output of the PR OCR system that provides information at the page, zone, line, and character levels, as given below:

Zone Level
Zone boundaries
Number of text lines
Line Level
Number of characters
Baseline
Average character height
Average font size
Character Level
Recognized 8-bit character
Confidence level (*1= lowest, 9 = highest*)
Bounding box
Font size
Font attributes (*normal, bold, underlined, italics, superscript, subscript, and fixed pitch*)

3 Zone Features for Document Labeling

Most features and rules derived for labeling techniques in this paper are based on an analysis of the page layout for each journal, and generic typesetting knowledge for English text [9]. Both geometric and non-geometric features are considered here.

The geometric layout features are calculated based on the zone location, zone order, and zone dimensions. Title zone is usually located in the top half of the first page of an article with the biggest font size. As reported in a title page study [10], roughly 96% of titles have the largest font compared to other zones in the top half of the first page. Normally, title is followed by author, affiliation, and other publisher information. The font size of the author zone is usually smaller that that of the title zone. The non-geometric features derive from the zone contents, and can involve aggregate statistics, font characteristics such as total characters, total capital letters, total punctuation marks, etc. Table 1 shows a list of features used both in the rule-based system and in the back-propagation neural network.

For the rules-based system, approximately 50 features are extracted from the PR OCR output. In addition to the features shown in column 2 of Table 1, extensive word matching based on cue words is used as shown in Table 2. Word matching relies upon lists of cue words commonly associated with particular label types.

Word matching is very important for the system since a zone has a higher probability of being labeled as "Affiliation" zone when a zone has many country, city, and school names. Seven database tables with word lists have been assembled and the Ternary Search Tree algorithm [11] is used as a search engine for the word matching shown in Table 2.

4 Labeling using Rule-Based System

NLM's MEDLINE database contains bibliographic records from about 3800 journals. Their physical layouts can be categorized into several hundred types. Figures 1(a), 1(b), and 1(c) show some examples of types consisting of one column, a combination of one and two columns, and two columns, respectively. We define Figure 1(a) as Type 1, Figure 1(b) as Type 12, and Figure 1(c) as Type 2, respectively. It is very difficult to design a single automatic labeling (AL) module that can handle all types of journals. Therefore, we classify journals as belonging to specific types, and design an AL module for each particular type. Since Type 12 occurs most frequently in our journal collection, we will make an AL module for it first and will handle other types in the future.

For our purpose, we are interested in five zone labels in an article: title, author, affiliation in upper portion of a page (upper affiliation), affiliation in lower portion (lower affiliation), and abstract. The remaining zones are labeled as "others". Four kinds of rules, called rules 1, 2, 3, and 4 are developed for each label type. Rules 1, 2 and 3 are different for each label classification, while rule 4 is the same for all. The proposed AL technique consists of four steps as shown in Figure 2 and described in the following paragraphs.

In the first step, a zone is labeled by rule 1. For example, when a zone has a higher Probability of Correct Identification (PID) for title (PID >= 100), the zone is labeled as title.

In the second step, previous labeling results are checked again by rule 4. For example, when two separate zones are both labeled as author but they are not close to each other, one zone is then removed fro the author category.

In the third step, in addition to rule 2, rules 1 and 4 also are applied again to make sure that at least one zone is labeled as title, author, abstract, and upper affiliation or lower affiliation. For example, when a zone labeled as author does not have any information about author (Number_Middlename = 0 and Number_Degree = 0), geometric features are then used to do the labeling. That is, if a zone does not have an information about title and upper affiliation and it is located between title and upper affiliation, the zone is labeled as author.

In the fourth step, the PR segmentation problem of splitting a zone (such as title zone) into multiple zones (multiple title zones) is handled by all rules and any remaining unlabeled zones are labeled in this final step. The detailed rules for each label type are shown in the following:

Let Max_Font_Size represent the biggest font size in a page and Height_Article be the difference between the bottom and top coordinates of the bottom-most and top-most zones, respectively.

4.1 Rules for Title

Rule 1

```
1. Sentence_Headtitle == 0
2. Font_Size == Max_Font_Size
3. Number_Degree < 3 or Percent_Degree < 10
4. Number_Middlename < 3 or
   Percent_Middlename < 10
5. Coordinate_Upper < Height_Article /3
6. Coordinate_Lower < Height_Article /2
7. If all of above conditions are satisfied {
       If ( Font_Size == Max_Font_Size )
           PID = 100
       Else If ( | Font_Size - Max_Font_Size | < 3 )
           PID = 99
       Else
           PID = (Font_Size - Min_Font_Size) ×
                 100/(Max_Font_Size – Min_Font_Size)
   }
   Else {
       PID = 0
   }
```

Rule 2

If(PID < 100) pick a zone having the highest PID for title.

Rule 3

1. Distance from a zone to title is smaller than that of any other labels.
2. Font_Size, Font_Attribute, Avg_Line_Height, and Avg_Line_Space of a zone must be similar to those of title zone.

Rule 4

```
Coordinate_Upper of title
  < Coordinate_Upper of author
  < Coordinate_Upper of upper affiliation
  < Coordinate_Upper of abstract
  < Coordinate_Upper of lower affiliation
```

4.2 Rules for Author

Rule 1

```
1. Coordinate_ Upper < Height_Article /2
2. Font_Size < Font_Size of title
3. Number_Word >= 2
4. Number_Affiliation <= 3 or
   Percent_Affiliation <= 30
5. Sentence_Headtitle == Sentence_Abstract == 0
   Sentence_Introduction == 0
6. If all of above conditions are satisfied {
       If ( Percent_Degree+Percent_Middlename > 28 )
           PID = 100;
       Else
           PID= (Percent_Degree+
                     Percent_Middlename) × 100/28
       If ( Percent_Capitalcharacter > 50 ) {
           If (PID > 50)
               PID = 100
           Else
               PID = PID + PID /2
       }
   }
   Else {
       PID = 0
   }
```

Rule 2

If (PID < 100) pick a zone having the highest PID for author.

Rule 3

1. Distance from a zone to Author zone is smaller than any other label zones.
2. Font_Size, Font_Attribute, Avg_Line_Height, a Avg_Line_Space of a zone must be similar to those of author zone.

Rule 4

Same as rule 4 for title described in section 4.1.

4.3 Rules for Upper Affiliation

Rule 1

```
1. Upper_Coordinate  < Height_Article /2
2. Lower_Coordinate < Height_Article×3/4
3. Number_Word >= 2
4. Number_Degree < 3 or Percent_Degree < 30
5. Number_Middlename < 3 or
   Percent_Middlename < 30
6. Percent_Capitalcharacter < 50
7. Sentence_Headtitle == Sentence_Abstract == 0
   Sentence_Introduction==0
8. If all of above conditions are satisfied {
       If ( Number_Affiliation >= 2 ) {
           If (Percent_Affiliation >= 30 )
               PID =100
           Else
               PID = Percent_Affiliation×100/30 }
```

```
        Else {
            If (Percent_Affiliation >= 30 )
                PID =50;
            Else
                PID = Percent_Affiliation×50/30
        }
    }
    Else {
        PID = 0
    }
```

Rule 2
If (PID < 100), pick a zone having the highest PID for upper affiliation.

Rule 3

1. If (PID > 25 and the next zone has Sentence_Received ==1) PID = 100.
2. Distance from a zone to upper affiliation zone is smaller than any other label zones.
3. Font_Size, Font_Attribute, Avg_Line_Height, and Avg_Line_Space of a zone must be similar to upper affiliation zone.

Rule 4
Same as rule 4 for title described in section 4.1.

4.4 Rules for Lower Affiliation

Rule 1

1. Upper_Coordinate > Height_Article /2
2. Lower_Coordinate > Height_Article ×3/4
3. Number_Words >= 2
4. Number_Degree < 3 or Percent_Degree <= 25
5. Number_Middlename < 3 or Percent_Middlename <= 25
6. Percent_Capitalcharacter < 50
7. Sentence_Headtitle == Sentence_Abstract == 0 Sentence_Introduction == 0

```
8. If all of above conditions are satisfied {
        If ( Number_Affiliation > 2 ) {
            If( Percent_Affiliation >= 30 )
                PID =100
            Else
                PID = Percent_Affiliation×100/30
        }
        Else {
            If( Percent_Affiliation >= 30 )
                PID =50
            Else
                PID = Percent_Affiliation×50/30
        }
        If (Sentence_Affiliation > 0) PID=PID+50
    }
    Else {
        PID = 0
    }
```

Rule 2
If(PID < 100), pick a zone which has the highest PID for lower affiliation.

Rule 3

1. Distance from a zone to lower affiliation zone is smaller than any other label zones.
2. Font_Size, Font_Attribute, Avg_Line_Height, a Avg_Line_Space of a zone must be similar to lower affiliation zone.

Rule 4
Same as rule 4 for title described in section 4.1.

4.5 Rules for Abstract

Rule 1

1. Zone is bigger than title, author, upper affiliation, and lower affiliation zones.

```
2. If all of above conditions are satisfied {
        If (Previous Zone has Sentence_Abstract == 1)
            PID = 100
        If (Previous Zone has Sentence_Received == 1)
            PID = 100
        If (Next Zone has Sentence_Introduction == 1)
            PID = 100
        If (Next Zone has Sentence_Keyword == 1)
            PID = 100
    }
    Else {
        PID = 0
    }
```

Rule 2
None

Rule 3

1. Distance from a zone to abstract zone is smaller than any other label zones.
2. Font_Size, Font_Attribute, Avg_Line_Height, Avg_Line_Length, and Avg_Line_Space of a zone must be similar to those of abstract zone.

Rule 4
Same as rule 4 for title described in section 4.1.

5 Labeling using Neural Network System

Before a neural network model can be used as a pattern classifier, its structure has to be designed and trained. We discuss in this section the selection of training and testing data sets, a method to train and test the neural network, and the neural network structure design.

5.1 Training and Testing Data Sets

Since each journal has its own page layout and style setting, our preliminary approach is to create a neural network for each journal type. A neural network for each particular journal type is designed, trained, and tested with its own data. For each journal type, a group of at least four journal issues is selected to create the training and data sets. The training data set is used to design the neural network while the testing data set is used to estimate the classification accuracy.

Sixteen different journal types consisting of 66 issues were selected for the experiment for a total of 2176 binary images. These images are 8.5 x 11 inches and scanned at 300 dpi resolution.

5.2 Cross-Validation Method

For purposes of generalization, the cross-validation (CV) technique [12] is used by randomly dividing the training data set into five data groups of which four data groups create *a CV-train set* and one remaining data group is considered as *a CV-test set.* As a result, there are five pairs of a CV-train set and a CV-test set that are used to train and test the back-propagation neural network. The modified weights corresponding to the winning pair of a CV-train set and a CV-test set, the one yielding the highest classification accuracy, are chosen to be the final weights for the neural network.

5.3 Back-Propagation Neural Network

Back-propagation (BP) [12, 13, and 14] is a multi-layer neural network using sigmoidal activation functions. The network is made up of an input layer, hidden layers, and an output layer and nodes in each layer are fully connected to those in the layers above and below. Each connection is associated with a synaptic weight. The BP network is trained by *supervised learning*, using a gradient descent method, which is based on the least squared error between the desired and the actual response of the network.

In this project, a two-layer BP network is implemented with an input layer – thirty-eight text zone features shown in Table 1, a five output layer (title, author, affiliation, abstract, and others), and one single hidden layer of which the number of nodes is 16. Therefore, the two-layer BP network architecture is 38-16-5. Each input vector of the training data set is presented to the network many times and the weights are adjusted on each presentation to improve the network's performance until the network stops improving. Two learning factors that significantl affect convergence speed, as well as accomplish avoiding local minima, are the learning rate and the momentum. The learning rate determines the portion of weight needed to be adjusted. Even though a small learning rate guarantees a true gradient descent [14], it slows down the network convergence process. The momentum determines the fraction of the previous weight adjustment that will be added to the current weight adjustment. It accelerates the network convergence process. During the training process, the learning rate was adjusted to bring the network out of either its local minima (where the network has converged but its output error is still large) or its no-gain mode (the network mode in which its output error does not change or changes very little over ma cycles). The learning rate ranges from 0.001 to 0.1, and the momentum is 0.6.

6. Experimental Results

6.1 The Rule-Based System

There were 90 rules generated for the Type 12 and 38 journals consisting of 1407 articles were selected for experiment. Experimental results showed that 1402 articles were labeled correctly with 99.6% of correct recognition rate. We had five errors in labeling affiliation zones due to the incorrect font attributes and poor contents obtained from the output of the PR OCR system.

6.2 The Neural Network System

The BP neural network was trained with all five pairs of a CV-train set and a CV-test set. The average training time spent for the each pair was about 4 hours. The BP neural network configuration associated with the winning pair was evaluated on the testing data set. The result showed that the average classification accuracy on the testing data set was about 97.0 %. Most errors were due to the segmentation problem generated from the PR OCR output that split zone of interest (such as title zone) into multiple zones, as well as merged several different zones (such as author and affiliation zones) into a single zone.

7 Summary and Conclusion

Two automated labeling techniques, a rule-based system and a back-propagation neural network, have been presented in this paper. Both techniques yielded very good performance and showed the possibility of extension to other journals as prototypes.

The rule-based labeling technique uses 90 rules for the journal layouts designated "Type 12" journals. More rules are expected to be added to handle other types of journals. This labeling technique employed both geometric and non-geometric zone features as well as geometric relations between zones as the basis for the proposed set of rules. Other rules can be changed or added easily since there is no training procedure. However, label classification time is proportional to the number of rules. In addition, much time and effort are needed to devise rules that can be used for more than one type of journal.

In the case of the back-propagation neural network technique, label classification time is very fast and the results are stable regardless of the journal types. However, it is hard to use the geometric relations between labels as features and it is time consuming to train the module and tune its learning parameters. It is also hard to analyze wrong labeling results. The most serious drawback is that the whole neural network must be trained again when we have new types of journals to label.

Since each system has advantages and disadvantages, our future approach is to combine these systems together with a voting procedure to improve the labeling results. Another AL module using a decision tree algorithm shall be implemented and the results of these three AL modules will be voted on to improve the accuracy of label classification.

References

[1] F. Hones and J. Lichter, Layout Extraction of Mixed Mode Documents, *Machine Vision and Applications* 7, pp. 237-246, 1994.

[2] S. Taylor, R. Fritzson, and J. Pastor, Extraction of Data from Preprinted Forms, *Machine Vision and Applications* 5, pp. 211-222, 1992.

[3] S. Tsujimoto and H. Asada, Major Components o a Complete Text Reading System, *Proc. IEEE*, Vol. 80, No. 7, pp. 1133-1149, 1992.

[4] Y. Tateisi and N. Itoh, Using Stochastic Syntactic Analysis for Extracting a Logical Structure from a Document Image, *Proc. IEEE Int. Conf. Neural Networks*, Vol. 2, pp. 391-394, 1994.

[5] T. Hu et. al., A Prototype for Extracting Logical Elements from Tables of Contents of Journals, *Int. Assoc. Patt. Recog. Workshop on Doc. Analysis System*, Malvern, PA, 1996

[6] J. Liang et. al., The Prototype of a Complete Document Image Understanding System, *Int. Assoc. Patt. Recog. Workshop on Document Analysis System*, Malvern, PA, 1996.

[7] G. Pearson and G. Thoma, Manual Verification and Correction of Automatically Labeled Zones: User Interface Considerations, *Proc. SDUIT '99*, Annapolis, MD, April 1999.

[8] Prime Recognition Inc., Prime OCR Access Kit Guide, version 2.70, San Carlos, CA, 1997.

[9] G. Nagy, At the Frontiers of OCR, *Proc. IEEE*, Vol. 80, No. 7, pp. 1093-1100, 1992.

[10] J. H. Ling, The Title Page as The Source of Information for Bibliographic Description: An Analysis of its Visual and Linguistic Characteristics, University Texas at Austin, 1987.

[11] J. Bentley and B. Sedgewick, Ternary Search Trees, *Dr. Dobb's Journal*, pp. 20-25, April 1998.

[12] D. R. Hush and B. G. Horne, Progress in Supervised Neural Networks - What's New Since Lippmann? *IEEE Signal Processing Magazine* : pp. 8-39, 1993.

[13] R. P. Lippmann, An Introduction to Computing with Neural Nets, *IEEE Acoustics, Speech and Signal Processing Magazine* 4(2), pp. 4-22, 1987.

[14] J. M. Zurada, Introduction to Artificial Neural Systems, *West Publishing Company*, St. Paul, Minnesota, 1992.

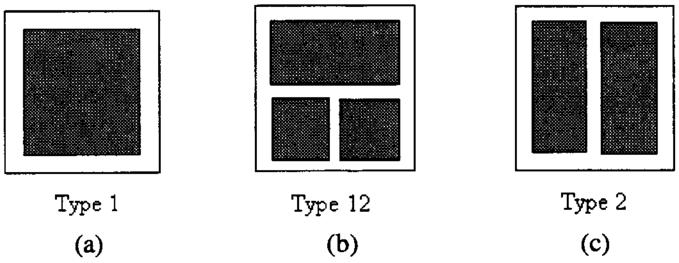
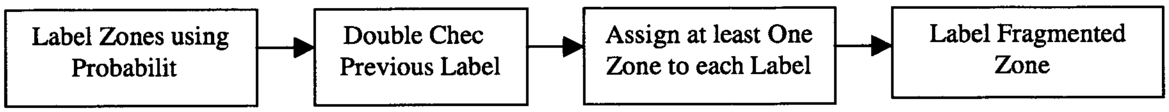


Figure 1. Examples of journal types.



Label Zones using Probabilit
Double Chec Previous Label
Assign at least One Zone to each Label
Label Fragmented Zone

Figure 2. The procedure for automatic labeling.

Table 1. Features Associated with Each Zone

The rule-based system variables in this table are unnormalized while the neural network-based system variables are normalized either to page dimensions (NTPD) or to zone contents (NTZC).

Zone Features	Rule-based System	Neural Network System
Geometric Features:		
Zone coordinates, pixels	Coordinate_Left, _Right, _Upper, _Lower	NTPD
Zone height and width, pixels	Height_Zone, Length_Zone	NTPD
Zone centroid (X and Y)		NTPD
Zone shape: 100 log(height/width)		NTPD
Zone area		NTPD
Median height and length of lines	Avg_Line_Height, _Length	
Median vertical spacing between lines	Avg_Line_Space	
Zone order, in sequence by top edge	R****	
Non-Geometric Features:		
Lines	Number_Line	R
Total characters	Number_Character	R
7-bit capital characters [A-Z]	Number_Capitalcharacter	NTZC
7-bit lower case characters		NTZC
Numerals		NTZC
Punctuation group ! " # $ % & ' ( ) * , - . : ; ? @ [\ ] ^ _ ` { | } ~		NTZC
Math symbol group (w/o minus) + / < = >		NTZC
3 groups of symbol pairs: [ ] ( ) { }	(special cases: parentheses)	NTZC
Commas	Number_Comma	NTZC
Other separately-totaled punctuation characters { . - ; : ' " * }	(special cases: . - ;)*	NTZC
Other separately-totaled characters	(special cases, e.g., ©)	
Highest-confidence characters (= 9)		NTZC
Less-than-highest confidence character (< 9)		NTZC
Characters with particular font attributes (6 non-disjoint totals, & total of attribute-frees)		NTZC
Number of words	Number_Words	
Number of initials (e.g., " A.")	Number_Middlename***	
Average font size		R
Maximum font size	R	
Dominant {Font Attribute, Font Size} Pair	Font_Attribute, Font_Size	
Zone avg. font size:Page avg. font size		Ratio

* The median line height, length, and vertical spacing are derived by first calculating the median upper and lower character boundaries for each line.

** While the rule-based system doesn't total up individual punctuation marks throughout the zone, it does calculate totals within particular places, e.g., number of dashes at the ends of lines during word count generation.

*** includes "jr.", "sr.", "II", etc.

**** R = numerical raw counts.

Table 2. Some features used in the Rule-Based System.

Zone Features	Description
Number_Degree	Number of "M.D., Ph.D., M.S., R.N, ..."
Number_Affiliation	Number of "names of city, country, hospital, department, ..."
Percent_Capitalcharacter	Percentage of Number_Capitalcharacter per character
Percent_Degree	Percentage of Number_Degree per word
Percent_Affiliation	Percentage of Number_Affiliation per word
Percent_Middlename	Percentage of Number_Middlename per word
Sentence_Headtitle	Check the existence of a word such as "review, letter, note, ..."
Sentence_Abstract	Check the existence of a word such as "abstract, summary,..."
Sentence_Subabstract	Check the existence of a word such as "aim, background, design, result,..."
Sentence_Keyword	Check the existence of a word such as "keyword, index word,..."
Sentence_Introduction	Check the existence of a word such as "introduction,..."
Sentence_Received	Check the existence of a word such as "received, revised,..."
Sentence_Affiliation	Check the existence of a word such as "correspondence, to whom, mailing,..."

Multi-lingual Processing

Issues in Cross-Language Retrieval from Document Image Collections

Douglas W. Oard
College of Library and Information Services
University of Maryland, College Park, MD 20742
oard@glue.umd.edu, http://www.glue.umd.edu/~oard/

Abstract

Over the past decade, broad-coverage cross-language text retrieval has progressed from isolated experiments on small collections to establish credible performance in large-scale evaluations. Extending this capability to document image collections presents some additional challenges that have not yet been well explored. This paper presents a general framework for cross-language retrieval, specializes that framework to retrieval from document image collections, and identifies opportunities for closer integration of the key enabling technologies and resources.

1 Introduction

Information retrieval systems seek to help users obtain information objects from large collections [2]. Early systems typically relied on manually assigned indexing terms, and such "controlled vocabulary" techniques were widely used in libraries to support the retrieval of printed documents. As storage costs declined and processing power improved, "free text" searching became cost effective and was widely deployed. Early applications of free text searching were limited to cases in which character-coded electronic text was available. More recent work on searching document image collections has yielded promising results, however, particularly when high-resolution document images are available [3].

Another trend with important implications for the nature of information retrieval is the rapid expansion in trans-boarder information exchange. Although research libraries and other specialized institutions have always collected documents written in many languages, modern networks now make vast collections of multilingual information available to any user. The past decade has seen substantial progress on the development of techniques for using queries expressed in one natural language to find documents written in another, a task that is typically referred to as Cross-Language Information Retrieval (CLIR) [9].

Present CLIR techniques are limited to electronic text, however. This paper proposes a framework for applying what we know about document image retrieval and cross-language retrieval to search multilingual collections of document images.

2 Framework

Figure 1 depicts a simplified process model for interactive information retrieval.

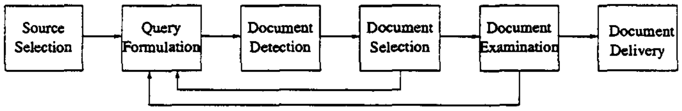


Figure 1: Information Retrieval process Model.

Source Selection. Information retrieval systems seek to provide information objects that contain information relevant to the user's information need. The first challenge is thus to select a system (or set of systems) that might contain information of the type desired. This is often a manual process, and it will not be addressed further in this paper.

Query Formulation. It is usually assumed that the user has a fairly specific information need that can be satisfied by some set of documents within the collection. The goal of the query formulation stage is to help the user develop the best possible formulation of the query. This is often an iterative process, as shown by the feedback loops from subsequent stages in Figure 1.

Document Detection. Detection is a general term that encompasses both searching relatively static collections and filtering dynamic document streams. The typical approach is to compute a figure of merit for each document that reflects the degree to which that document matches the query.

Document Selection. Interactive information retrieval is a synergistic process in which the machine applies relatively simple techniques to quickly cull promising documents from a large collection and then human abilities to rapidly recognize complex patterns are exploited once a manageable number of candidates have been identified. A compact display of important selection cues (title, author, date, etc.) is needed in the selection interface.

Document Examination. When the full text of the document is easily available, users are often able to improve their selection decisions by examining the document itself. Hypertext interfaces that support rapid browsing are often used for this purpose.

Document Delivery. Browsing interfaces provide one form of access, but sometimes additional processing is needed before the document can be used effectively. A printed copy may be desired, for example, or a professional translation of foreign language materials may be needed. Delivery is not discussed further in this paper, but it is identified as a separate stage here in order to emphasize that the purpose of the examination interface is to support choice, rather than use, of the documents being examined.

The remainder of this section explores the design of components to support the four central stages of this process model that are specialized to cross-language document image retrieval.

2.1 Support for Query Formulation

Queries can be posed explicitly, either as some form of selection criteria (using Boolean logic and proximity operators, for example) or as a set of "natural language" search terms. Alternatively, the query might be expressed implicitly by providing one or more examples of desirable (and/or undesirable) document images, and the user might be allowed to specify which aspects of the example(s) are particularly salient. For example, the user might wish to designate the body of a business letter as an example, but the addressee to which the example letter was sent might be of no consequence. The two techniques can be combined, using an explicit query to locate some document images and then enriching the query with selected document images as positive and/or negative examples, a process known as "relevance feedback." The key point here is that the query may contain character-coded electronic text, examples of document images, or a combination of the two. This means that CLIR systems for document image collections must generally search across modalities (between character-coded text and document images) as well as across languages.

2.2 Document Detection

Figure 2 shows the key components of the cross-language document detection stage. Most cross-language retrieval techniques are configured to process a specific language pair. When the document language cannot be reliably inferred from metadata or from the document source, automatic language identification techniques can be used to select appropriate language-specific processing (cf., [6]). If languages for which language-specific processing is not provided might be present in the collection, the language identification component can also be used to reject documents written in those languages.

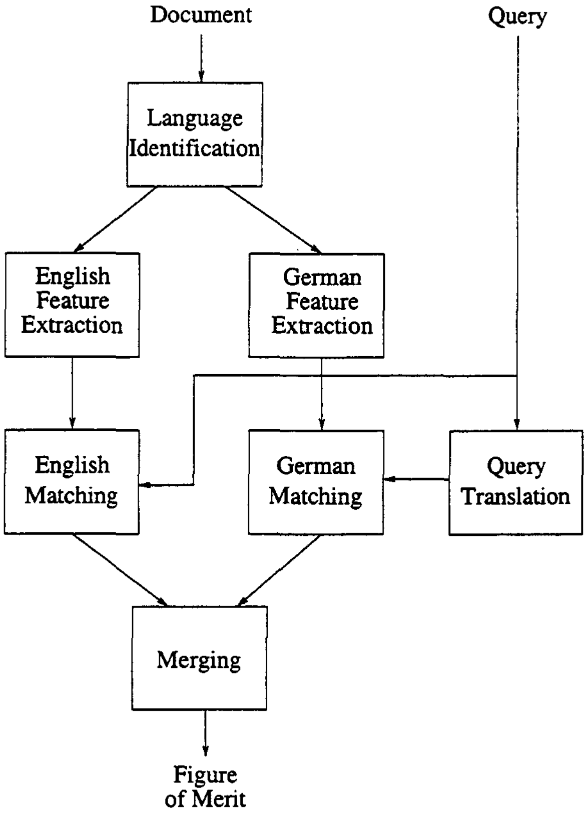


Figure 2: Cross-language document detection using query translation (English queries, English and German document images).

2.2.1 Feature Extraction

Two broad categories of features can be exploited for document image retrieval: document content and document structure. Content is typically characterized by identifying features (known generically as "terms") that are related to meaning and then weighting each term in a way that seeks to characterize that term's contribution to the meaning of a document. Three factors are generally used in the weight computation: the number of instances of that term in the document (more are better), the total number of term instances in the document (fewer are better), and the number of documents in which the term appears (fewer are better) [10].

Terms are extracted from document images by applying Optical Character Recognition (OCR) to identify individual characters and then combining the recognized characters until terms with the desired granularity are formed. The white space (spaces, tabs, etc.) that marks word boundaries can provide a useful cue to the appropriate granularity in some languages, but others (e.g., Chinese) lack reliable orthographic clues. Linguistic constraints and lexical knowledge can be used to identify plausible term boundaries in such cases, but OCR errors could complicate that processing by introducing symbols that appear to violate linguistic constraints. Any choice of terms naturally confounds some meanings and obscures the relationship between others. Often several words can be used interchangeably to convey nearly the same meaning (e.g., happy or glad). Optical Character Recognition (OCR) can exacerbate that problem, sometimes producing different results for separate instances of the same word within a single document.

Two general approaches have been developed for mitigating the effect of OCR errors on feature extraction in information retrieval applications. Both exploit observed regularities in character recognition errors. Character-confusion statistics can be used directly to postulate alternate strings (perhaps with lower weight) that might have resulted in the recognized characters. The same technique can be used with character-recognition algorithms that produce n-best (rather than 1-best) outputs. The other approach is to recognize character classes that exhibit little inter-class confusability rather than to recognize individual characters [12]. Terms formed from resulting "shape codes" exhibit greater ambiguity of meaning than the original words would have. Information retrieval systems perform fairly well in the face of increased ambiguity, particularly if relatively long queries (or examples of desirable documents) are provided [11], and the use of shape codes offers computational advantages over incorporation of character-confusion statistics.

Classification based on physical structure (layout) can be used directly to distinguish different document types such as business letters and newspapers. Physical structure can also provide cues about the logical structure of a document, and the logical structure can help to ascribe context to the terms. In a business letter, for example, it might be useful to know whether a name appears as the originator, as the addressee, or in the body of the letter. This contextual information can be used as an additional source of evidence for term weighting (e.g., giving more weight to terms in the lead paragraph of a news story) or as a basis for supporting queries that are matched against specific document components (e.g., a search for business letters to a specific addressee). Physical structure exhibits both cross-linguistic variations (e.g., vertical vs. horizontal writing) and cross-cultural variations within a single language (e.g., metric vs. U.S. letter paper sizes).

2.2.2 Cross-Language Matching

In cross-language retrieval it is necessary to (1) translate the terms in the query representation into the language(s) in which the documents are written, (2) translate the terms in the document representation into the supported query language(s), or (3) translate the terms in both into some common feature space. Query translation is the most efficient approach, and satisfactory response time is generally easily achieved when the queries are relatively short and are posed as electronic text. Long queries or instances of relevance feedback that might require on-the-fly OCR could shift that balance in favor of advance translation of the terms in every document.

There are four ways of obtaining the knowledge needed to translate the terms in documents and/or queries: (1) looking up term translations in a bilingual (or multilingual) lexicon, (2) algorithmically recognizing terms that are likely to be translation equivalents, (3) extracting useful relationships from a bilingual (or multilingual) corpus, or (4) by asking the user. A bilingual lexicon identifies one or more "target language" translations of each source language term, and it may include additional information such as part of speech or commonly co-occurring words that help to select to the correct translation. Some lexicons list translations in order of predominance in either general usage or in some application domain, and that information can be used as a basis for weighting alternatives when a single translation cannot be identified.

Unfamiliar names and newly introduced terminology pose a problem for systems that depend solely on lexical translation knowledge. When the source and target languages share a common character set, one simple technique is to retain unrecognized terms in the hope that they might be names or some other strings that would have the same representation in the source and target languages. More sophisticated cognate matching techniques can be applied (cf., [7]), and techniques which account for character-recognition errors and character-set differences are also available (cf., [5]).

Corpora (collections of documents that that use terms in representative ways) provide another source of translation knowledge that can be used alone or in conjunction with lexical and/or algorithmic sources. Parallel corpora, bilingual connections of translation equivalent documents, can be aligned to the sentence level fairly easily if sentence boundaries can be accurately detected since sentence length patterns are typically preserved across languages. Term co-occurrence statistics across aligned sentence pairs can then be used to postulate likely translations or as an indication of relative predominance among candidate translations for a term [8]. Cognate matching and/or a bilingual lexicon can be used to identify related regions in "comparable corpora" that contain documents in each language that are topically comparable but that are not translation equivalents (cf., [?]). There are also a number of less direct ways to improve the quality of lexicon-based translation using corpus statistics (cf., [1]).

It is clearly not possible to depend upon the user as the sole source for translation knowledge (since that would not be a cross-language retrieval problem!), but users with no knowledge of the target language might still help improve the accuracy of query translation performed using other techniques. Near-synonyms often group differently in different languages, so retranslation of each target language candidate back into the source language will sometimes provide even a monolingual user with enough cues to select the proper translation. For example, the German word "wagen" translates to either "car" or "risk" in English. The English word "car" retranslates to "wagen" and "auto," which could help a German speaker recognize the correct translation if reference to an automobile had been intended.

Three factors can adversely affect the performance of document or query translation: (1) translation ambiguity, (2) gaps and mismatches in lexical coverage, and (3) incorrect translation of noncompositional phrases. The first two factors deserve particular attention in the case of cross-language document image retrieval. Uncertain character recognition will necessarily magnify the effect of translation ambiguity somewhat no matter what technique is used, but the use of shape codes rather than confusion statistics could result in explosive growth of translation ambiguity. It is thus likely that shape codes will prove useful only for target language recognition. The second point is that lexical-coverage mismatch problems could be exacerbated in cross-language document image retrieval systems that use monolingual lexical resources for OCR error correction. Closely coupling the correction and translation processes could thus prove beneficial.

The remaining components of the document detection stage are essentially the same as those used in any cross-language retrieval application. Once the query and the document are represented by term weights in the same feature space, standard algorithms such as vector space, probabilistic, or Boolean matching can be performed. The result of this matching is a figure of merit that reflects the degree to which each document is estimated to satisfy the query. In monolingual applications, these values are typically used to construct a best-first ranked list of documents. The values are, however, not generally comparable across collections, nor are they generally comparable across different queries for a given collection. When multiple document languages are searched separately with different query translations, the values computed for each collection must be adjusted if a single rank-ordered list is desired. The nature of the adjustment depends on details of the translation and matching algorithms that are difficult to estimate, so the performance of matching in each language on a training collection with known relevance judgments is typically used as a basis for tuning this "merging" component (cf. [?]).

2.3 Selection and Examination

Because information retrieval systems typically make little use of factors such as word order and context, undesirable documents are invariably presented, even near the top of a list ranked in "best-first" order. Effective retrieval is thus a synergistic process in which the machine rapidly culls a manageable set of promising documents from the collection so that the user can quickly choose the most interesting documents. Recognition and translation errors make the machine's task more challenging than would be the case for monolingual retrieval of electronic text, so it is particularly important to provide the best possible support for selection and examination in cross-language document retrieval applications.

In the selection interface, documents are typically presented in a single ranked list, with each document represented using a compact set of features that users might find helpful in recognizing interesting documents. Users generally find document titles to be particularly valuable selection cues, so when structural cues are available to help locate a useful title within a document they should be exploited. Titles are often expressed as noun phrases, and simple techniques that produce readable title translations can be built by leveraging the limited range of linguistic phenomena that must be accommodated in such cases (cf. [4]). Users also typically find a few salient terms chosen from the document to be useful. The techniques used to select and weight terms for retrieval could also facilitate term selection for this purpose (unless shape codes are used). Temporal and numerical information that is typically found in a selection interface (e.g., the date the document was acquired and its length) are easily incorporated since no usual processing is needed.

Full text examination has proven to be quite popular in modern information retrieval systems. Monolingual document image retrieval poses no particular challenges in this regard since page images are easily displayed if adequate storage and bandwidth are available, but if translation is needed then two potential problems arise. One potential problem with a serial combination of OCR and translation is cascading errors. This architecture has be implemented in the Army Research Laboratory's Forward Area Language Converter (FALCon) system, and it appears that the resulting translations are of some value for assessing the contents of a document image that the user would otherwise be unable to read.[1] By more closely coupling the OCR and translation components, it may be possible to further improve the readability of the translation and thus improve the utility for document examination in a document image retrieval application.

The other problem is that both optical character recognition and machine translation are far slower than an interactive user might desire. Although information retrieval tasks are frequently modeled as rather narrowly goal-directed, experience suggests that many interactive search processes are marked by dynamic exploration and serendipitous discovery. Exploration and discovery would benefit from the availability of responsive retrieval systems that are able to operate inside the user's decision cycle. At present the only practical way to assure uniformly responsive support for full-text examination would be to perform massive translation in advance, but caching strategies offer a practical alternative that could provide rapid access to translations of frequently retrieved document images. Faster algorithms for each task, coupled with the ever-faster machines promised by Moore's Law, may ultimately obviate this concern completely.

3 Conclusions

Although broad-coverage cross-language document image retrieval systems do not yet exist, all of the enabling technology is now available and modular approaches based on existing components (page decomposition, optical character recognition, query translation, and machine translation) could easily be constructed. Optimal performance will likely require a closer degree of integration, however, both at the level of lexical resources and between the recognition and translation components. What is needed now is a testbed on which alternative integration strategies can be explored. The development of such a tool would be a significant step towards improved access to that portion of the world's storehouse of knowledge that presently exists only in printed form.

[1] http://rpstl.arl.mil/ISB/falcon.htm

Acknowledgments

The author is grateful to Bob Allen for his comments on an earlier version of this paper. This work has been supported in part by DARPA contract N6600197C8540.

References

[1] Lisa Ballesteros and W. Bruce Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 1997.

[2] D. C. Blair. *Language and Representation in Information Retrieval.* Elsevier, Amsterdam, 1990.

[3] David Doermann. The indexing and retrieval of document images: A survey. Technical Report CS-TR-3876, University of Maryland, Computer Science Department, February 1998.

[4] Genichiro Kikui, Yoshihiko Hayashi, and Seiji Suzaki. Cross-lingual information retrieval on the WWW. In *Proceedings of the First Workshop on Multilinguality in Software Engineering: The AI Contribution (MULSAIC).* European Coordinating Committee for Artificial Intelligence, August 1996.

[5] Kevin Knight and Johnathan Graehl. Machine transliteration. In *Seventeenth International Conference of the Association for Computational Linguistics*, 1997. http://www.isi.edu/natural-language/projects/nlg-publications.html.

[6] Dar-Shyang Lee, Craig R. Nohl, and Harry S. Baird. Language identification in complex, unoriented, and degraded document images. In *Proceedings of the Second IAPR Workshop on Document Analysis Systems*, pages 17–39, October 1996. http://cm.bell-labs.com/cm/cs/who/hsb/pub.html.

[7] I. Dan Melamed. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Third Workshop on Very Large Corpora*, 1995. http://www.cis.upenn.edu/~melamed/.

[8] I. Dan Melamed. Emperical methods for MT lexicon construction. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas*, 1998. http://www.cis.upenn.edu/~melamed/.

[9] Douglas W. Oard and Anne Diekema. Cross-language information retrieval. In *Annual Review of Information Science and Technology*, volume 33. American Society for Information Science, 1998.

[10] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beauliew, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Department of Commerce, National Institute of Standards and Technology, November 1994. http://trec.nist.gov.

[11] Mark Sanderson. Word sense disambiguation and information retrieval. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 142–151. Springer-Verlag, July 1994. http://www.dcs.gla.ac.uk/ir/papers/Postscript/sanderson94b.ps.gz.

[12] Alan F. Smeaton and A. L. Spitz. Using character shape coding for information retrieval. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 974–978, 1997.

OCR for Minority Languages

Christopher Hogan
Language Technologies Institute
4910 Forbes Avenue
Pittsburgh, PA 15213 USA
chogan@cs.cmu.edu

Abstract

In this paper I discuss the difficulties encountered when applying Optical Character Recognition (OCR) to minority languages. In particular, I explore the case of developing OCR for Haitian Creole (HC), a vernacular, minority language. Although HC is written with a variant of the Roman alphabet, no OCR device has ever been developed specifically with HC in mind, with the result that recognition can be fairly poor. I present a technique for post-processing OCR output that is independent of the OCR device being used, and demonstrate that it can improve OCR recognition for HC.

1 Introduction

Optical Character Recognition (OCR) is a important and necessary step in bridging the gap between the printed form of text that continues to play an important role in our society, and it's electronic form, which is increasingly necessary in order to deal efficiently with the huge amount of information that is now available. Increasingly, this technology has become cheaper and more available to anyone who can afford a computer.

Unfortunately, OCR technology, like most language technologies involving a large cost for research and development, has focussed almost exclusively on the major languages in the world, to the exclusion of the vast majority of languages for which there is not enough of a market to justify such an expenditure. Typically, the majority languages are English, the major European languages (French, German, Spanish, Italian), and perhaps Japanese. Non-majority languages, then, include not only those languages which have a small speaker-base, such as Icelandic, Eskimo and Haitian Creole, but also many significant world languages which are either politically or economically disadvantaged, such as Hindi, Chinese, Croatian, *etc.*

Nevertheless, the need for languages technologies to be adapted to non-majority languages is particularly acute for several reasons. Many non-majority languages are fighting an uphill battle against the influence of majority languages. The existence of useful technologies for majority languages only serves to increase the usefulness of switching to those languages while the lack of such technologies for non-majority languages eliminates alternatives to such a switch. Secondly, many non-majority languages have rich printed literary traditions, but have not begun the switch to electronic creation and dissemination of information; for these languages, there is a particular need for OCR. Finally, there are still many languages which are primarily oral and only infrequently written. The potential exists in such languages for language technologies to play a role in the formation and standardization of a literary and computational society.

The approach I advocate for bringing these technologies to non-majority languages is evolutionary rather than revolutionary. Rather than investing significant amounts of money and effort in developing technologies for such languages, it makes sense to adapt those which exist for the majority languages to the extent possible. This is the approach taken in this paper for OCR, and one which I believe can be made to work for many other languages and technologies.

In this paper I present a method for capitalizing on existing technologies when developing Optical Character Recognition systems for non-majority languages. The system I present is generalizable to a wide variety of languages, and is quickly and cheaply adapted to a given language. As a test case, I exhibit an OCR system built for Haitian Creole, a vernacular minority language. Such a system is, I argue, an example of an economically feasible way to bring language technologies to non-majority languages.

2 Optical Character Recognition

The process of Optical Character Recognition (OCR) is in some ways similar to that of acoustic speech recognition. In both cases, the text to be recognized is "encoded" in a modality other than the symbolic one that is desirable for use on a computer. The non-text representation must be "decoded" in order to become useful. The standard treatment of this kind of problem is to statistically model the process of recognition, then to use that model to "decode" images into the text they represent. This leads to the following fundamental equation:

$$\begin{aligned} T^* &= \arg\max_T P(T|I) && (1) \\ &= \arg\max_T \frac{P(I|T)P(T)}{P(I)} && (2) \\ &= \arg\max_T P(I|T)P(T) && (3) \end{aligned}$$

Where T^* is the decoded text, I is the image to be decoded, $P(I|T)$ is the *image model*, which captures the relationship between the image and the text, and $P(T)$ is the *language model*, which represents the *a priori* likelihood that a given sequence of text represents the language that we are trying to decode. The language model prevents the system from producing output that does not correspond to the real language, and thus acts as a corrective influence to the actual recognition mechanism.

In an OCR device that has been developed in this way, the image model and the language model remain distinct. Such a setup would allow one to swap out the language model for one language in favor of one for a different language, while preserving the same image model. This would permit easy change of the language (assuming, of course, that the image model did not also need to change), as no consideration of the image would need to be taken when modifying the language model.

The reality is, of the OCR software I have used, not one makes use of a user-modifiable language model, or anything more than a dictionary of words to model language-specific characteristics. While some may make use of character-level language models, such models are difficult to change while at the same time failing to model word-level phenomena. Thus, it is unfortunately necessary to use English- or French-specific image and character models to analyze Haitian Creole, Croatian, or any number of other languages written in variants of the Roman alphabet.

The solution I propose for dealing with the shortcomings of current OCR packages is to post-process the output of the OCR engine with a language-specific spelling corrector. Spelling Correction is usually designed to correct text that has been produced directly by human users, and which contains errors that have been introduced by typos or lack of knowledge on the part of those users. In the present case, spelling correction is used to correct errors introduced by the OCR device. While all OCR devices make some errors, and thus can benefit from spelling correction; in the case that an OCR device is used to recognize a language for which it was not designed, the number of errors may be significant, and some kind of language-specific correction is certainly indicated.

Performing spelling correction as post-processing of OCR output has an unexpected positive side effect. In order to change the language that is recognized, it is only necessary to modify the spelling corrector, not the OCR device itself. Thus, a closed OCR can be modified to perform OCR for any number of similar languages. I will describe the results of using a French OCR device together with a language-specific spelling corrector to recognize Haitian Creole.

3 Minority Languages

A large number of the world's 6000 languages are what I will call *minority languages*. A *minority* language is a language that either has few speakers, or is otherwise politically or economically marginalized. Other terms used are: *less-commonly taught languages* or *low-density languages*[1], each indicating marginalization in a different way, yet typically referring to the same set of languages. The results of this marginalization are a lack of written and computational resources in the language, a loss of the language in favor of more prestigious languages, and possibly language death from lack of transmission.

Besides being minority languages, many languages are also *vernacular languages*, defined as an " 'everyday spoken language or languages of a community, as contrasted with a standard or official language'—generally, a 'Low' as opposed to a 'High' variety in Ferguson's [11] terms" [27]. Being primarily spoken, rather than written, vernacular languages typically experience low literacy rates, have no significant corpus of written materials, and very often have no standard written form.

3.1 Haitian Creole

Haitian Creole (HC) is a creole language [15] that arose as a result of the contact between French and several African languages that occurred in the context of the slave trade in the Caribbean. The result is a language with a French-based lexicon, but non-French grammar. The identity of the African languages that contributed to the grammar is a subject of debate [20, 19, 26].

Haitian Creole may be considered a minority language. Although more than 95% of Haitians speak

[1] *Low-density* is the current term of choice when referring to language for which few computational data resources exist.

only HC[13], and the language is in little danger of dying out (with 7,382,000 speakers [13]), it suffers from depressed social status *vis-à-vis* French, which, until 1986 was the only official language of Haiti, and which continues to enjoy higher official and social recognition. In addition, HC is spoken by the poorest people in a country which is the poorest in the Western Hemisphere [7, 29]. Thus, HC is classified as a minority language based on its low social and economic status.

Haitian Creole is also a vernacular language, having very little tradition of writing. Until recently, all education, and most written materials, including the Haitian Constitution were only available in French. Although this situation has changed somewhat with the recognition of HC as an official language of Haiti, there remains much to be done. Currently, primary education is conducted in HC, but only one university in Haiti (*Université Caraïbe*) uses HC as a language for education. Literacy overall remains low (45–85% depending on the estimate [7]), and there is a lack of written materials (some newspapers, few books, no libraries).

One result of HC's being a vernacular language is a lack of standardization in the writing system. During the last century, 11 distinct writing systems have been developed for HC [30, 24], including some based on French, and others based on more general linguistic interpretations of Roman characters. The current system, known as IPN (*Institute de Pédagogie National*), is non-French based. The characters used are those in the English alphabet plus *è*, *ò* and sometimes *à*. Of these, only *ò* is not also found in French.

In spite of the current standard, there is still much variation in written HC. Although the language has been standardized at the phoneme level (*i.e.* the standard specifies a specific character to represent each phoneme), it has not been standardized at the word level (*i.e.* the standard does not specify a specific way to spell each word). This situation may be compared to English, which has been standardized at the word level, but is inconsistent at the phoneme level. The result is that words which are really the same, but which are pronounced differently either because of different linguistic context or different dialects of the speakers, may be written differently. The problem is particularly acute for foreign words (like names), as it is unclear whether it is the spelling or the pronunciation that should be preserved, and, if the latter, how non-Creole phonemes should be converted. As an example, we [10] have recorded up to 23 variations on the word *Washington*. Beyond this, there is general uncertainty about how to use the spelling conventions, and some influence from previous spelling systems which older speakers may have learned as youths. As an example, the word for 'week' is fairly typical of the kind variations that are present in HC [2] (lexical frequency is given in parentheses):

semèn	(295)
semenn	(28)
semen	(20)
semènn	(11)
senmenn	(2)

3.2 NLP Tools for Minority Languages

Recently, there has been a good deal of discussion about the possibility of developing Language Technologies for minority languages [31, 12]. The basic issue is that a large number of the world's languages are endangered [17], the majority for socio-economic reasons.[2] Until now, the availability of computers, and more specifically, the Internet, has only served to increase the hegemony of English over other languages by making English all the more available. Language Technologies have the potential to reverse this trend by making available Machine Translation, Information Retrieval and other systems that allow a user to access information in his or her own language. The primary barrier to this goal is cost: it is currently too expensive to develop new systems for every available language. There are two ways to overcome this barrier, both of which are probably necessary: (1) spend the money that is necessary, or (2) develop techniques that cost less.

Optical Character Recognition is a case where inexpensive techniques can be developed. Other language technologies, such as Machine Translation or Speech Recognition, deal with aspects of language that vary substantially from language to language, requiring significant reconfiguration for a new language. Most of the world's languages, however, are written in one of a few widespread writing systems (Roman, Cyrillic, Arabic, Han, Devanāgarī, *etc.* [9]). Thus, OCR systems developed for one language are potentially transferable to many other language which use the same writing system.

In this paper I will show that it is possible to implement a language-specific post-processing module that makes it possible to use French OCR systems for the recognition of Haitian Creole. While these languages are related lexically,[3] their orthographic representations are fairly distinct. The technique, however, is general, and could be used to develop language-specific OCR for any language that uses the Roman alphabet.

[2] Namely that some other language is more socially prestigious or that there are economic benefits to speaking it.

[3] By 'related', I do not intend to imply that there is standard historico-linguistic connection between the language, as there is between English and German, but rather that many, if not most of the lexical items in HC are taken from French.

4 Applications

While the goal of Optical Character Recognition is often the symbolic text itself, sometimes the text is also the input to some other computational process [14]. In particular, the text may serve as input to other natural language processing tasks, such as Machine Translation (MT) or Information Retrieval (IR). However, such external components are typically not equipped to handle errorful input, and may react in unexpected ways.

The case of Machine Translation is informative. All current machine translation systems are designed to take symbolic text as input.[4] If the translation system is knowledge-based, as most high-quality systems currently are, the range of forms that it will accept will be limited to the forms present in its lexical database or accepted by its morphological rules. Usually these forms do not include misspellings. In the case that the MT system is empirically-based (trained on naturally occurring data), the system may be able to recognize such misspellings as have been encountered sufficiently frequently before. However, the types of errors encountered in OCR output are typically related to confusability between the forms of letters, while the kinds of spelling errors typically committed by humans are related to confusability between the sounds that the letters represent. These errors usually give rise to different forms, so that an empirically-based MT system trained on online text would probably not be substantially better off than a knowledge-based one. One extreme solution would be to train an empirically-based MT system on OCRed documents so that it would learn to translate correctly the kinds of forms present in such documents.

The results of presenting uncorrected text to most MT systems is unpredictable. In most cases, MT systems will simply pass through words that they do not recognize, under the assumption that they may be proper names. In extreme cases, however, errors in the input may cause components, such as parsers, to degrade in extremely ungraceful ways. Even in cases where the components have been designed with robustness in mind, the designers have typically taken account of only one kind of ill-formedness, and are not equipped to tackle the kinds of errors produced by OCR systems.

4.1 Translation of Errorful Text

In order to demonstrate the effect of ill-formed input on an MT system, I have undertaken a small experiment in translating from French to English. A single paragraph of French newswire was selected from a well-known newswire service [1]. The original French is presented below:

Original French Text:

> *Le prochain sommet de l'OTAN fin avril à Washington affirmera "que la porte de l'OTAN reste effectivement ouverte" et annoncera "un programme pour préparer les pays candidats à respecter les critères stricts" d'adhésion à l'alliance, a déclaré vendredi le secrétaire d'Etat américain Madeleine Albright, ajoutant que "L'OTAN est maintenant rejointe par trois fières démocraties qui ont prouvé qu'elles étaient capables de remplir leurs responsabilités d'alliés".*

The paragraph was translated into English using the SYSTRAN translation system [23] as provided by the Altavista web searching system [4]. The resulting English is shown below:

Translation of 0% degraded French:

> *The next node of NATO at the end of April in Washington will affirm "that the gate of NATO remains indeed open" and will announce "a program to prepare the applicant countries to respect the strict criteria" of adhesion to alliance, declared Friday the American Secretary of State Madeleine Albright, adding that "NATO is now joined by three proud democracies which proved that they were able to fill their responsibilities for allies".*

The French text was then artificially degraded by modifying a proportion of the characters. The degraded text was again translated into English. The results are presented below for 1%, 5%, and 10% degradation.

Translation of 1% degraded French:

> *the next node of NATO end April with Washington1affirmera "that the gate of NATO remain indeed open" and announce "a program to prepare the country candieats to respect the critèrfs strict" some adhesion with alliance, have declare Friday the secrétairf of State amérjcain Madeleine Albright, add that "The NATO be now join by three proud democracy which have prove ru' they be able to fill their responsibility of alliét".*

Translation of 5% degraded French:

> *Next the sommeu ee NATO at the end of April in Wasiington will affirm "that*

[4] How that text is interpreted is a different issue: some MT systems can interpret SGML or HTML. Nevertheless, there is (as yet) no MT system that can operate directly from sound or image input without first converting it to text.

> *the porue NATO remains indeed open" and will aonpncera "a program to prepare the countries candjdats with respecufr the turicts criteria" of bdhésion to the amliance, declared vendrfdi the Secretary of State américajn Madeleine Albright, akovtant that "NATO is now rejoinue by troit proud eémocraties which proved that they were able to rfmplir their allied rfsponsacimities e"'.*
>
> **Translation of 10% degraded French:**
>
> *Lf prpchajn sommft ee l1PTBN gin April 1 Washington will agfirmera "that the portf1de NATO rette efgecuiwemfnt ouvertf" and annoncesa "a qrogrbmme for pr1parfr1les qays candidatt with respectfr puts csiuères strjcts" adiésion at alliance, declared Friday the secrétaise of State anéridain Madelejne Albrihht, bkoutant qve "NATO is maintenbnt rejojnte not three fièses démocsatjfs which ont1prouvé that they éuaient able to fill allied leursrfsponsacimities e"'.*

At 10%, the output of translation is almost incomprehensible, although the French input is still more or less readable. For the most part, when the MT system encountered unknown words due to misspellings, it simply passed them through unchanged. In several cases, however, the grammatical information in the misspelled word caused changes in the translation of the sentence. In the 1% error condition, for example, the future tense marker *will* was missed, and several French noun-adjective pairs were incorrectly left in that order, rather than being swapped into the correct adjective-noun order, as in the 0% condition.

If, as indicated by [18], OCR rates are typically 7–16%, some kind of correction is definitely indicated before Machine Translation is performed.

4.2 FALCon

The current OCR correction mechanism was designed for use as part of a prototype system called FALCon [5] developed by the Army Research Labs (ARL). FALCon, which stands for Forward Area Language Converter, is a device that delivers scanning and language translation capabilities to the person in the field, and is pictured in Figure 1. FALCon consists of a sheet-feed scanner and a standard laptop computer packaged in a briefcase together with battery, electrical and telecommunication support. FALCon's software includes OCR, commercial and research MT systems for many European languages and Haitian Creole, and keyword highlighting mechanisms.

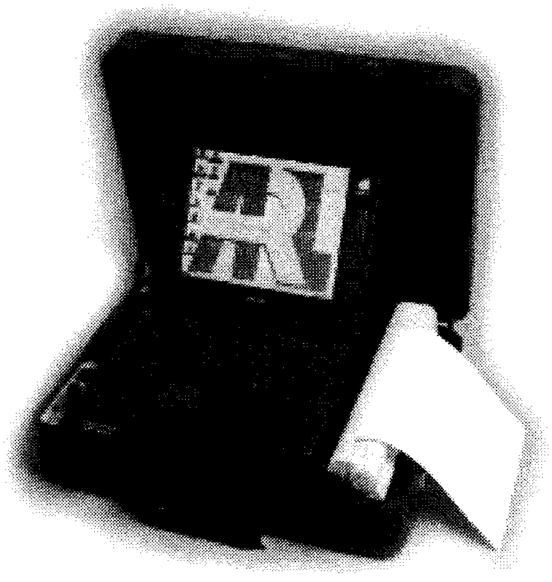


Figure 1: The FALCon System (picture courtesy of ARL)

FALCon is intended to enable a user to rapidly examine documents and assess whether they should be forwarded to human translators for more accurate translations. In this use, all components of FALCon are combined, as pictured in Figure 2. The OCR corrector component is not available for all languages, and may therefore not be applied in all cases.

5 Spelling Correction

Fully automated spelling correctors typically consist of two components [18]. One component, the generator (GEN), generates alternate spellings of misspelled words, perhaps by relying on a dictionary. For an interactive spelling corrector, this is all that is needed: the human is expected to choose from among the choices presented. For full automation, however, a selection component (SEL) must be added to select from among the choices produced by the generator. This component may take context into account or rely on other heuristics.

As a GEN component, I have drawn on [22], which presents a spelling corrector based on finite-state technology. The basis for this component is a search algorithm for finite-state grammars which includes spelling correction. The search algorithm relies on the notion of edit distance, defined as the minimum number of insertions, deletions or replacements of a symbol or transpositions of adjacent symbols that are necessary to convert one string into another (see Figure 3). The algorithm searches for all strings recognized by the finite-state grammar that are within a given edit distance of the misspelled word. Words outside of this edit distance are eliminated as can-

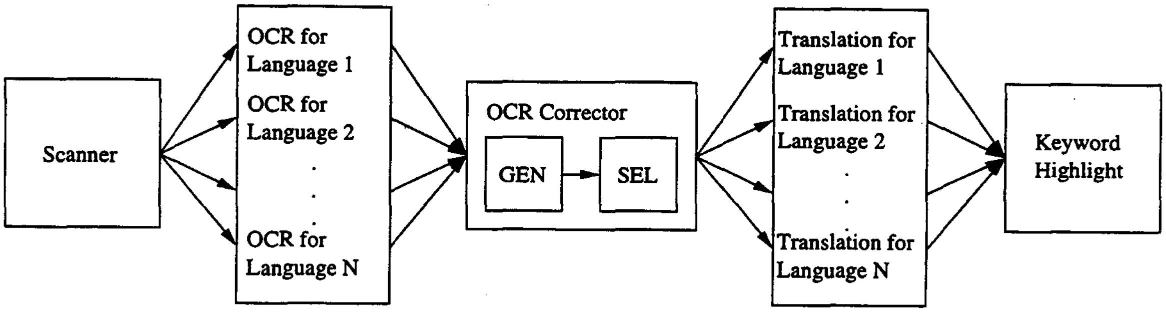


Figure 2: Block Diagram of FALCon functional units

Let:
$X_{1...m}$ be a base-1 string with m characters
$Y_{1...n}$ be a base-1 string with n characters
X_i be the ith character of X
Y_j be the jth character of Y
$c(ch_1, ch_2)$ be the cost of substituting ch_1 for ch_2
$d(ch)$ be the cost of deleting ch

Then:

$$ed(X_{1...i}, Y_{1...j}) = \begin{cases} \textit{Boundary conditions:} & \\ \max(m,n) & \text{if } i = -1 \vee \\ & \quad j = -1 \\ j+1 & \text{if } i = 0 \\ i+1 & \text{if } j = 0 \\ \textit{Substitutions:} & \\ c(X_i, Y_j) + ed(X_{1...i-1}, Y_{1...j-1}) & \text{if } \exists c(X_i, Y_j) \\ ed(X_{1...i-1}, Y_{1...j-1}) & \text{if } X_i = Y_j \\ \textit{Transpositions:} & \\ 1 + \min(ed(X_{1...i-2}, Y_{1...j-2}), & \text{if } X_{i-1} = Y_j \wedge \\ \qquad ed(X_{1...i}, Y_{1...j-1}), & \quad X_i = Y_{i-1} \\ \qquad ed(X_{1...i-1}, Y_{1...j})) & \\ \textit{Deletions:} & \\ d(X_i) + ed(X_{1...i-1}, Y_{1...j}) & \text{if } \exists d(X_i) \\ d(Y_j) + ed(X_{1...i}, Y_{1...j-1}) & \text{if } \exists d(Y_j) \\ 1 + \min(ed(X_{1...i-1}, Y_{1...j-1}), & \text{if otherwise} \\ \qquad ed(X_{1...i}, Y_{1...j-1}), & \\ \qquad ed(X_{1...i-1}, Y_{1...j})) & \end{cases}$$

Figure 3: The Generalized Edit Distance Function.

didates as soon as possible. During the search, the edit distance threshold is increased until a matching word is found, or until a user-defined maximum acceptable value is reached.

This spelling correction algorithm is designed to operate on arbitrary finite-state automata, and is extendible to finite-state transducers. The full generality of finite-state transducers may be necessary for languages with complex morphology, such as Turkish or Finnish. For Haitian Creole, however, which has very little morphology, the full expressive power of finite-state automata is unnecessary. Nevertheless, finite-state techniques, which are known for their computational efficiency, enable the algorithm to work for most languages. For those languages, such as Haitian Creole, that do not require as much expressive power, a simple finite-state automaton that only recognizes the words in a fixed vocabulary may suffice.

In addition to the corrected forms, the spelling corrector also passes the original spelling through. This form is given a low weight, and is passed to the language modeller along with the corrected forms, which are given a higher weight. Doing this permits the language modeller the option of selecting correctly spelled out-of-vocabulary words if it is able to do so. This may be the most desirable behavior for word classes such as proper nouns which are likely to be out of vocabulary.

The SEL component used in the present system is a trigram backoff language modeller that is also used as a component of our MT system [6]. In addition to probabilities derived from the language model, the modeller is able to take into account scores that have been passed in from the translation engines. These scores can be used to bias the language modeller for or against certain types of words, and is used in the current system to prefer forms predicted by the corrector to the original forms provided by OCR.

5.1 Improvements

During initial testing, it became clear that the algorithm needed to be modified slightly to improve its ability to work with a given language and OCR software. One of the most common mistakes that all OCR engines make when recognizing Haitian Creole is to substitute unaccented letters for accented ones (i.e. 'e' for 'è', 'o' for 'ò'). Because this type of mistake causes humans very few difficulties, errors of this type are also very likely to be the result of human error in the production of the original text.

As a solution to this problem, and to allow greater flexibility for the user of the system, the user may define modifications with a cost other than one (the default cost assigned to all substitutions, deletions and transpositions by the edit distance function). See Figure 3 for such a generalized edit distance. By defining modifications with very low cost (such as zero), the edit distance can be biased in favor of those modifications if they exist. The current system allows zero-cost substitutions and deletions. The following is the list of zero-cost replacements that have been found to be useful for Haitian Creole.

a	⟷	à
e	⟷	è
o	⟷	ò
r	⟷	w

(r ⟷ w is used because those characters are very often substituted for one another in Haitian writing [3]. Although this is an orthography problem, and not a problem with the OCR device, it can be corrected nevertheless.)

In addition, deletions of characters that are not in the alphabet are freely permitted. In this way, artifacts of recognition that cannot be words in Haitian Creole can be automatically eliminated.

6 Results

The figures on the next page show the results of using the OCR corrector. Some of these figures are taken from a study conducted by ARL [25]. Figure 4 shows the original Haitian Creole text. It may be noted that there are some errors in the original text: *lòt* 'other' is also spelled *lot*, and there is an extraneous *d* inserted before *jen*.

Figures 5 and 7 are the results of running two different OCR devices on the original text. Both OCR devices have difficulties with *ò*, which is the only Haitian Creole character which is not used in French (both OCR packages were for French).

Figures 6 and 8 show the results of running the OCR corrector on the recognized input. Neither punctuation nor capitalization are preserved. The OCR corrector is able to correct many of the errors introduced by the OCR device, especially those resulting from loss or modification of accents. Furthermore, the corrector catches some of the errors introduced by human error (such as *lot* for *lòt* in Figure 8).

7 Previous Work

The present system is based on ideas from Tong [28], and is similar to the statistical methods described in [18, 21, 8].

7.1 Tong 1995

The system outlined by Tong [28] calls for a language-specific GEN component. The component is implemented as a weighted confusion matrix which specifies the joint distribution of characters input to and output from the OCR device. The confusion matrix is determined by comparing the output of the OCR device to the true original text which was presented to it. The confusion matrix is used to compute, for each word output from the OCR device, the ten most probable input forms which might have caused it. These ten forms are then output to the SEL component.

The SEL component in [28] is a statistical language modeller of the type commonly found in Speech Recognition systems[16]. In this case, a trigram backoff model was used.

The present system has modified the GEN component in Tong's system substantially. I believe that the confusion matrix method is inadequate for the following reasons.

- The confusion matrix has no knowledge of words in the target language; all it knows about is the relationship between characters on the input and output of OCR. The result is that many of the top ten forms mentioned earlier are likely to be non-words. That is, given a form that is output by the OCR device, whether correctly spelled or not, ten forms will be generated that are likely to have given rise to that form, as predicted by the confusion matrix. The problem is that some of these forms may not be correctly spelled words. This makes the task of SEL more difficult, since it is unable to distinguish errors generated by the confusion matrix from those generated by the OCR device itself. The result is that SEL is more likely to pick a non-word.

- Along the same lines as the previous problem, there is no guarantee that any of the top ten confusable forms are actually correctly spelled words. In this case, SEL can never make a correct, or even reasonable, choice.

- It is necessary to train the confusion matrix before it can be used. Training involves selecting

Pèp Ayisyen,
Depi plis pase 4 mwa Ayiti mare nan yon gwo kriz politik: Premye Minis la bay demisyon l yon bò, yon lot bò, eleksyon yo gen pwoblèm. An reyalite, se kriz elektoral la ki pou yon gwo pati responsab kriz gouvènmantal la.
Se pou sa mwen te di, depi fen mwa d jen an, li pap fasil pou nomnen yon lòt premye minis san kesyon eleksyon an pa jwenn yon solisyon.

Figure 4: Original version of Haitian Text Sample

Pèp Ayisyen,
Depi plis pase 4 mwa Ayiti mare nan yon gwo kriz politik: Premye Minis la bay demisyon l yon bà, yon lot bà, eleksyon yo gen pwoblèm. An reyalite, se krix elektoral la ki pou yon gwo pati responsab kriz gouvènmantal la.
Se pou sa mwen te di, depi fen mwa d jen an, li pap fasil pou nomnen yon làt premye minis san kesyon eleksyon an pa jwenn yon solisyon.

Figure 5: First OCRed version of Haitian Text Sample

Pèp ayisyen depi plis pase 4 mwa ayiti mare nan yon gwo kriz politik premye minis la bay demisyon l yon bà, yon lòt ba eleksyon yo gen pwoblèm. an reyalite se kriz elektoral la ki pou yon gwo pati responsab kriz gouvènmantal la te pou sa mwen te di depi fen mwa a jen an li pap fasil pou nomnen yon lòt premye minis san kesyon eleksyon an pa jwenn yon solisyon

Figure 6: Corrected form of First OCRed version

Pèp Ayisyen,
Depi plis pase 4 mwa Ayiti mare nan yon gwo kriz politik. Premye M˜s la bay deniisyon 1 yon bo, yon lot bo, eleksyon yo gen pwoblèm. An reyalite, se kriz elektoral la ki pou yon gwo pati responsab kriz gouvènmantal la.
Se pou sa mwen te di, depi fen mwa d jen an, li pap fasil pou nomnen yon lôt premye minis san kesyon eleksyon an pa jwenn yon solisyon.

Figure 7: Second OCRed version of Haitian Text Sample

Pèp ayisyen depi plis pase 4 mwa ayiti mare nan yon gwo kriz politik premye 4 la bay demisyon l yon bo, yon lòt bò eleksyon yo gen pwoblèm. an reyalite se kriz elektoral la ki pou yon gwo pati responsab kriz gouvènmantal la te pou sa mwen te di depi fen mwa a jen an li pap fasil pou nomnen yon lòt premye minis san kesyon eleksyon an pa jwenn yon solisyon

Figure 8: Corrected output of second OCRed version

on-line text (from the same language and preferrably the same domain as the target text), printing it out using a device and style similar to that of the text one would like to recognize, scanning the printouts, and comparing the output of OCR to the digital original. Not only is this time-consuming, but there are many variables which are likely to affect the quality of the result.

- Once a confusion matrix has been generated, it is specific to the OCR device used during training. Different OCR devices have different confusion characteristics, making it difficult to design a generic GEN component (so as to avoid having the user perform training).
- A device-specific GEN component is also undesirable from an esthetic standpoint. The OCR corrector should depend only on the language being recognized, as does a standard spelling corrector. All device-specific characteristics should be contained in a different component, preferrably relegated to the OCR software itself. The introduction of device-specific characteristics into the spelling corrector breaks this neat symmetry.

8 Conclusion

A large number of the languages in the world may be classified as minority languages. Such languages have received few benefits from the significant progress that has been made in language technologies. In this paper, I have argued that certain language technologies, and optical character recognition in particular, can be easily and cheaply carried over to minority languages.

I have presented a technique for adapting commercially available OCR systems for a majority language (in this case French) to a minority language (in this case Haitian Creole). The technique is general, and can be used to adapt any OCR device to recognize languages written in the same writing system. Furthermore, the technique proposed here relies in no way on OCR device-specific characteristics, and does not require training involving the OCR device.

In the future, I hope to run more rigorous tests of the system, both of the component alone, and as part of the FALCon platform. I am also investigating the possibility of using the OCR corrector to normalize orthographic variants, as part of a Machine Translation system.

References

[1] Agence France-Presse. Available online at: http://www.afp.com

[2] Allen, J. H. 1998. Lexical Variation in Haitian Creole and Orthographic Issues for MT and OCR Applications. In the *Workshop on Embedded MT Systems: Design, Construction, and Evaluation of Systems with an MT Component.* Held in Conjunction with Association for Machine Translation in the Americas (AMTA-98), 28 October 1998, Langhorne, Pennsylvania.

[3] Allen, J. H. and Hogan, C. M. 1998. Evaluating Haitian Creole Orthographies from a Non-literacy-based Perspective. Presented at the *Society for Pidgin and Creole Linguistics*, New York City.

[4] The AltaVista Web Search Engine. Available online at: http://www.altavista.com

[5] Army Research Labs. *Falcon System.* Available online at: http://rpstl.arl.mil/ISB/falcon.html

[6] Brown, R. and Frederking, R. E. 1995. Applying Statistical English Language Modeling to Symbolic Machine Translation. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI '95)*, pages 221–239.

[7] Central Intelligence Agency. 1998. *The World Factbook.* Available online at: http://www.odci.gov/cia/publications/factbook

[8] Church, K. W. and Gale, W. A. 1991. Probability Scoring for Spelling Correction. *Statistics and Computing*, 1, 93–103.

[9] Daniels, P. T. and Bright, W., eds. 1996. *The World's Writing Systems.* Oxford University Press, Oxford, England.

[10] Eskenazi, M., Hogan, C., Allen, J., and Frederking, R. 1998. Issues in Database Design: Recording and Processing Speech from New Populations. In *Proceedings of the First International Conference on Language Resources and Evaluation*, 28–30 May 1998, Granada, Spain. Vol 2, pp. 1289–1293.

[11] Ferguson, C. 1959. Diglossia. *Word* 15, 325–340.

[12] Gerber, L., moderator. 1998. The Forgotten Majority: Neglected Languages. Panel at the *Meeting of the Association for Machine Translation in the Americas (AMTA '98).* Langhorne, Pennsylvania.

[13] Grimes, B. F. 1996. *Ethnologue.* Summer Institute of Linguistics, Dallas, Texas. Available online at: http://www.sil.org/ethnologue

[14] Hogan, C. Embedded Spelling Correction for OCR with an Application to Minority Languages. In the *Workshop on Embedded MT Systems: Design, Construction, and Evaluation of Systems with an MT Component.* Held in Conjunction with Association for Machine Translation in the Americas (AMTA-98), 28 October 1998, Langhorne, Pennsylvania.

[15] Holm, J. 1988. *Pidgins and Creoles. Vol 1: Theory and Structure.* Cambridge University Press, Cambridge, England.

[16] Jelinek, F. 1985. Self-organized language modeling for speech recognition. IBM report.

[17] Krauss, M. 1992. Statement of Mr. Michael Krauss representing the Linguistic Society of America. In *U.S. Senate, Native American Languages Act of 1991: Hearing before the Select Committee on Indian Affairs.* Washington, DC: Government Printing Office, pages 18–22.

[18] Kukich, K. 1992. Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*, 24:377–439.

[19] Lefebvre, C. 1996. The tense, mood and aspect system of Haitian Creole and the problems of transmission of grammar in creole genesis. *Journal of Pidgin and Creole Languages*, 11(2), 231–311.

[20] Lumsden, J. S. 1994. Possession: Substratum semantics in Haitian Creole. *Journal of Pidgin and Creole Languages*, 9(1), 25–49.

[21] Mays, E., Damerau, F. J., and Mercer, R. L. 1991. Context Based Spelling Correction. In *Information Processing Management* 27(5):517–522

[22] Oflazer, K. 1996. Error-tolerant Finite-state Recognition with Applications to Morphological Analysis and Spelling Correction. *Computational Linguistics*, 22(1):73–90.

[23] Ruby, C. 1998. System Demonstration: SYSTRAN Enterprise. In *Proceedings of the Meeting of the Association for Machine Translation in the Americas (AMTA '98).* Langhorne, Pennsylvania, pp. 498–500.

[24] Schiefflin, B. and Doucet, R. C. 1992. The 'Real' Haitian Creole: Metalinguistics and Orthographic Choice. *Pragmatics*, 2:3, 427–443.

[25] Schlesiger, C. D. and DeCrozan, L. J. 1998. *Comparison of Two French OCR Packages—How They Handle Haitian-Creole Text.* Personal communication.

[26] Singler, J. V. 1996. Theories of creole genesis, sociohistorical considerations, and the evaluation of evidence: The case of Haitian Creole and the Relexification Hypothesis. *Journal of Pidgin and Creole Languages* 11(2), 185–230.

[27] Tabouret-Keller, A., Le Page, R., Gardner-Chloros, P., and Varro, G., eds. 1997. *Vernacular Literacy: A Re-evaluation.* Oxford: Clarendon Press.

[28] Tong, X. 1995. *Automatic Word Error Correction in Text and OCR Post-Processing.* Master's Thesis, Computational Linguistics Program, Carnegie Mellon University.

[29] US Department of State, Bureau of Public Affairs. 1995. Background note: Haiti, June 1995. Available online at: http://www.tradecompass.com/library/dos/bnotes/HAITI.html

[30] Valdman, A. 1988. Diglossia and Language: Conflict in Haiti. *International Journal for the Sociology of Language*, 71, 67–80.

[31] Williams, B., Nadeu, C. and Monaghan, A., organizers. 1998. *Workshop on Language Resources for European Minority Languages.* Granada, Spain.

Recent Improvements in the BBN OCR Syste

Richard Schwartz, Zhidong Lu, Prem Natarajan
Issam Bazzi, Andras Kornai, John Makhoul

BBN Technologies, GTE
Cambridge, MA 02138, USA

ABSTRACT

We describe several improvements that we have made in the BBN BYBLOS OCR System. First, we adopted continuous density hidden Markov models (HMMs) rather than discrete density HMMs. This resulted in improved accuracy when more training data is available. It also allowed us to use unsupervised speaker adaptation algorithms (borrowed from speech recognition) for adaptation to font, style, and quality. Second, we sped up the character recognition by a factor of about 50 so that a full page of 2,000 characters requires about 30 to 40 seconds for processing. Third, we tested the system on Chinese characters. This required development of tools to create a training corpus from available sources. It also required techniques for dealing with an open set of characters, where some of the characters may have no real training data. The end result was 1.2% character error on newspaper data.

1. INTRODUCTION

We have previously described the use of the BBN BYBLOS speech recognition system for language-independent OCR [1-4]. After finding lines of text, we compute a feature vector for each narrow vertical slice of the image of each line. We then have a one-dimensional input sequence of feature vectors, which we model as the output of a hidden Markov model (HMM). Beyond the initial image manipulation and feature extraction, the remainder of the processing (model training and recognition) uses the BBN BYBLOS speech recognition system [5] with no source code modifications. We have previously reported man advantages to this approach. First, there is no explicit character segmentation process in either the training or recognition. This allows us to train on a large corpus with minimal effort. Second, the recognition process does not suffer from segmentation errors, and therefore is quite able to handle scripts where the characters are connected - such as Arabic. Finally, we can use all of the mature techniques developed in speech recognition for creating more powerful models or making the system robust to various degradations.

At the preceding SDIUT97 workshop [3], we reported ver promising results on Arabic using the ARPA/SAIC Arabic Corpus [6] and on English OCR using the University of Washington Corpus [7]. This system used discrete HMM models to model the features that result for the characters. It also ran quite slowly, taking 30 minutes or more for a full page of text. It had only been used for alphabetic languages with relatively fe characters. And it had only been tested on relatively clean images.

Since the SDIUT97 workshop, we have made several improvements to the system including:

- upgrading the models from discrete densities to continuous densities, and adding techniques for unsupervised adaptation to font, style, and quality,
- making it much faster,
- extending the system to Chinese OCR, including the development of tools for rapid creation of training corpora, and
- developing techniques for dealing with degraded documents.

This paper discusses the first three items above, while the fourth – dealing with degraded input – is dealt with in a separate paper.

Discrete density HMMs are relatively straightforward to implement but research speech recognition systems no commonly use continuous density HMMs in order to provide more detailed models when more training data is available. There are several reasons that continuous density HMMs are preferable for this effort. First, we had observed that the accuracy of our discrete density HMM system did not improve significantly when we increased the amount of training data. We expect that the continuous density system would be able to benefit from larger training corpora. Second, there are several speaker adaptation algorithms that have been developed for continuous density HMMs. These algorithms can be used for adaptation to the variations in OCR: font, style, and quality. Finally, it was important to us that, wherever possible, we use the same software base for our OCR and speech recognition efforts.

The initial research OCR system was not practical in that it took a very long time (about 1/2 hour) to process a full page of text. This was, in part, due to not having paid attention to speed issues, and also because HMM systems perform a huge search over all possible character sequences with all possible segmentations. While the basic Viterbi beam search reduces the search space, the computation can be quite large. In addition, when we converted the system to use continuous density HMMs, the computation increased significantly because of the large number of Gaussian densities that needed to be evaluated. We streamlined the initial image processing stages and worked on techniques to speed up the recognition phase by a factor of about 50. As a result, the system takes between 30 and 45 seconds to process a full page of text. While this is not as fast as some systems, it makes the system usable.

We were concerned that it would be hard to extend the system to Chinese because of the large number of characters. We didn't know whether the system would be able to deal such detailed characters. Another concern was that a modest sized training corpus of real images would not cover all of the characters even once, which would present a problem for estimating models for new fonts. We also had no corpus of Chinese images and had to develop an inexpensive way of creating one. Our solution was to use sources for which we could find both images and online versions. This required developing techniques to automatically align the characters in the transcripts to the images.

We believe these extensions to the system are important in that they indicate that the system can be made practical and can, in fact, be used for many different languages. In section 2 we review the basic concepts in the HMM-based OCR system. Section 3 describes the results obtained in moving to continuous density HMMs. In Section 4, we describe the techniques that we used to speed up the system by a factor of 50. And Section 5 describes our work on Chinese OCR.

2. Review of OCR Using BYBLOS

We have previously described how we adapted our BBN BYBLOS continuous speech recognition system to the OCR problem. We review the basic techniques here very briefly. For a more complete discussion, see [3]. The fundamental idea, as illustrated in Figure 1, is to convert the two-dimensional image of a line of text into a one-dimensional sequence of feature vectors, so that we can treat the entire image as the output of the combined HMMs for the characters in the line. For each narro vertical slice of the image we compute features as a function of the vertical position. We also compute the derivatives of these features in both the vertical and horizontal directions. Finally we compute several local angles and correlation features to represent angular information. The result is a vector of 80 features for each slice or "frame". (We use the terms ***window, frame, slice,*** interchangeably. Also it is obvious that one can compute other features for each frame.) Once we have done this, we can use the BYBLOS speech recognition software without modification, since we can treat characters as if they were phonemes and words and sentences in text just like words and sentences in speech.

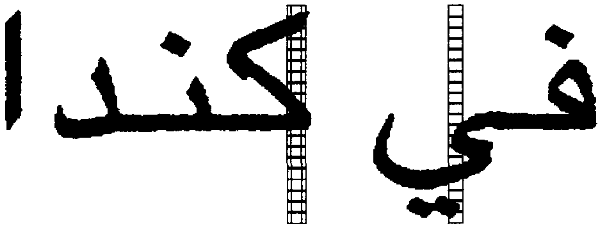


Figure 1: We divide a line of text into overlapping windows or 'frames'. We then compute a vector of features from the pixels within that window.

Figure 2 illustrates the overall BBN BYBLOS OCR system. (This block diagram is essentially the same for speech recognition as for OCR.) The system consists of two major parts: the training system and the recognition system.

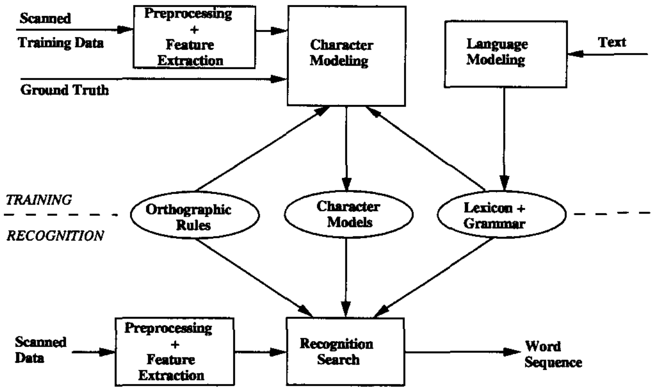


Figure 2: Block diagram of the OCR system.

The training system is provided with scanned images of lines of text and the matching text level transcriptions for those lines. The images are processed first by rotating the image appropriately and then by finding the locations of the lines. Then, features are extracted from each line image. In particular, we divide the line into narrow overlapping vertical slices of the image and extract a feature vector from each slice. The training system uses the forward-backward algorithm [8] to estimate the character models. A language model (prior model on character sequences) for recognition can also be estimated from the same text or a larger set of plain text. This can take the form of a probability distribution on likely character sequences, a lexicon of likely words, and a prior distribution on likely word sequences. The orthographic rules provide simple information about the language, such as the writing direction of the text lines.

In the recognition process, the image is preprocessed in the same way as during training. Then, the goal is to search for the most likely sequence of characters, given the extracted feature vectors from the scanned images, together with the different knowledge sources estimated in training, such as character models, lexicon, and grammar. We use a multi-pass fast search algorithm [9-12 instead of using the Viterbi algorithm because the Viterbi algorithm would be quite expensive when the state space includes a very large vocabulary and a bigram or trigra language model.

The goal of the system is to be language-independent in that it can easily be trained on matching images and text for almost an language and then used to recognize that language. The features extracted are not designed for any particular language and since the training does not require that the images be aligned at the character level, it is extremely easy to create a sufficient training corpus for a new language.

3. Continuous Density System

As stated above, we wanted to use continuous density HMMs for the OCR system for several reasons. In our original system, the models used were "tied-mixture models". First, we divided the entire input feature space into disjoint regions. Then, we defined a single gaussian distribution from the data within each region. The state distributions for all of the states of all of the characters are then estimated as a mixture of these gaussians, where the mixture weights are estimated using the Forward-Backward estimation algorithm. The advantage of this kind of system is that it is very easy to create and the computation is small because we only need to identify the likely gaussians for each input vector once. Although this system technically uses continuous (gaussian) densities, it functions as a discrete density system, since the set of gaussians is shared by all of the states.

By "continuous densities HMMs", we mean that we can use different sets of Gaussian distributions as the basis of the probability density associated with different states of the HMM. For example, we can define a distinct set of gaussians for each of the characters and this same set of gaussians is used for all of the states associated with that character. Thus, only the mixture weights vary from state to state within the character. In speech recognition, this type of HMM is called a "Phoneme-Tied Mixture" (PTM) HMM. Typically we use a set of 256 Gaussians for each character. (For speech recognition we often define systems that have more distinct sets of gaussians.)

We use the same multiple-step procedure for estimating these models that we use in our speech recognition system. First, we compute the feature vectors for a large number of images. We divide the feature space into 256 regions using a binar clustering and k-means process [13]. We define a gaussian for each region simply by computing its sample mean and diagonal variance. Then, we take this set of gaussians as the initial estimates of the gaussians for each of the characters. When we use the Forward-Backward algorithm to reestimate the distributions, the gaussians for each character diverge and the mixture weights for the different states are estimated.

While this provides a reasonable model, we find that a second round of estimation is beneficial. We use the model estimated above to determine the alignment of image frames to the states of the corresponding character HMMs. Then we choose a rando sample of the vectors for a character to initialize a new set of means for that character, which are estimated by going through all of the data several times. Finally, we use the Forward-Backward algorithm again to get final estimates of all of the model parameters. While this process sounds somewhat involved, it results in reliable estimates of the model parameters without the need for any manual alignment of any data. Thus, the human effort is minimized, which is one of our primar goals.

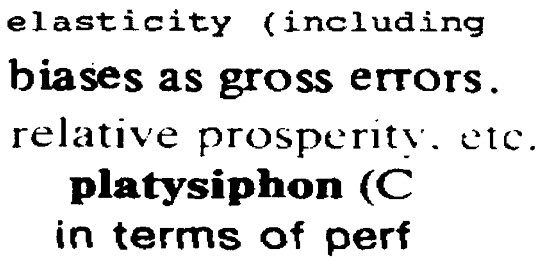
One difference between text images and speech is that text images typically have less variability. In particular, the background is often uniformly white, while in speech, "silence" is a continuous noise signal. Therefore, if we did nothing special, many of the estimated gaussian densities would have zero variance. To avoid this problem we simply added a small constant value to the estimated variances.

To reduce the dimensionality of the 80-dimensional input feature vector, we use linear discriminant analysis (LDA) [14] and choose the 15 features with the largest eigenvalues. This reduces computation and storage while improving the robustness of the estimates of the gaussian densities.

With the continuous-density HMMs, we have more parameters to describe the character models. We found that the continuous density system obtained the same performance as the discrete system when trained on small amounts of training data, but (in contrast with the discrete system) improved further for larger amounts of data. Also the continuous-density HMMs allow us to use the unsupervised adaptation techniques developed for speech recognition. Unsupervised adaptation is useful in dealing with degraded documents.

3.1 Improvement in Performance

One advantage of using the continuous-density system is that more training would improve the performance of the system. We can demonstrate the improvement in the performance of the continuous-density system with a test on English OCR. In the rest of this paper, we refer to the character error rate (CER), which is the total rate of the substitution, deletion, and insertion errors, as the accuracy measure for our OCR system. We used the University of Washington English Document Image Database I [7] to train and test our omnifont system. There are 958 image zones from books, journals, and magazines. However, in most of our experiments, we used a small subset of this data to train our models. A 90-character set was used. We used a character trigram prior language model, which was also estimated from the text of this corpus (excluding the test data). Figure 4 shows some samples from the corpus.



elasticity (including

biases as gross errors.

relative prosperity, etc.

platysiphon (C

in terms of perf

Figure 4: Sample images from UW Database I

We trained our omnifont English OCR system with either a training set of 100,000 characters or a training set of 600,000 characters from the UW Database and tested the system on a disjoint test set from the UW Database.

The CER for different training conditions and different models is shown in Table 1. A CER of 2.1% was obtained without a lexicon on the discrete-density system trained on the 100,000-character training set. No improvement was observed when we trained the discrete-density system on the 600,000-character training set. A CER of 2.1% was also obtained without a lexicon on the continuous-density system trained on the 100,000-character training set. But a big improvement was observed when we trained the continuous-density system on the 600,000 character training set where a CER of 1.2% was obtained. This is because the continuous-density system has the capability to accommodate more variations or degradations in it models.

System	Corpus Size	Char Error Rate (%)
Discrete-density	100K or 600K	2.1
Continuous-densit	100K	2.1
Continuous-densit	600K	1.2

Table1: Improvement in English OCR

4. Recognition Speed

In speech recognition, a system is considered fast if it operates in "real time", which is about 12 phonemes per second. Our speech recognition system uses 5-state HMM models for each phoneme, and the average number of observation vectors is about 8 per phoneme. In contrast, our BYBLOS OCR system uses 14-state models for each character, and the average number of observation vectors per character is about 25. So we could expect that the OCR system would run about 10 times slower or about one character per second. But OCR systems routinely run at 100 characters per second or more.

We increased the speed of the BYBLOS OCR syste dramatically by a combination of techniques. First, we integrated the various preprocessing steps, such as image rotation, linefinding, and feature extraction so that we avoided unnecessary I/O. Then we sped up the procedures for finding lines and for feature extraction by simple modifications. The bulk of the computation is taken up by the recognition search. The recognition search was sped up by a combination of three techniques: fast gaussian computation, two-pass search, and aggressive beam search pruning. Each of these will be described below.

4.1 Fast Gaussian Computation

We found that most of the computation in our continuous densit recognition system is in the Gaussian computation, i.e. finding the closest Gaussians to compute the output probability for the feature vector of each frame. Usually there are 256 Gaussians defined for each of the characters. We had developed a technique called "Fast Gaussian Computation" (FGC) to speed up the gaussian computation for the BYBLOS speech recognition system [15].

Our algorithm is a simplification of the algorithm presented in [16]. In the FGC method we first divide up the feature space into disjoint regions using a binary clustering algorithm. Typically, we might define 1024 regions. Given any feature vector, it is possible to determine the region that contains this vector using a binary search. Then, for each frame in the training set, we determine which region the frame is in, what gaussian was used, and which character the frame is part of. We then record in each region of the space, the list of all of the gaussians that were ever used for each of the characters.

During recognition, we first determine in which region the feature vector lies. This is done only once per frame. Then, when we consider a particular character, we only need consider those gaussians that have appeared within that region in the training data. Typically, the number of gaussians required is a small fraction of the full set. Thus the computation of gaussians is reduced by an order of magnitude without loss in performance. The entire system is sped up by a factor of three or more.

4.2 Two-pass Search

Our recognition search algorithm is a fast two-pass search, consisting of a coarse forward pass and a fine backward pass [10]. The forward pass computes the probability of each character (or word) ending at each frame in the input. The backward pass processes the line in the reverse direction using more detailed models. But whenever the search algorith transitions (backwards) to another character, it need onl consider those characters that were found to be likely to end at that frame of the input. The product of the forward and backward scores provide an ideal pruning score to minimize the computation.

The forward pass usually takes most of the time because it searches among all the character models while the backward pass searches only among the candidates selected by the forward pass. Since the purpose of the forward pass is only to find a list of likely characters, we find that it can be done using much less detailed models without any noticeable loss in the accuracy of the backwards pass. One useful feature developed for the BYBLOS speech recognition system is that the HMMs for phonemes (characters here for OCR) used in the two passes can be different. So we can use simple models in the forward pass to speed it up and use fine models in the backward pass to increase the accuracy. For example, we find that if we use 7-state discrete HMMs in the forward pass and 14-state HMMs in the backward pass, the computation in the forward pass is reduced dramatically, resulting in overall speed increase, while the accuracy is the same.

4.3 Pruning the Search

The goal of the recognition search is to find the character sequence that has the highest likelihood. The basic HMM search uses a time-synchronous "beam search" technique in which we discard any hypotheses that appear to be unlikely to result in the highest score. We find that we can often discard a large percentage of the hypotheses without any loss in accuracy. We found that our system could be tuned to discard a much larger number of hypotheses without significant loss.

4.4 Speed vs. Accuracy

Table 2 shows the results of the speed-up of a unifont Arabic OCR system. The speed-up factor of 50 was obtained with onl a loss of 0.17% absolute in CER. We obtained similar results in English. About half of the computation is taken up by the various preprocessing stages, and the other half is taken up by the recognition search. The resulting speed of about 45 characters per second is not extremely fast. But it is now within the range of usability and it is clear that it could be sped up by another small factor if necessary.

System	Speed (char/sec)	Char Error Rate (%)
Original	0.7	0.77
Fast	45	0.94

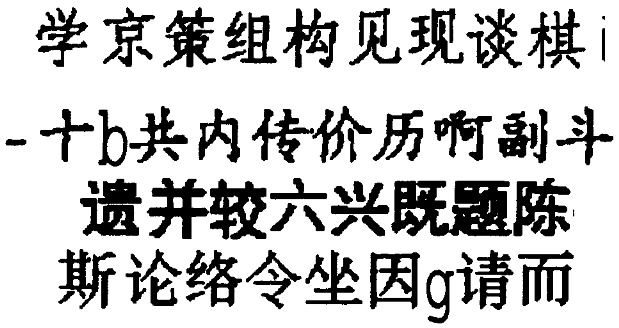
Table 2: Speed-up Results on an Arabic OCR syste

5. Chinese OCR

To further demonstrate the language independence of our approach, we extended our system to Chinese, which differs fro languages like Arabic and English in that the script does not have a small number of characters from which all words are constructed. Therefore, a training corpus, is unlikely to contain samples of all the characters that are expected in test data. Also, Chinese characters in general have very complicated structure, and it is not obvious from the outset that the simple models described in Section 2 are appropriate to model these complex characters.

5.1 Computer-Generated Chinese Corpus

In order to test the basic model on Chinese characters, we first collected a computer-generated corpus by printing and scanning images of all the 3,755 unique characters in GB2312-80 Level I of simplified Chinese and of 115 Roman/punctuation characters in 4 fonts (Fangsong, Hei, Kaishu, and Song). The size of the character set was 3,870. We printed and scanned 14 samples of each of the 3,870 characters in each of the four fonts. The characters occur in random sequences. Figure 5 shows a sample from the computer-generated corpus.



学京策组构见现谈棋i
-十b共内传价历啊副斗
遗并较六兴既题陈
斯论络令坐因g请而

Figure 5: A sample of the computer-generated Chinese corpus

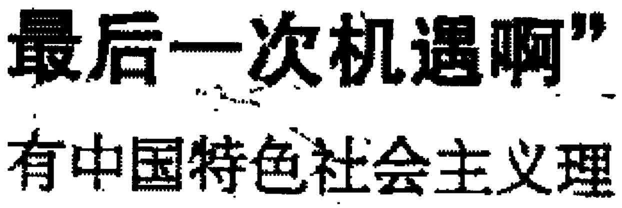
5.2 Unifont and Multifont Experiments

The first experiments were performed on the computer-generated corpus to test whether the HMM technology works for Chinese OCR at all. Each character was modeled by a 14-state HMM, just like for Arabic and English. No language model was used. In the unifont experiments, where the system was trained and tested on a single font at a time, the character error rate (CER) ranged from 0.1% to 0.4% for the four fonts, with an average of 0.3%. In the multifont experiment, where the system was trained and tested on a pool of data from the four fonts, the average error rate was 0.5%. Both of these results demonstrated that the complexity of the characters did not present any problem for the models we were using and the system could distinguish a large number of characters.

In order to get a sense of how different the four fonts were fro each other, we ran a cross-font experiment in which the syste was trained on three of the fonts and tested on the fourth. The result was a CER of 10%, indicating that the fonts are quite different from each other. This experiment suggested that in order to obtain good performance on real data, which is bound to look different from the computer-generated data, we would have to collect training data from actual printed sources.

5.3 OCR on Real Data

We did not have any corpus of real images with transcriptions for Chinese. So we needed to collect our own corpus. But we did not want to transcribe a large corpus, because this requires substantial effort – especially in Chinese where the training set might need to be larger and where the characters are harder to type. Therefore, we looked for sources where we could scan an image and could also find a text version. We found several newspapers, magazines, and books that we could buy that were also on line. We decided to use the Chinese newspaper People's Daily as our first source. We collected a real printed corpus of 60,000. The corpus has only 2,600 unique characters and is mainly a unifont corpus with some minority fonts. Figure 6 is a sample from this corpus. As can be seen, this data contains significant variability.



最后一次机遇啊"
有中国特色社会主义理

Figure 6: A sample from the printed Chinese corpus

However, the internet versions did not have linefeeds within a paragraph. So we developed a technique to determine the line-by-line transcription of the images from the paragraph transcriptions. The procedure starts by using an existing model of the characters (based on computer-generated images in this case) to recognize the characters in the images. Although the CER was quite high (about 10%), we found that we could align the recognized characters with the true characters in the online version automatically. We used the program that is normall used for aligning reference and hypothesis strings to compute error rates. Since we are only concerned with the location of the new line in the text version, we have no problem with substitution or insertion errors. The only uncertainty is when a character in the true text is deleted so we do not know which line of the image it is on. This only occurs on 5% of the lines, and we know that the problem has occurred so we can examine those lines or simply discard them from the training data. This procedure thus makes it even easier to create a training corpus for a new language, since we can determine the alignment of text to the lines of the image automatically.

For our experiments with real Chinese data, we used both the real and the computer-generated data for training but only real data (a disjoint set) for testing. From the real corpus we used just 16,000 characters for training, which covered only about 1600 unique characters. From the computer-generated data we used 3,870 unique characters, each having 14 training tokens in each of the four fonts. The test set contained 1225 unique characters, many of which did not appear in the real training data. Frequency weighted, 2% of the character samples in the test data had no training.

We found that the most effective way to combine the computer-generated training data and the real data was to simply train on both the data sets, with a larger weight on the real data. This way, any character with real data training is based mostly on the real data, while any character with no real training still has a model based on the computer-generated data. The CER results for several conditions are given in Tables 3 and 4.

Training Set	Language Model	CER
Computer	None	10%
Computer	Character trigra	7%
Computer + Real	Character trigra	1.2%

Table 3: CER for Chinese on newspaper data.

Number of samples of real training data	CER
0	5%
1 to 5	3%
>5	<1%
All characters	1.2%

Table 4: CER as a function of amount of real data for each character when trained on a mix of computer-generated and real data.

We can see that the use of a trigram language model on characters reduces the error rate from 10% to 7% when the character models are constructed entirely from computer-generated data. When we add the training from real data, we see that the CER drops to 1.2 %, because most of the character samples in the test have at least one sample of training. Table 4 gives a breakdown of the CER as a function of the amount of real training data for that character. We see that even characters with no training now have a CER of 5%. The improvement from 7% is due to the fact that the overall error rate is lower, and frequentl the recognition is helped by an adjacent character being correct. Once we have even a few training samples, the error rate decreases. And with more training it is lower still. It is interesting to note that, in general, the Chinese characters have very low error rates. The characters with the highest error rates were punctuation and English characters.

This result shows that it is possible to achieve a low error rate on Chinese character recognition of real images even though we did not have training data for all of the characters, and many of the characters had very little real training data.

6. Conclusions

In this paper, we presented several advances in the BBN BYBLOS OCR system, which can perform open-vocabular OCR on Arabic, Chinese, and English. The system is based on Hidden Markov Models and utilizes the same advanced technology that is used for speech recognition. Many of the changes were to incorporate the more advanced features of research speech recognition systems, including continuous-density HMMs, unsupervised adaptation, and fast search algorithms. We reported the improvement in Arabic and English OCR on the newly updated continuous-density system to sho that the continuous-density system improves significantly with a larger training corpus. We sped up the system by a factor of 50 without a significant loss in CER. The system is fast enough for practical use for Arabic and English OCR. Also we have extended the system to recognize Chinese with high accuracy on real newspaper data.

7. References

[1] J. Makhoul, R. Schwartz, C. LaPre, C. Raphael, and I. Bazzi, "Language-Independent and Segmentation-Free Techniques for Optical Character Recognition," Document Analysis Systems Workshop, Malvern, PA, pp. 99-114, October, 1996.

[2] R. Schwartz, C. LaPre, J. Makhoul, C. Raphael, and Y Zhao, "Language-Independent OCR Using a Continuous Speech Recognition System," Proc. Int. Conf. on Pattern Recognition, Vienna, Austria, pp. 99-103; August 1996.

[3] J. Makhoul, R. Schwartz, C. LaPre, I. Bazzi, Z. Lu, P. Natarajan, “A Language-Independent Methodology for OCR,” Proc. Symp. Document Image Understanding Technology (SDIUT97), Annapolis, MD, 1997.

[4] I. Bazzi, C. LaPre, R. Schwartz, and J, Makhoul, "Omnifont and unlimited vocabulary OCR system for English and Arabic," Proc. International Conference on Document Analysis and Recognition, Ulm, Germany, Vol. 2, 842-846, 1997.

[5] L. Nguyen, T. Anastasakos, F. Kubala, et al., "The 1994 BBN/BYBLOS Speech Recognition System", Proc. of ARPA Spoken Language Systems Technology Workshop, Austin, TX, Jan. 1995, pp. 77-81.

[6] R.B. Davidson and R.L. Hopley, "Arabic and Persian OCR training and test data sets," Proc. Symp. Document Image Understanding Technology (SDIUT97), Annapolis, MD, 303-307, 1997. R.B. Davidson and R.L. Hopley, "Arabic and Persian OCR training and test data sets," Proc. Symp. Document Image Understanding Technology (SDIUT97), Annapolis, MD, 303-307, 1997.

[7] I.T. Phillips, S. Chen, and R.M. Haralick, "CD-ROM document database standard," Proc. Int. Conf. Document Analysis and Recognition, Tsukuba City, Japan, pp. 478-483, Oct. 1993.

[8] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, Vol. 77, No. 2, pp. 257-286, Feb. 1989.

[9] S. Austin, R. Schwartz, P. Placeway, "The Forward-Backward Search Algorithm", Proc. of IEEE ICASSP-91, Toronto, Canada, May 1991, pp. 697-700.

[10 L. Nguyen, R. Schwartz, et al., "Search Algorithms for Software-Only Real-time Recognition", Proc. of ARPA Human Language Technology Workshop, Princeton, NJ, Mar. 1993, pp. 411-414.

[11 R. Schwartz, L. Nguyen, and J. Makhoul, "Multiple-Pass Search Strategies," in Automatic Speech and Speaker Recognition: Advanced Topics, C-H. Lee, F.K. Soong, K.K. Paliwal, Eds., Kluwer Academic Publishers, 429-456, 1996.

[12 L. Nguyen, R. Schwartz, "Efficient 2-Pass N-Best Decoder", Proc. EuroSpeech '97, Rhodes, Greece, Sept. 1997, pp. 167-170.

[13 J. Makhoul, S. Roucos, and H. Gish. "Vector Quantization in Speech Coding," Proc. of IEEE, Vol. 73, pp. 1551-1588, November, 1985.

[14 R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems, " Annals of Eugenics 7, 179-188, 1936.

[15 J. Davenport, R. Schwartz, L. Nguyen, "Towards a Robust Real-Time Decoder," Proc. of ICASSP '99, Phoenix AZ, March 1999, p. II-645.

[16 M. Padmanabhan, E. E. Jan, L. R. Bahl, M. Picheny, "Decision-tree based feature-space quantization for fast gaussian computation", Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, Dec. 1997, pp. 325-330.

Arabic OCR Systems: State of the Art

Tapas Kanungo, Osama Bulbul, Gregry Marton, and Doe-Wan Kim

Center for Automation Research
University of Maryland
College Park, MD 20742
Email: kanungo@cfar.umd.edu
Web: http://www.cfar.umd.edu/˜kanungo

Abstract

Characterizing the performance of Optical Character Recognition (OCR) systems is crucial for monitoring technical progress, predicting OCR performance, providing scientific explanations for system behavior and identifying open problems. In this article we report on the performance of nine Arabic OCR systems: Sakhr versions 3.01 and 5.0, Onset versions 2.0 and 3.0, two BBN algorithms, two Mitek algorithms, and one NDI-DOD algorithm. Sakhr and Onset are commercial products whereas NDI-DOD, BBN and Mitek are research algorithms.

The evaluation methodology was as follows. We used the SAIC Arabic dataset as our benchmark dataset. The dataset contains 344 scanned document images from newspapers, magazines, books and laser-printed articles. Each image is a single column text zone and for each image the corresponding groundtruth (manually typed correct symbolic text) is provided. The groundtruth is in CP1256 encoding.

The images were processed by the OCR algorithms and the output text files were compared with the corresponding groundtruth using a string matching algorithm. Several indicators of performance such as character and word accuracy, character and word precision, substitution/deletion/insertion errors, and average time taken were measured. A statistical method – the paired model – was then used to compare the accuracies of the OCR algorithms and compute statistical significance of the results.

Acknowledgements

We would like to thank Steve Dennis and Glenn Van Doren from the Department of Defense, and Melissa Holland, Ann Brodeen and Barbara Broome from the Army Research Laboratory for sponsoring this research. Coley Donovan provided us with the DOD error counting tools and the NDI-DOD OCR algorithm. Melissa Holland and Mohammad Shihada (Apptek) provided us with the Sakhr 3.01 algorithm and the Onset 3.0 algorithms. Gadi Mazor (Onset) provided us with the Onset 2.0 algorithm. Jim Campbell (Department of Defense) provided us with the Mitek algorithms. Zhidong Lu (BBN) provided us with the BBN algorithm results.

Arabic Text Recognition System

Andrew Gillies, Erik Erlandson, John Trenkle, Steve Schlosser
Nonlinear Dynamics Incorporated
123 N. Ashley Street, Suite 120
Ann Arbor, MI 48104

Abstract

This paper describes a system for the recognition of Arabic text in document images. The system is designed to perform well on low resolution and low quality document images. On a set of 138 page images digitized at 200x200 dpi the system achieved a 93% correct character recognition rate. On the same pages digitized at 100x200 dpi, the system achieved an 89% character recognition rate. The systems processes a typical page with simple layout and 45 lines of text in 90 seconds on a 400 Mhz Pentium II running Linux.

1. Background

The written form of the Arabic language presents many challenges to the OCR developer. The text is written right–to–left and uses a script alphabet in which consecutive letters within a word are joined to one another by a baseline. In order to accommodate the baseline, the Arabic alphabet has four forms of each letter: an isolated form, an initial form, a medial form, and a final form. Several letters in the alphabet disobey this convention and are missing their medial and final forms. When one of these non–joining characters is encountered within a word, the preceding letter assumes it's final (or isolated) form, and the non–joiner assumes its initial (or isolated) form.

Arabic text contains a large number of special forms, called ligatures, which replace particular character pairs or even triples. For example, when the LAM character is followed by the ALEF character they will almost always be combined into a single ligature character called the LAM–ALEF. While use of the LAM–ALEF ligature is almost universal, most ligatures are optional, at the discretion of the typographer. We have encountered over 200 ligatures in our development effort, although many of these are extremely rare, occurring mainly in older typeset books.

Arabic text is often justified so that the right and left edges of the text column are aligned. In a Roman alphabetic setting this would be accomplished by stretching the spaces between words to fill out the desired length. In Arabic, portions of the baseline are stretched. These stretched baselines, called kashidas, occur in different words throughout the line.

Finally, although classic Arabic texts use a relatively limited number of font faces, new typographic systems have led to the proliferation of Arabic font faces which are almost as varied as those for Roman alphabets.

Because of these, and related challenges, the state of the art in Arabic OCR significantly lags that for Roman text. Our laboratory is committed to producing an Arabic OCR which will be usable in a variety of settings. Of particular importance is the ability of the system to operate effectively on low resolution and low quality images. While there is still much room for improvement, our current system is beginning to meet our project goals.

2. System Description

The system described in this paper is a complete Arabic page reader implemented in the UNIX environment. The system takes in document pages as binary TIFF images, and produces Unicode text files as output. A block diagram of the system is shown in Figure 1. This section describes the operation of the four modules shown in Figure 1.

Page Decomposition

The raw page image arrives at the page decomposition module. The module starts with a preprocessing step to remove graphic elements and clean up the page. It then decomposes the page into text blocks.

Each text block is segmented into individual lines of Arabic text. The text line images are normalized to a height of 40 pixels and passed to the text segmentation module.

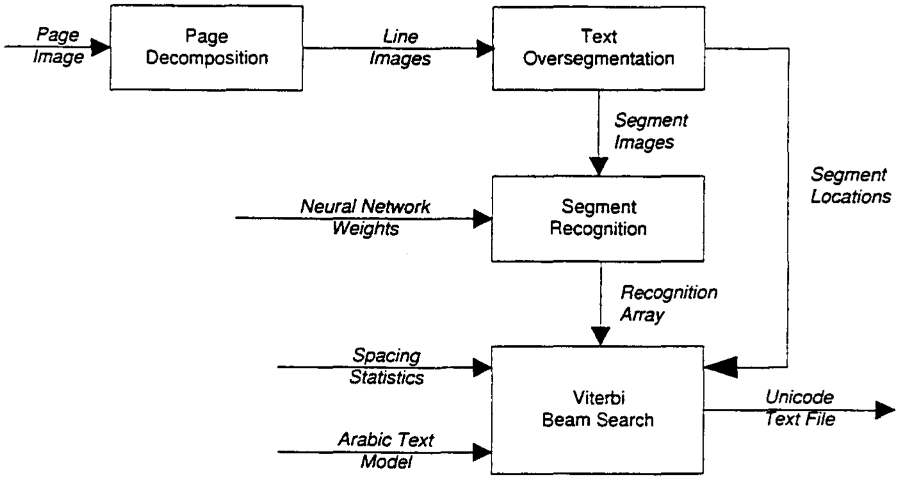


Figure 1. Arabic Text Recognition System

Text Segmentation

The text segmentation module is illustrated in Figure 2. This module is an oversegmenter, aiming to produce atomic image segments which are no larger than a single character. In other words, each atomic segment should come from only a single character of the ideal segmentation. If this goal is met, then the ideal segmentation can be produced by combining the atomic segments in the appropriate groups. Of course, the combination must be done in such a way as to maintain the spatial relationships between the atomic segments in the group. The Viterbi algorithm, discussed later in this paper, produces the appropriate groups as a by–product of the recognition process.

The text segmentation module begins by forming the 8–way connected components of the text image. Each component is analyzed to see if it needs splitting. The splitting algorithm chooses a set of split points which divide a component into a number of non–overlapping pieces, the union of which equals the initial component.

The split–points are chosen heuristically using two methods. The first method suggests split points which occur at locations where the objective function f(x) achieves a local minimum. (Note: y–coordinates increase downward from the top of the image)

$f(x) = MAX(B-Top(x), 0) + MAX(Bottom(x)-B, 0)$

Top(x) = y–coordinate of the topmost pixel in column x of the component

Bottom(x) = y–coordinate of the bottommost pixel in column x of the component

B = y–coordinate of the nominal baseline in the normalized line image

The second splitting method looks for local minima in the y–coordinate while tracing the top half of the contour around the component. Unlike Top(x), this method can trace parts of the stroke which are in the "shadow" from an overhanging part of the stroke. The two methods often produce identical or near identical split point suggestions. In this case, the redundant split points are discarded. In addition, if a split produces a very small atomic segment (less than 20 pixels in area, and less than 4 pixels wide) that split point is discarded.

The result is a list of image components called *atomic segments*. The atomic segments are ordered in a left to right fashion based on the center of the minimum bounding box enclosing the segment. The atomic segments are combined in groups of from two to five consecutive atomic segments. The complete set of segments includes both the atomic segments and the combined segments. This results in $K < 5N$ segment images, where N is the number of atomic segments. Among the K segments are the ideal characters of the text, indicated by dashed boxes in Figure 2. The K segment images are passed to the segment recognition module.

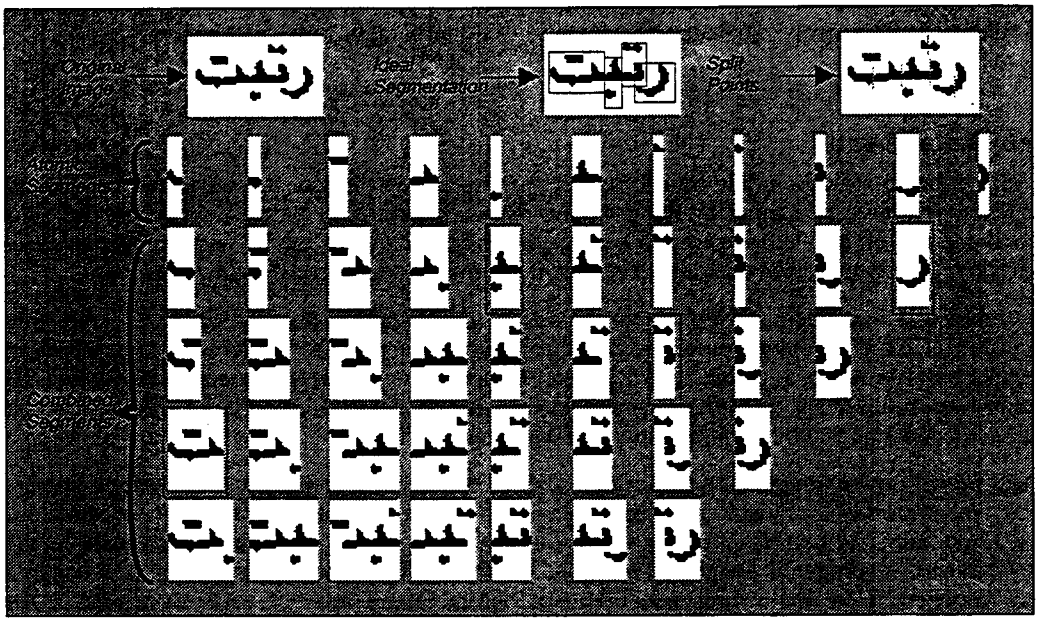


Figure 2. – Text Segmentation

In addition to the segment images, the segmentation module produces information about the spacing between segments. This takes the form of a gap size array which measures the width, in pixels, of the gap between consecutive atomic segments. The gap width can be positive, zero, or negative, for kerned segments. In computing the gap between two atomic segments numbered *x–1* and *x* we consider all segments with index less than *x* as belonging to one super segment, and all segments with index greater than or equal to *x* as belonging to another super segment. The gap is defined as the distance between the rightmost edge of the left super segment and the leftmost edge of the right super component. The gap function used in the Viterbi algorithm, described below, is defined as:

gap(x) = the gap size in pixels between atomic segments x–1 and x.

Segment Recognition

The segment recognition module runs a neural network classifier on each of the *K* input segments. The neural net has 424 input cells, two hidden layers with 60 and 90 cells respectively, and an output layer with 229 cells. The net has a total of 51,450 weights.

The 424 element input–layer feature vector includes 40 projection features and 384 chaincode features. The projection features contain a 20 element horizontal projection and a 20 element vertical projection of the segment image. The projections are taken with respect to a 40 x 40 pixel box in which the segment image is left–justified, and hence each count represents two rows or columns of the image. Also, if the segment is less than 40 pixels wide, the vertical projection is padded with zeros on the right.

The chaincode features are based on a chain–like representation of the edges (or borders, or contours) of the image components before splitting. Each point on the chain can be uniquely associated with one of the atomic segments, and hence with each of the combined segments. At each point along the chain, the direction of travel (clockwise around the component) and curvature of the contour are computed. These measurements use a window of 4 chain points on each side of the point in question. The 384 chaincode features correspond to quantizing the x–coordinate, y–coordinate, direction, and curvature values into 4, 4, 8, and 3 bins, respectively. For each dimension, quantization is gaussian weighted between neighboring bins so that a point in the center of a bin contributes a weight of 1.0 to that bin, and a point on the border between two bins contributes 0.5 to each of the two bins. Thus, each chaincode point contributes to up to 2x2x2x2 = 16 feature values. The chaincode features are sparse, in that many of the values are zero for a typical image.

The output layer of the neural network contains 229 cells corresponding to 117 regular Arabic character forms, 80 ligature forms, 10 Arabic digits, 20 punctuation characters, and two reject classes. The outputs of the neural network for each of the segment images are combined into a N x 5 x 229 element array, called the recognition array. This array is passed to the Viterbi algorithm for decoding.

Viterbi Beam Search

The Viterbi beam search module transforms the information in the recognition array (and gap array) into a sequence of characters output by the program. The module uses a dynamic programming algorithm to match the array of segments against a model of Arabic text. The model encodes the rules of Arabic typography, for example the constraints between the forms of neighboring characters.

The Arabic text model comprises lexicon based word recognition, lexicon free word recognition, and recognition of Arabic punctuation and digits. The basic element of the model is a state, which ultimately associates a given image segment with a given character (or ligature). The complete Arabic text model contains over 100,000 states, most of which occur inside the lexicon based word recognition component. The lexicon contains 50,000 high frequency Arabic words.

The text model can be thought of as a directed graph where the nodes represent recognition states, and the arcs represent allowable transitions between states. There are two kinds of states: *real states* and *virtual states*. Each real state corresponds to a particular Arabic glyph — a character or ligature in a particular form. Thus each real state is associated with one of the 227 (229 net outputs – 2 reject classes) character classes recognized by the neural network. The virtual states are not associated with a recognized character, but instead are used to simplify the graph structure. In addition, a virtual state may encode a *space* between words in the text.

The text model can be thought of as a process for generating text. Any path through the graph corresponds to a sequence of characters emitted from the real states along the path, and spaces emitted from the space–encoding virtual states. Given a recognition array and gap array as input, the Viterbi algorithm computes the highest payoff path through the graph. The Unicode file is generated by outputting the emitted characters corresponding to this path.

The Viterbi algorithm computes a two–dimensional array $V(s,x)$ where s ranges over the states of the text model, and x ranges over -1 to $N-1$, where N is the number of atomic segments in the text image. The value $V(s,x)$ corresponds to the accumulated payoff for the best path to state s using atomic segments 0 through x. The V matrix is computed by an inductive formula:

initialization: $V(start,-1) = 0.0$ *where: start is a designated start state (virtual) of the graph*

induction: $V(s,x) = \underset{(s'x')}{MAX}\,[V(s',x')+(x-x')(R(s,x'+1,x)+S(s,x))]$

where:

s' ranges over the predecessors of state s

x' ranges over $x-5$ to $x-1$, covering characters comprised of from 1 to 5 atomic segments

$R(s,x'+1,x)$ = the neural net output for class s for the segment running from $x'+1$ to x

$S(s,x)$ = an adjustment based on the gap between segments $x-1$ and x (see below)

The term $(x - x')$ in the induction formula weights each contribution to the V matrix by the number of atomic segments in the segment being recognized. This has the effect of normalizing the Viterbi values in each column of the matrix.

The $S(s,x)$ adjustment is designed to take into account the gaps between segments in the recognition of the text. For this purpose, each real state is categorized into one of three space cases, based on its relation to the previous real state in the graph. The three cases are:

1) Touching: the two model characters are joined by a baseline. This case occurs only between pairs of joining Arabic characters within a word.
2) Within–word–gap: the two model characters are separate but without an explicit space character between them. This case occurs for the non–joining Arabic characters within a word, and also for

punctuation marks which can follow a word without an explicit space.

3) Between–word–gap: the two model characters are separated by an actual space character.

The $S(s,x)$ value is based on a set of statistics collected on these three cases. (See Figure 3.) Using training text with models of each line of text based on truth files we compute $P(g \mid c)$ where g is the gap width in pixels and c is the case of state s (1, 2, or 3). We can now write

$$S(s,x) = \alpha * \log P(gap(x) \mid case(s)) + \beta \qquad where: \quad \alpha = 0.5 \quad \beta = 0.067$$

The two parameters, α and β control the importance of the spacing information relative to the neural net recognition score. They were empirically set using a small subset of the training data.

The Viterbi module uses a beam search to avoid computing all values of the V matrix. The beam search works by eliminating any elements $V(s,x)$ in column x of the matrix whose values are less than $M(x) - \omega$ where $M(x)$ is the maximum value in column x and ω is the beamwidth parameter. In our system ω is empirically set to *1.0*. When elements in the V matrix are eliminated, all graph edges from those links are also removed from consideration. In this way, the beam search prunes the state graph as it proceeds along the columns of V. This procedure has resulted in a significant speedup of the Viterbi module, since in practice considerably less than 5% of the graph states typically become active at any given column of the V matrix.

Once the V matrix computation is complete, the final output is generated by a traceback of the maximal payoff path though V. This is accomplished easily, because the forward Viterbi step keeps a traceback pointer for every element in the V matrix. The traceback pointer for $V(s,x)$ points to the element $V(s',x')$ which maximized the expression in the induction formula for V. Traceback starts from the maximal element in the last column of V, and proceeds backward to the beginning of V. The Unicode characters corresponding to this sequence of states are written to the output file.

3. Training Procedures

Much of the work in the development of the Arabic page reader occurs in training the segment recognizer. The input to the training process is a set of page images together with truth files which contain an ASCII (Arabic modified) representation of the text on those pages. This section describes the training steps which ultimately produce the weight matrix of the recognizer's neural network. The process also produces the weight statistics used in the Viterbi module. The processing follows the diagram shown in Figure 3.

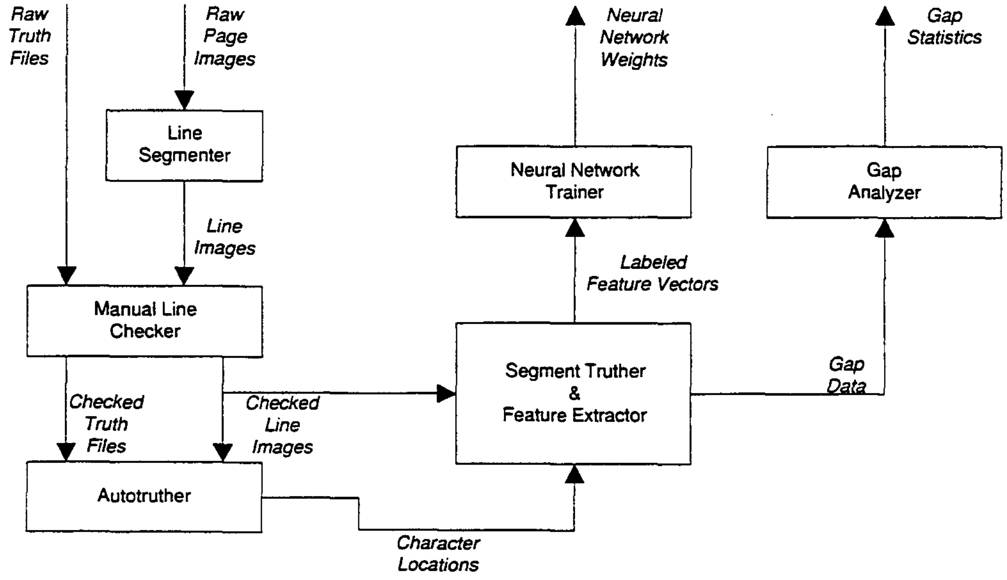


Figure 3. – Truthing Block Diagram

Line Segmenter

The first step in the training process is the line segmenter, which extracts individual lines of text from the page images. The line images are normalized to 40 pixels in height. Since the training pages are constrained to have a simple page layout, the segmenter can be rather simple. Nevertheless, the segmenter does make some errors.

Manual Line Checker

The output of the line segmenter, and the truth files are fed to a viewing program where the operator checks to make sure each image line is aligned with the correct truth line. Lines which are incorrectly aligned are discarded. The result is line images correctly paired with the truth text of the line.

Autotruther

The autotruther's job is to align the truth text with the image in such a way that the location and identity of each character in the image is known. The process uses a discrete distribution, left–to–right hidden Markov model (HMM) of the type described in [Rabiner 89] to accomplish this task. Note [2], also in this volume.

The autotruther creates a special model for each line in the truth file. This model encompasses all of the possible realizations of this text as a sequence of character glyphs. So, for example, when a pair of characters which can form a ligature is encountered, the model has a branch for the ligature glyph, and a branch for the pair of simple character glyphs. Each glyph model has from five to fifteen Markov states.

The autotruther uses a feature extractor which assigns a 96 element vector to each column in the line image. Each element of the vector is associated with a character stroke contour at a given orientation, curvature and y–location in a 5 pixel window around the column of interest. The vectors are then mapped into an integer in the range 0 to 99 by a vector quantization process. The vector quantization codebook is produced by a simple k–means clustering step on a sample of feature vectors extracted from line images. The result of feature extraction and vector quantization is a representation of the line image as a sequence of integers. The HMM is trained using the sequence/model pairs. Initial model parameters are estimated from a small amount of previously aligned character data.

Once the HMM is trained, it reprocesses the sequence/model pairs to arrive at the ultimate assignment of model states to columns in the image. Note that the HMM not only locates each character, but also chooses between alternative glyph representations of the truth text. The output of the HMM is a character location file which gives the glyph number, left x–coordinate and right x–coordinate for each character in line image.

Segment Truther & Feature Extractor

The character location files are used by the segment truther to assign truth to each *segment* produced from a training line image. The module starts by producing the segments in a manner identical to the online system. (See *Text Segmentation* section above.) It then compares the location of each segment with the character locations produced by the HMM. Using a dynamic programming procedure, the best alignment of segments to characters is produced. The segments which are aligned with HMM character locations are labeled as such, provided they meet some sanity requirements. The majority of segments, of course, do not correspond to single characters, but rather correspond to parts of one or more characters. The characters are labeled as either *small rejects*, meaning they contain only parts of a single ideal character, or *large rejects*, meaning they contain parts from two or more ideal characters. Again, sanity checks are performed to ensure that the classes assigned to each segment are known with high confidence. Segments which do not pass the sanity checks are eliminated from the training set.

At this point, the segment truther holds segment images with known character or reject classes. It then performs feature extraction on these segment images and writes the result into class labeled feature vector files. These files are used for neural network training.

The segment truther also records information about the gaps, or lack of same, between pairs of characters in the image. The result is a labeled gap file. Each line in the file contains the identity of the left and right hand glyph, the width of the gap between them, and a flag which indicates whether the gap corresponds to a space between words in the truth file. (See *Text Segmentation* section above for a description of gap width measurement.)

Neural Network Trainer

The neural network trainer takes the labeled feature vectors as input, and uses a modified backpropagation algorithm to generate the network weights. For the training sets used in this project, the modified algorithm gave a significant increase in segment recognition accuracy, from 86% using standard backprop to 91% using the modified version. The training time (number of iterations to convergence) was also reduced by a factor of six using the modified algorithm. Finally, the second choice capabilities of the network were significantly enhanced. That is, in the event that the correct class for a segment was the second choice of the network, the height of its activation value above the background was 50% greater for the network trained using the modified algorithm.

The modification to backpropagation which achieved these results is quite simple. The standard backpropagation algorithm sets targets, usually 0.1 and 0.9, for the cells in the output layer of the network. The cell representing the correct class receives a 0.9 target and all other cells receive a 0.1 target. The backpropagation algorithm computes a delta score which is the signed distance between the cell output and its target. A nonzero delta results in changing the weights so as to move the cell output in the direction of the target value. In the case of a cell for the true class producing a value greater than 0.9, this results in a movement of the weights to *reduce* the output of this cell. Similarly, when a cell representing an alternate class correctly produces an output less than 0.1, the standard algorithm tends to *increase* the output for this cell.

The modified backpropagation algorithm was inspired by the observation that the output vector of a *classification* neural network is not used as a point in N–space, but rather it is interpreted such that the cell with the highest activation represents the class associated with the input vector. With this in mind, the modified backpropagation algorithm sets the delta for the truth–cell to *zero* when that cell's activation exceeds 0.9, and sets the delta for any non–truth–cell to zero when that cell's activation falls below 0.1. The effect of this is to remove unnecessary constraints on the training process. For example,if the neural network has learned to set the truth–cell activation *higher* than 0.9 for some training vector, then there is no need to constrain the weights to set the truth–cell activation lower if it sees that input vector in the future. Similarly, non–truth–cell activations lower than 0.1 are also satisfactory. Removing these constraints on the backpropagation training results in faster training and higher recognition performance, as described above.

Gap Analyzer

The gap analyzer produces a set of statistics used by the Viterbi module to account for character spacing in the interpretation of text images. Each gap listed in the labeled gap file is assigned to one of three categories based on the glyphs on either side of it. (See the Viterbi Beam Search section for an explanation of these three classes.) The gap analyzer estimates P(g | c), the probability of observing a gap of a width g, given class c. The estimation is done by a simple histogram. A plot of P(g | c) is shown for each of the three gap classes in Figure 3.

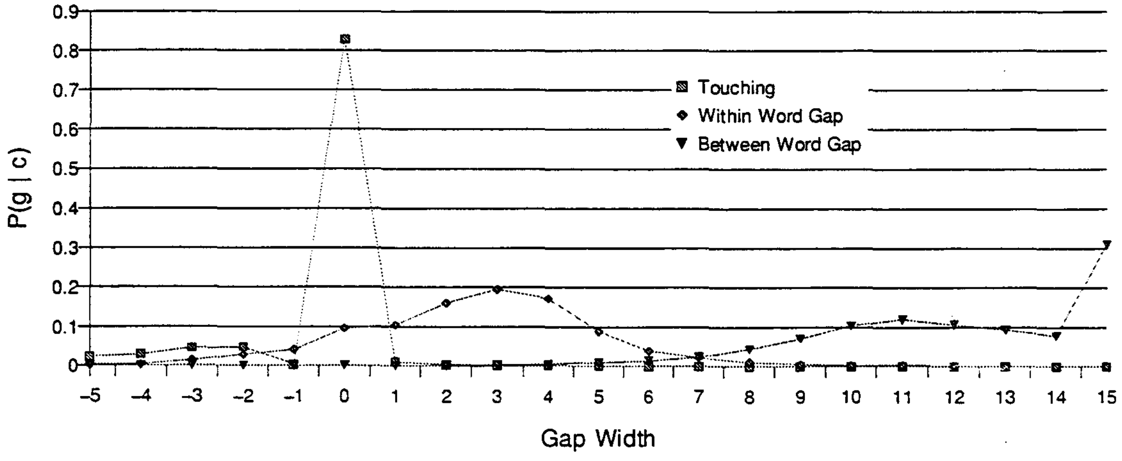


Figure 4. – Gap Statistics

4. Results

The page images used for training came from a data collection performed by SAIC. These consisted of 344 pages imaged at 600dpi. These images led to three datasets: Set1) the original images imaged at 600dpi, Set2) the images printed on paper at 600dpi and then re–imaged thorough scanner #1 at 200dpi, and Set3) the same printed pages imaged through scanner #2 at 200dpi. The choice to use two scanners was motivated by the desire to capture multiple image noise characteristics. A summary of the dataset sizes is shown in Table 1.

Dataset	Pages	Lines	Words	Characters
Set1	269	11,110	89,623	424,001
Set2	268	11,055	88,872	420,565
Set3	266	10,843	86,714	410,083
Total	803	33,008	265,209	1,254,649

Table 1. – Training Dataset Sizes

The system was tested on a total of 40 page images. The results are shown in Table 2. The test data included 20 pages digitized to 200x200 dpi, and the same 20 pages digitized to 200x100dpi. The results indicate a drop in performance due to the lowered resolution.

Dataset	Resolution	Pages	Lines	Words	Characters	Character Rate
Set2	200x200 dp	10	442	3495	17165	92.9%
Set3	200x200 dp	10	447	3689	18129	93.2%
Total	200x200 dp	20	889	7184	35294	93.1%
Set2a	200x100 dp	10	442	3495	17165	90.0%
Set3a	200x100 dp	10	447	3689	18129	88.6%
Total	200x100 dp	20	889	7184	35294	89.3%

Table 2. – Test Results

5. Conclusions

The paper addresses some of the technical details of the recognition algorithm. These include the combination of neural network outputs and spacing statistics into pseudo–probability scores, and some variations on the backpropagation algorithm used to train the neural networks.

6. References

[1] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" *Proc. IEEE 77*, 257–286, (1989).

[2] A. Gillies, R. Krug, "Automatic Groundtruth Assignment for Text Image Databases", in this Volume.

Abstracts

Multi-Modal Information Access

Francine Chen
Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304
fchen@parc.xerox.com

Abstract

At Xerox PARC, we are investigating methods for combining richer, "multi-modal" features to help users satisfy their information need. At one end of the spectrum, this involves identifying documents or passages of interest, providing simple, rapid access to information pertinent to a user's needs. At the other end, this involves examining a collection of documents to identify underlying structure and relationships.

In our models we envision that analogous to a database with various fields, the documents and passages in the collection are characterized by many different features, such as user information, analysis of text content to determine genre, or analysis of images. One of the difficulties in employing a heterogeneous set of multi-modal features is assigning weights to the importance of different features. In traditional systems that employ metadata, the metadata usually has finite, discrete values and a boolean system that includes or excludes particular values is used to identify data elements of interest. Extending the concept to multi-modal features that may not be discrete leads the question of how to combine the features. We have been developing two different methods of combining multi-modal features, each tailored to the application: 1) providing different views of a collection through multi-modal clustering and 2) helping a user to identify documents of interest in a system called multi-modal scatter/gather. In the first application, the features are considered simultaneously and the task is organization and analysis of a document collection. In the second application, one feature is considered at a time, as specified by a user in a browsing and retrieval task. In particular, a user specifies different multi-modal features to progressively narrow a collection to a small subset of interesting collection objects. Individual features then can be used to expand the set to include similar objects, some of which may have incomplete feature sets.

Currently our work has been applied to web documents, for which metadata is more easily derived than imaged documents. In the future, we would like to apply our techniques to imaged documents.

We are also developing content-based characterizations of documents that can be included in the set of multi-modal features. An example of this is the identification of the genre of a document based on textual cues[2]. Genre identification gives a user the ability to find documents of a particular type, in addition to topic. For example, a person may be interested in *technical articles* about lupus or *editorials* about the supercollider.

A paper on multi-modal scatter/gather[1] presented at the SPIE Document Recognition and Retrieval conference in Jan 1999 follows.

References

[1] Francine Chen, Ullas Gargi, Les Niles, Hinrich Schuetze "Multi-Modal Browsing of Images in Web Documents," Proceedings of SPIE Document Recognition and Retrieval VI, Daniel P. Lopresti and Jiangying Zhou, Editors, Vol. 3651, pp. 122-133, 1999.

[2] Brett Kessler, Geoffrey Nunberg, Hinrich Schuetze, "Automatic Detection of Text Genre," Proceedings ACL/EACL, Madrid, p. 32-38, 1997.

Multi-Modal Browsing of Images in Web Documents

Francine Chen[a] Ullas Gargi[b] Les Niles[a] Hinrich Schütze[a]

[a]Xerox Palo Alto Research Center, 3333 Coyote Hill Rd, Palo Alto, CA USA

[b]Dept. of Computer Science and Eng., Penn State University, University Park, PA USA

ABSTRACT

In this paper, we describe a system for performing browsing and retrieval on a collection of web images and associated text on an HTML page. Browsing is combined with retrieval to help a user locate interesting portions of the corpus, without the need to formulate a query well matched to the corpus. Multi-modal information, in the form of text surrounding an image and some simple image features, is used in this process. Using the system, a user progressively narrows a collection to a small number of elements of interest, similar to the Scatter/Gather system[1] developed for text browsing. We have extended the Scatter/Gather method to use multi-modal features. With the use of multiple features, some collection elements may have unknown or undefined values for some features; we present a method for incorporating these elements into the result set. This method also provides a way to handle the case when a search is narrowed to a part of the space near a boundary between two clusters. A number of examples illustrating our system are provided.

Keywords: multi-modal information access, image/document browsing and retrieval, clustering, web documents

1. INTRODUCTION

Much of the research in information retrieval has focused on retrieving text documents based on their textual content or on retrieving image documents based on their visual features. Recently, there has been some research on the use multi-modal features for retrieval.[2,11,12] We are investigating an approach to document browsing and retrieval in which a user iteratively narrows their search using both the image and text associated with the image, and possibly other types of information related to the document, such as usage. We refer to disparate types of information such as text, image features and usage as modalities.

In this paper, we present a method of information access to a collection of web images and associated text on an HTML page. Our method permits the use of multi-modal information, such as text and image features, for performing browsing and retrieval of images and their associated documents or document regions. In our approach, we use text features derived from the text surrounding or associated with an image, which often provides an indication of its content, together with image features. The novelty of our approach lies in the way it makes text and image features transparent to users, enabling them to successively narrow down their search to the images of interest. This is particularly useful when a user has difficulty in formulating a query well matched to the corpus, especially when working with an unfamiliar or heterogeneous corpus, such as the web, where the vocabulary used in the corpus or the image descriptors are unknown.

Our work can be thought of as an extension to image browsing. An ideal image browsing system would allow a user to browse images that may or may not have descriptive annotative text and use both text or image features. Users may wish to browse through image collections based either on their semantic content ("what does the image show?") or their visual content ("what does the image look like?"). Image retrieval systems are often based on manual keyword annotation or on matching of image features, since automatically annotating images with semantic information is currently an impossible task. Even so, a manually labeled image collection cannot include all the possible semantic significances that an image might have.

Current image retrieval systems commonly display a random selection of images (e.g., QBIC,[3] Virage[14]) or allow an initial text query as a starting point (e.g, QBIC,[3] Smith and Chang[11]). In the latter case, a set of images with that associated text is returned. The user selects the image most similar to what they are looking for, a search using the selected image as the query is performed and the most similar images are displayed. This process is repeated as the user finds images closer to what is desired. In some systems, the user can directly specify image features such as color distribution and can also specify weights on different features, such as color histograms, texture, and shape.[3] In web pages, text such as URLs may also provide clues to the content of the image. Current image retrieval technology also allows the use of URL, alt tags, and hyperlink text to index images on the web (e.g., Dunlop[2]; Smith[11]). One approach also attempts to determine for each word surrounding an image caption whether it is likely to be a caption word[8] and then match caption words to "visual foci" or regions of images (such as the foreground).[4] The Webseek image search engine[11] and MARS-2[10] allow for relevance feedback on images by marking them as positive or negative exemplars.

Using multi-modal features, our system permits quick initial focusing of the set of elements of interest, and then organization and expansion to include similar elements, some of which may have incomplete feature sets or occur in another cluster. One difficulty in the use of multiple features in search and browsing is combining information from the different features. This is commonly handled in image retrieval tasks by having weights associated with each feature that can be set by the user. In contrast to current image search systems, in our method of browsing and retrieval, a user employs different multi-modal features to progressively narrow a collection to a small subset of images of interest, with associated text, *without* weighting the different features. Each feature is used one at a time to either refine or enlarge the set of images. The image features are used independently of text features to create multiple clusterings in the different modalities that the user can navigate, using text (e.g., section headings, body text, abstract, title, "alt" tags in image anchors) and image features to refine the images in the collection.

Although the use of clustering in image retrieval is not new, it has usually been used for pre–processing, either to aid a human during the database population stage,[7] or to cluster the images offline so that distance searches during queries are performed within clusters.[5] In our work, we use iterative clustering and selection of cluster subsets to help a user identify images of interest. Clustering is used for interactive searching and presentation, and relevance feedback is implicit in the user's choice of clusters. Because the user is dealing with clusters, not individual images, the feedback step is also easier to perform. Our work is most similar to Scatter/Gather which was developed by Cutting et al.[1] for text browsing. Scatter/Gather iteratively refines a search by "scattering" a collection into a small number of clusters, and then a user "gathers" clusters of interest for "scattering" again. We have extended the Scatter/Gather paradigm to multiple modalities and have added an "expand" function so that elements from outside the working set can be incorporated into the working set.

In practice, an initial text query is used to find candidate images of interest. Some of the returned clusters containing images of interest are then identified by the user for further consideration. By expanding based on similarity of one image feature, the system then finds and presents image clusters that are similar to those represented by the initially selected clusters, but without associated text or with text not similar enough to the user-specified query. Thus the expand function permits relevant images that are absent in the original set as a result of the text query to be identified and included. The expand function can also identify for consideration elements that are near the feature space of interest, but that are — due to the partitioning at an earlier step — in another cluster.

2. CLUSTERING AND GATHERING SUBCOLLECTIONS

A preprocessing step is used to precompute information needed during browsing and to provide the initial organization of the data. A set of features, possibly from different modalities, is precomputed for each document image and stored as vectors. The text features include the words of text surrounding and associated with each image, the URL of the image, alt tags, and hyperlink text. The image features include a color histogram and a measure of color complexity. The documents are initially clustered into groups based on the text features.

To search for images, a user begins by entering a text query. A hypothetical session is illustrated in Fig. 1 where: a node represents the data in a cluster; the solid arrows represent the scattering or gathering of data in a node; and the dashed lines represent movement of a subset of data in a node to another node, as in the expand function. The precomputed text clusters are ranked by similarity to the query terms using the cosine distance and the highest ranking clusters are displayed by representative text (see Fig. 1a). The user then selects the clusters that are most similar to their interest. This may include all or a subset of clusters (see Fig. 1b). One of two operations is then performed: 1) The elements in the selected clusters are reclustered based on a selected feature (see Fig. 1c) or 2) The selected clusters are expanded to new similar (dashed lines in Fig. 1d) clusters based on a selected feature. The new clusters are displayed as representative text or images, depending on whether the selected feature is derived from text or image data. The selected feature may be any of the precomputed features. By reclustering, the user can refine the set of images. By expanding, images similar in the specified feature, possibly with missing values in other features, can be brought into the set of images for consideration.

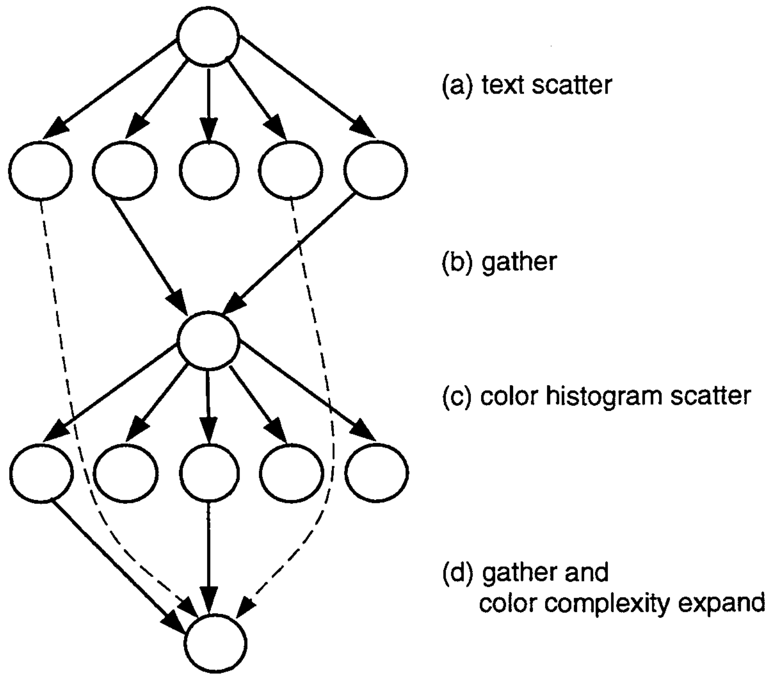


Figure 1. Hypothetical session of scattering and gathering collection elements in different modalities.

Clustering is performed using a standard k-means clustering algorithm with a preset number of clusters. In the preprocessing step, the number of clusters is larger than the number of clusters presented to the user. This is because only a subset of clusters will be presented in response to the initial text string query. In our case with an initial text query, we cluster to 20 clusters and return the 5 most similar clusters. The clusters selected by the user for gathering are then reclustered, where the number of clusters is equal to the number of clusters to be displayed, again 5 in our case. Each subsequent gather and scatter operation results in 5 clusters. As each operation is performed, cluster results are stored. This permits backing up and is also needed by the expand operation.

3. EXPANDING SUBCOLLECTIONS

The expand function addresses a problem with progressively narrowing a search based on different features: images with missing values will be eliminated from consideration. For example, some documents contain images with no associated text, or text unrelated to the contents of the image. In other cases, the text surrounding the image has no relevance to the semantic content of the image. Another problem with progressively narrowing a search is that the search may be narrowed to a part of the space near a boundary between two clusters.

The expand operation adds images or clusters to the current set, based on similarity in one feature dimension. Because only one feature is examined at a time, the distance metric can be different for each feature. For example, we use the cosine distance for text similarity and the normalized histogram intersection as an indication of histogram similarity. The expand operation is performed in one of two ways. The first method insures that the elements of the current clusters remain and the set size is increased by adding to the current working set some elements that are close to the working set based on the selected feature. The mean of the selected feature for the current working set is computed, and then those elements, represented as vectors, in the entire database that are closest to this mean are added. This is most appropriate for text features. A second method is to add elements that are close to each displayed representative in the working set. This is more applicable to the image features, in which the clusters are represented by selected images instead of a compilation of the elements used to represent text. However, if the text is represented by selected documents, this method of expansion would also be appropriate.

Referring to the example in Fig. 1, expansion is performed by identifying the most similar clusters based on the color complexity feature. In this way, images with no relevant text are identified if they are similar, in this case based on the complexity feature, to images with relevant associated text. For example, the terms in some URLs are not informative (e.g. the terms in the URL http://parcweb/project/anyproject/pics/fig1.tif are: parcweb, project, anyproject, pics, fig1 and tif). By first identifying images that are associated with terms of interest and then expanding to images similar in another feature, such as the color complexity feature, a larger number of images can be identified without starting the search over or requiring the use of weights.

4. FEATURES AND DISTANCE METRICS

The system currently uses three simple features. Two of the features are image-based; the third is text-based. We chose these features because we wanted simple, understandable features that would illustrate our method for combining image and text modalities in information access. Because each feature is used separately, the most suitable distance metric can be applied to each feature. In the future, we would like to enlarge the set of features to include features that other researchers have found most useful, such as the use of local color histograms for different image regions, segmentation and texture features.

4.1. Text Feature

The text feature is a *term vector*, where the elements of the vector represent terms used to represent "documents" and the terms are derived from text surrounding an image, image URL, and page URL. Currently, we limit the scope of the surrounding text to 800 characters preceeding or following the image location. If a horizontal rule, heading or another image occurs prior to the limit being reached, the scope ends at the rule, heading or image. A stop-list is used to prevent indexing of common terms with little content, such as prepositions and conjunctions. Currently, the terms from the different sources are combined into one term vector. An alternative would have been to separate the terms from the different text sources. A single term vector was used in part because the amount of text associated with an image is fairly small, in comparison to the amount of text in normal documents. The vector similarity is computed using the cosine distance:

$$d(t_1, t_2) = \frac{\sum_i t_1(i) t_2(i)}{\sqrt{\sum_i t_1^2(i) \sum_j t_2^2(j)}},$$

where t_1 and t_2 represent the term vectors from the two documents for which the similarity is to be computed.

4.2. Color Histogram Feature

A single color histogram is used as the color feature. The feature space is converted to HSV, and two bits are assigned to each dimension. The histogram is normalized so the bin values sum to one for each image. The distance between histograms is computed similarly to the *intersection measure* by Swain and Ballard,[13] but with normalization by the largest bin value:

$$1.0 - \frac{\sum_i \min(h_1(i), h_2(i))}{\sum_i \max(h_1(i), h_2(i))},$$

where h_1 and h_2 represent the normalized color histograms for the two images. Thus the distance is symmetric with respect to the two images. A symmetric distance is needed in our framework because we are computing distances between an image and another image or centroid for clustering purposes, rather than retrieval purposes.

4.3. Complexity Feature

The complexity feature attempts to capture a coarse semantic distinction that humans might make between images: that between simple logos and cartoons at the one extreme, which are composed of a relatively small number of colors with regions of high color homogeneity, and outdoor photographs on the other, which are composed of a relatively large number of colors with fine shading. The feature is derived from the run-length of the colors. In particular, run-lengths of the "same" color are identified in the x and y directions. A histogram is computed for each direction, where the bins represent the percentage of the image width or image height a run spans in the x or y direction, respectively. The count in each bin is the number of pixels in the image belonging to that particular run-length. Another way to interpret this is that the value added to a bin for each run is weighted by the length of the run, giving greater weight to longer runs. The total number of elements in a histogram is the number of pixels in the image. With the distance metric used, there is no need to normalize the sum of the bins.

The distance metric between two vectors, v_1 and v_2, is the average of the similarity between each pair of histograms:

$$d(v_1, v_2) = .5\frac{\sum_i x_1(i)x_2(i)}{\sqrt{\sum_i x_1^2(i)\sum_j x_2^2(j)}} + .5\frac{\sum_i y_1(i)y_2(i)}{\sqrt{\sum_i y_1^2(i)\sum_j y_2^2(j)}},$$

where the similarity is computed using the cosine distance, x_1 and x_2 represent the x-run-length histograms, and y_1 and y_2 represent the y-run-length histograms for the two images.

5. REPRESENTING CLUSTERS

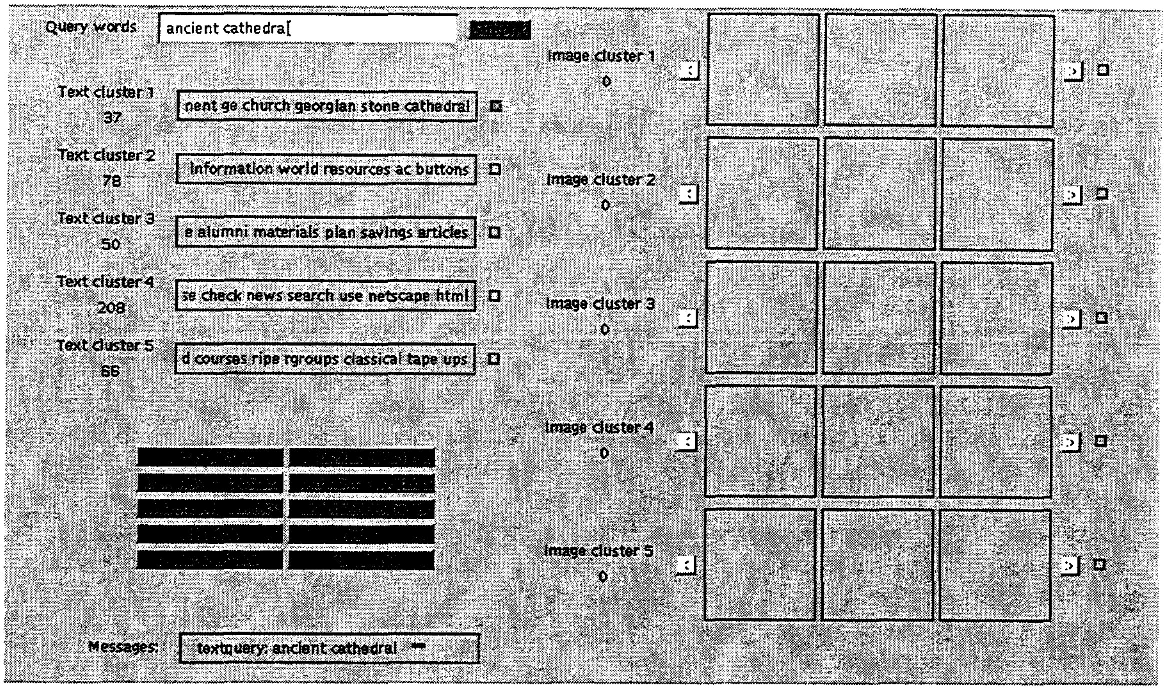
When using a clustering scheme such as Scatter/Gather, it is necessary to display or represent the clusters to the human user during a browsing session. A text cluster can be represented in a number of ways, the most common being the selection of a set of words that are in some way most representative of the cluster, and displaying them. In our work, clusters based on text features are represented by high frequency content words. When image clusters need to be so represented, it is less meaningful to choose image features that are common to the cluster members and display them, since these will not, in general, have semantic meaning to the user. Some systems display a collection of images in a two dimensional space using multi-dimensional scaling (e.g., Rubner *et al.*,[9] Marks *et al.*[6]). To display the clusters more quickly, we select a small number of representatives from each cluster and display only those representatives. The representatives are comprised of: 1) the three images closest to the centroid of the cluster and 2) three images representative of subregions of the cluster. The three subregion representatives are computed by removing the three most central images, computing three subclusters, and using the image closest to the centroid of each subcluster. This representation provides a sense of the cluster centroid and the range of images in the cluster. The representative images could also have been placed on a 2-D display using multi-dimensional scaling, but for the examples in this paper, we display the representatives in a row of three "centroid" images or three "subcluster" images (e.g., see Fig. 4). This permits very similar images, such as thumbnails and multiple copies of originals, to be more readily identified.

6. EXAMPLES

In our current work, we have used a collection of web documents containing 2310 images as our corpus. Web documents contain many of the same types of "meta-information" that can be found in scanned images of documents and can be used to infer the content of a document or the components in a document. By working with web documents, the issues involved with identifying components and layout in an image are minimized, while permitting development of techniques for using meta-data in the retrieval process. In the future, we would like to extend this work to collections of scanned documents.

To prevent the corpus from being dominated by "uninteresting" images such as logos and icons that are so ubiquitous on the Web, we applied some simple, and somewhat arbitrary, criteria that images must satisfy to be included in the corpus. (Note that it was not necessary, nor a goal of this work, to include *all* images of any particular class, only to assemble an interesting corpus from what's available on the Web, so we intentionally set a high reject threshold.) An image was required to have height and width of at least 50 pixels, and to contain at least 10,000 total pixels. An image was also required to pass some color-content-based tests: that no more than 90% of the image be composed of as few as 8 colors, no more than 95% of the image be composed of as few as 16 colors, and that the RGB colorspace covariance matrix of the image's pixels be non-singular. Qualitatively, these criteria ensure that the images are not simple line drawings, and contain enough variety of color content to be well-differentiable by the color features described above. We did not screen for multiple versions of the "same" image, so the corpus does contain identical images, as well as an image and a thumbnail of the image.

We present three sample sessions illustrating the use of "scattering" and "gathering" in different modalities. The first example illustrates the use of the text feature to first narrow the collection and then use of an image feature to organize the results. The user starts by typing in the text query "ancient cathedral". A snapshot of the screen displaying five returned text clusters is shown in the left half of Fig. 2. These clusters are the clusters closest to the query terms. The most frequent content terms in each cluster are displayed to represent each cluster. The user can scroll each text window to view additional representative terms for a text cluster. The user decides to scatter the first text cluster containing the terms "ancient" and "cathedral" again based on text. A snapshot of the screen



Query words: ancient cathedral[

Text cluster 1 37: nent ge church georgian stone cathedral ■

Text cluster 2 78: information world resources ac buttons □

Text cluster 3 50: e alumni materials plan savings articles □

Text cluster 4 208: se check news search use netscape html □

Text cluster 5 66: d courses ripe rgroups classical tape ups □

Image cluster 1 0

Image cluster 2 0

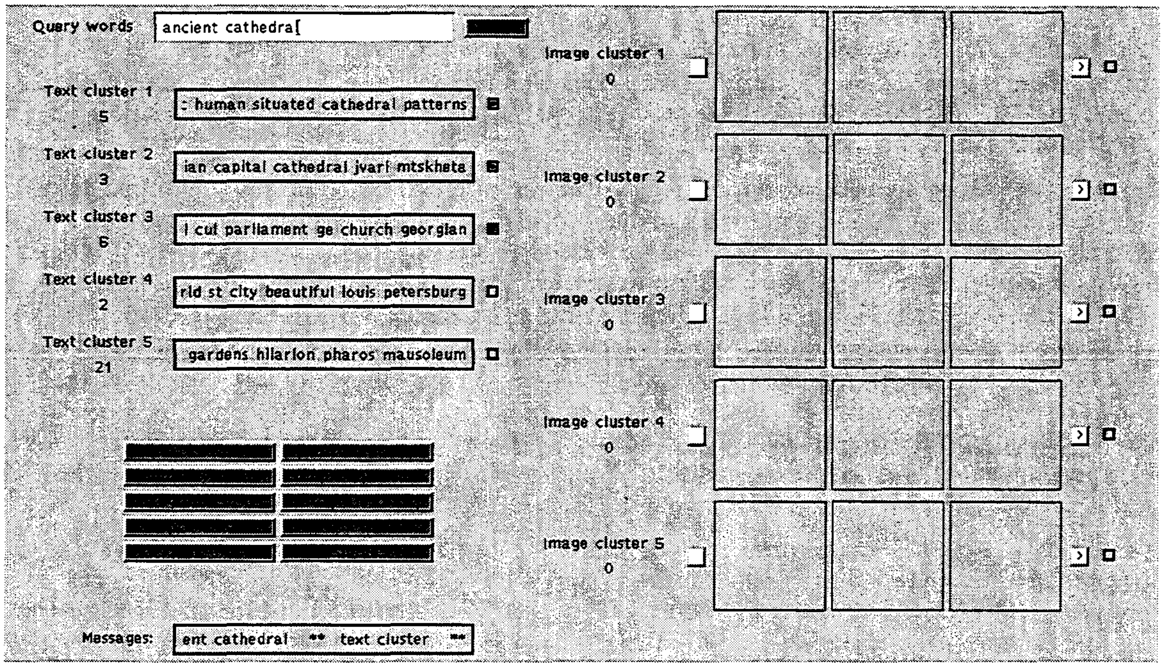
Image cluster 3 0

Image cluster 4 0

Image cluster 5 0

Messages: textquery: ancient cathedral **

Figure 2. Text clusters returned in response to the query "ancient cathedral".



Query words: ancient cathedral[

Text cluster 1 5: : human situated cathedral patterns ■

Text cluster 2 3: ian capital cathedral jvari mtskheta ■

Text cluster 3 6: l cul parliament ge church georgian ■

Text cluster 4 2: rld st city beautiful louis petersburg □

Text cluster 5 21: gardens hilarion pharos mausoleum □

Image cluster 1 0

Image cluster 2 0

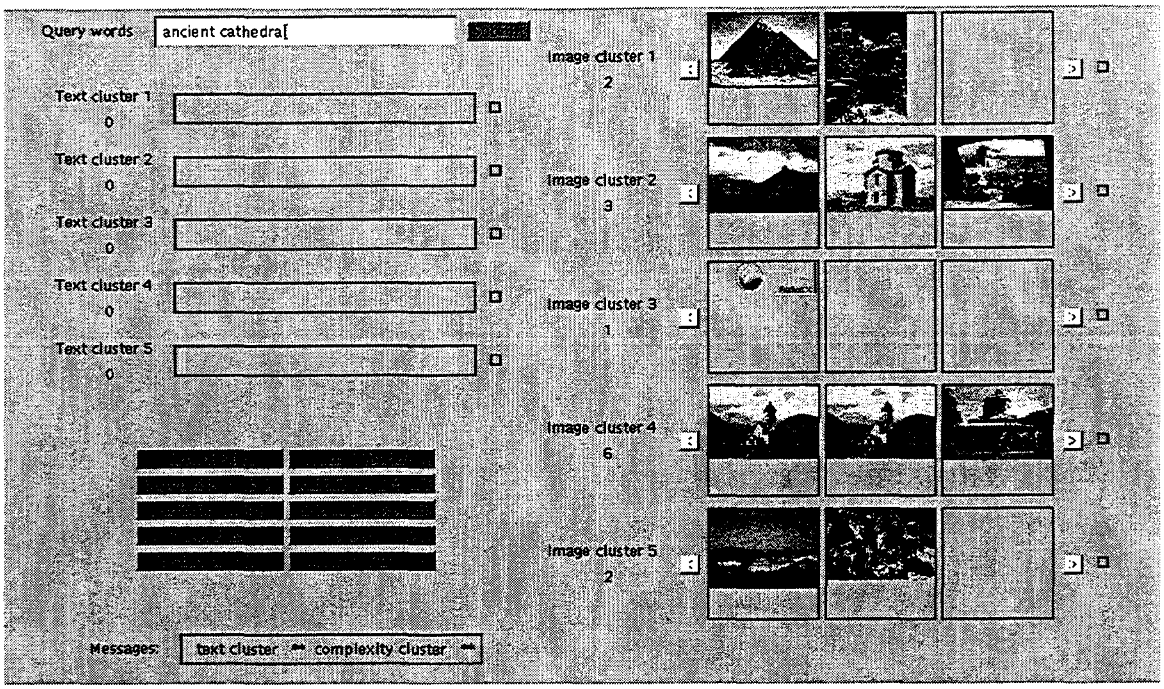
Image cluster 3 0

Image cluster 4 0

Image cluster 5 0

Messages: ent cathedral ** text cluster **

Figure 3. Text clusters returned after scattering Text cluster 1 in Fig. 2



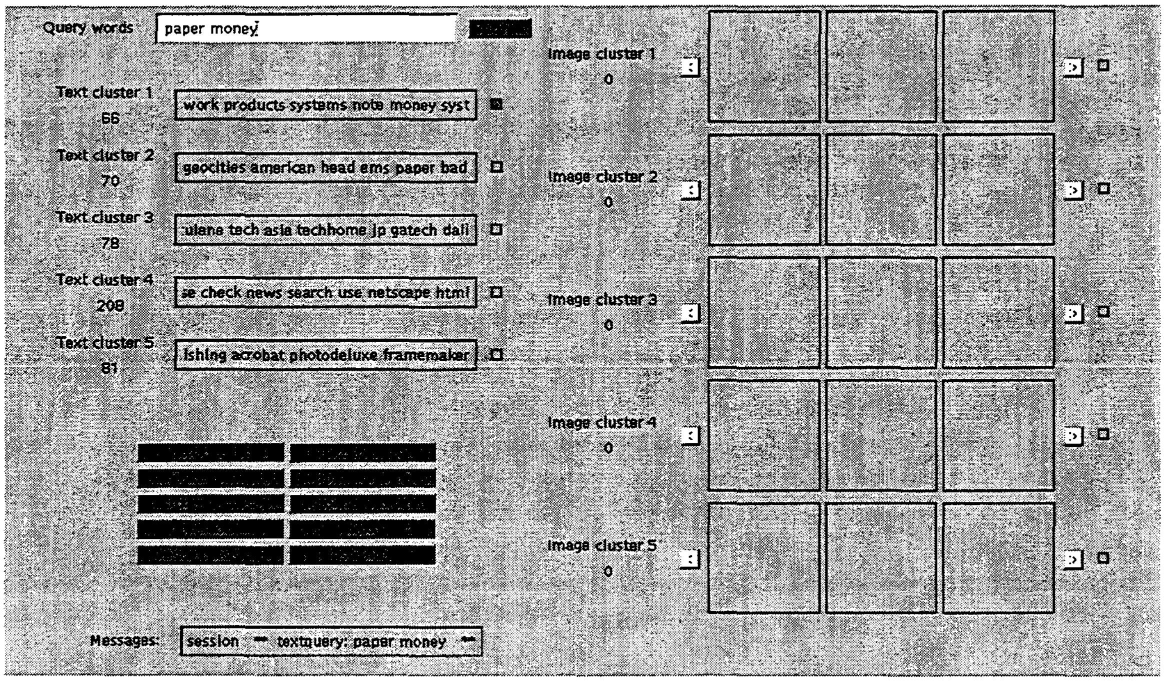
Query words: ancient cathedra[

Text cluster 1: 0 □
Text cluster 2: 0 □
Text cluster 3: 0 □
Text cluster 4: 0 □
Text cluster 5: 0 □

Image cluster 1: 2 □
Image cluster 2: 3 □
Image cluster 3: 1 □
Image cluster 4: 6 □
Image cluster 5: 2 □

Messages: text cluster → complexity cluster →

Figure 4. Image clusters returned after clustering based on the complexity feature.



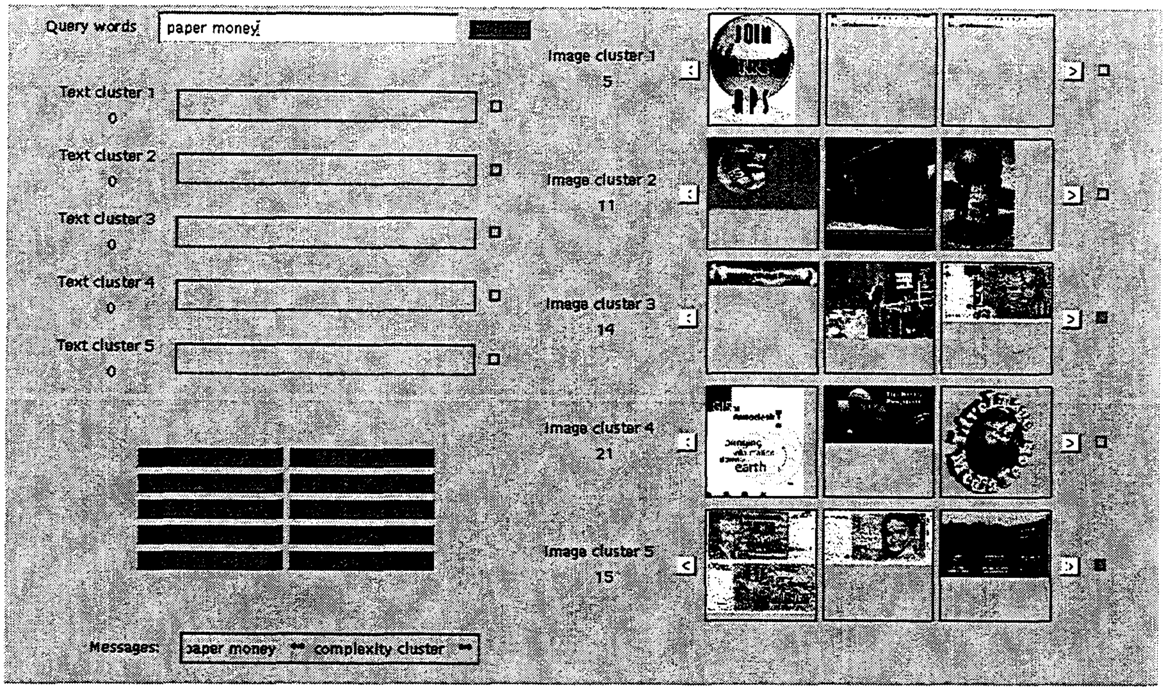
Query words: paper money

Text cluster 1: 66 — work products systems note money syst ■
Text cluster 2: 70 — geocities american head ems paper bad □
Text cluster 3: 78 — ulene tech asia techhome jp gatech dali □
Text cluster 4: 208 — se check news search use netscape html □
Text cluster 5: 81 — ishing acrobat photodeluxe framemaker □

Image cluster 1: 0 □
Image cluster 2: 0 □
Image cluster 3: 0 □
Image cluster 4: 0 □
Image cluster 5: 0 □

Messages: session → textquery: paper money →

Figure 5. Text clusters returned in response to the query "paper money".



Query words: paper money

Text cluster 1: 0 □
Text cluster 2: 0 □
Text cluster 3: 0 □
Text cluster 4: 0 □
Text cluster 5: 0 □

Image cluster 1: 5
Image cluster 2: 11
Image cluster 3: 14
Image cluster 4: 21
Image cluster 5: 15

Messages: paper money ** complexity cluster **

Figure 6. Image clusters returned after clustering Text cluster 1 in Fig. 5 based on the complexity feature.

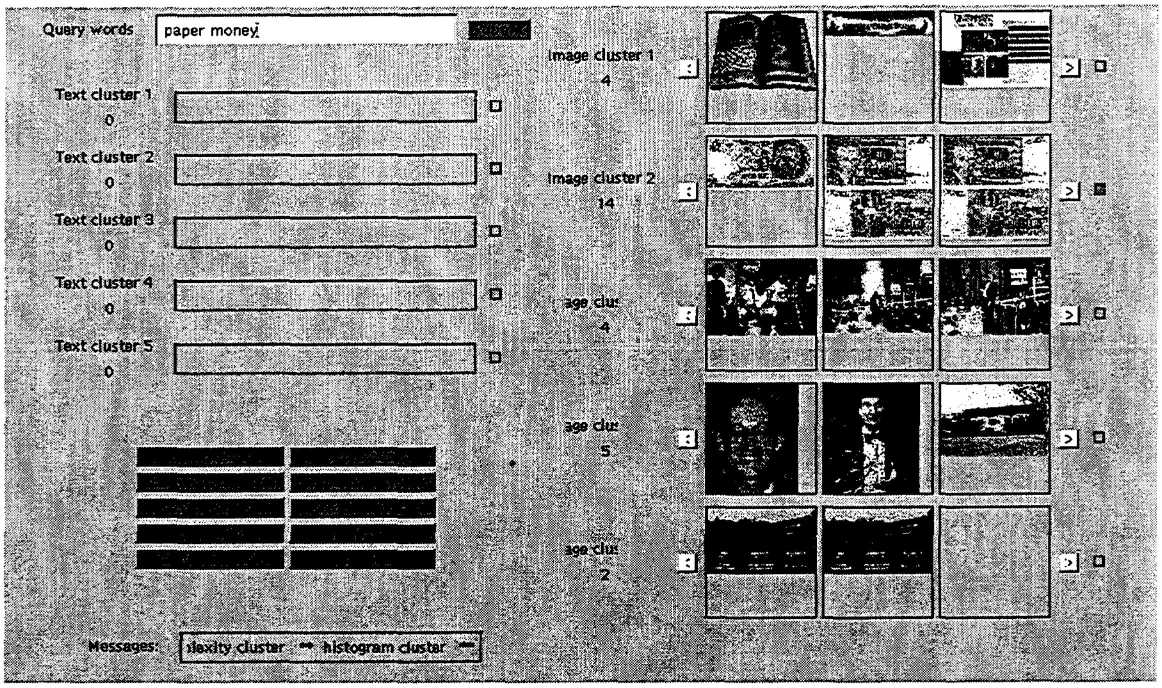
displaying the five resulting text clusters is shown on the left half of Fig. 3. The user selects the three clusters that contain the terms "ancient," "cathedral" and "church" to gather and selects complexity as the feature for scattering.

A snapshot of the screen after clustering based on the image complexity is shown in Fig. 4. The representative images closest to the centroid are displayed. By clicking on the arrows next to each image cluster, the user can move between the centroid and subcluster representative views. Image clusters 1, 2 and 4 contain images primarily of "ancient" buildings and monuments, including old churches and cathedrals. Image cluster 3 contains a logo and Image cluster 5 appears to contain miscellaneous items.

In the second example, our hypothetical user is trying to find a number of images of paper money in our corpus. An initial query of "paper money" is given and the text clusters are shown in Fig. 5. The first text cluster contains the word "money" as well as the word "note". This cluster looks promising so the user selects it. Text cluster 2 contains the word "paper", but the surrounding words do not indicate that the desired sense of the word paper is being used, so this cluster is not selected. Since money is printed in many colors, the color complexity measure is more appropriate to use initially as an image feature. Text cluster 1 is scattered based on the color complexity feature and the clusters are shown in Fig. 6. Image clusters 3 and 5 contain images of paper money, so they are gathered and then scattered based on the color histogram feature this time. The resulting image clusters are shown in Fig. 7. Image cluster 2 contains 14 images, and the central representatives are all images of paper money. This cluster is scattered again based on the histogram feature and we note that it contains many images of paper money, as shown in Fig. 8. Some of the images appear to be duplicates, but in this case they are actually a thumbnail and the full-size image. Examination of the sub-cluster representatives reveals some images in the subclusters that do not contain money, but which have similar colors to the money images.

This example illustrates the use of different features in serial combination to selectively narrow the set of images to a set of interest. Scattering is used to help organize a larger collection into smaller subsets. Gathering permits different collections to be combined and reorganized together.

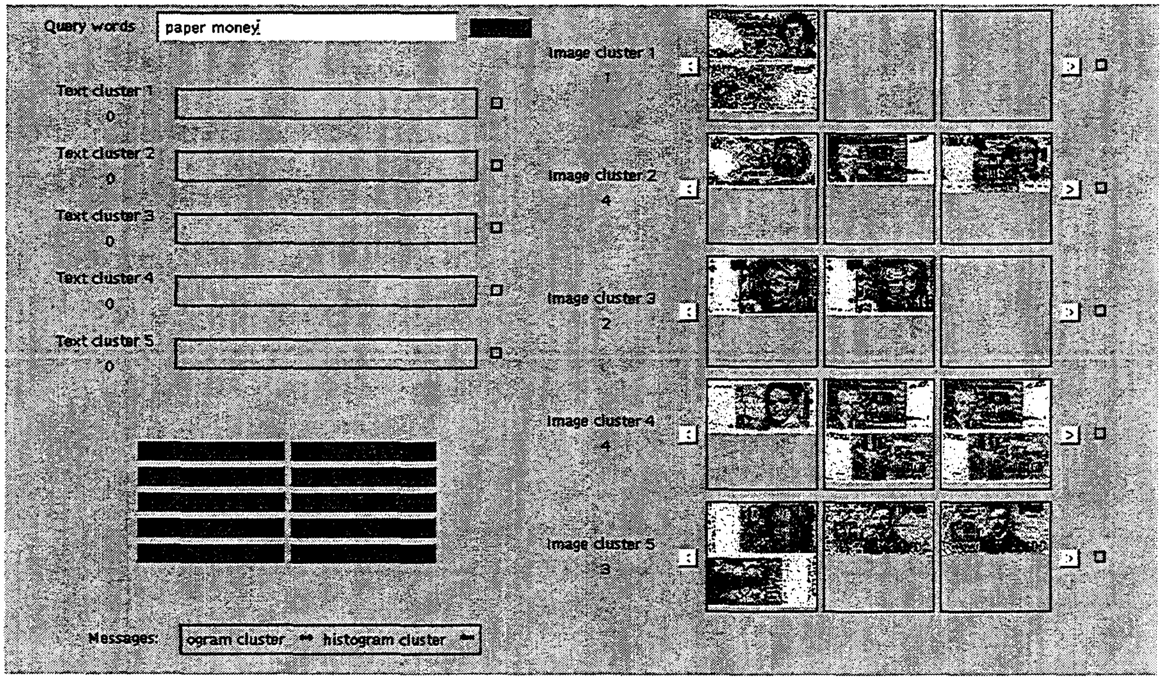
In the final example, the user is searching for pyramids and types in the query "pyramid egypt". The returned text clusters are shown in Fig. 9. The user selects the first text cluster to be scattered based on the complexity feature, and representative images from the resulting image clusters are shown in Fig. 10. The user notes that there are outdoor scenes with stone in image clusters 2 and 4 and selects those for further clustering based on the color



Query words: paper money

Messages: lexity cluster ⇒ histogram cluster ⇒

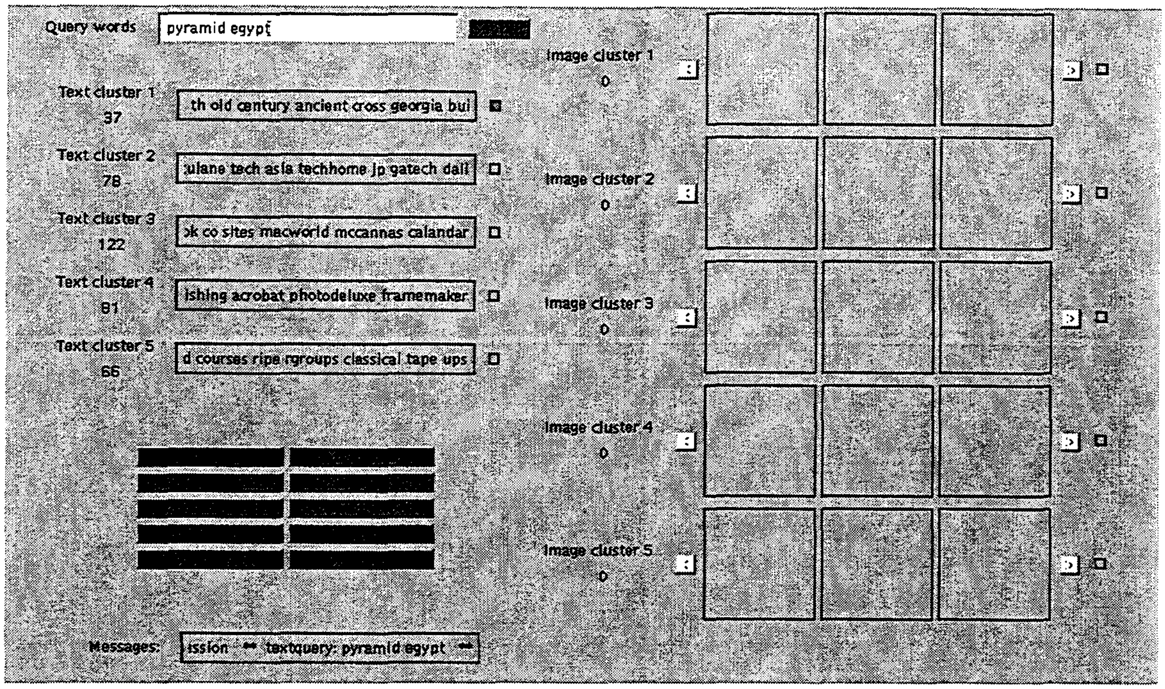
Figure 7. Image clusters returned after clustering Image clusters 3 and 5 in Fig. 6 based on the color histogram feature.



Query words: paper money

Messages: ogram cluster ⇒ histogram cluster ⇒

Figure 8. Image clusters returned after clustering Image cluster 2 in Fig. 7 based on the color histogram feature.



Query words: pyramid egypt

Text cluster 1: 37 — th old century ancient cross georgia bui ■

Text cluster 2: 78 — ulane tech asia techhome jp gatech dail □

Text cluster 3: 122 — ɔk co sites macworld mccannas calandar □

Text cluster 4: 81 — lshing acrobat photodeluxe framemaker □

Text cluster 5: 66 — d courses ripe rgroups classical tape ups □

Image cluster 1: 0

Image cluster 2: 0

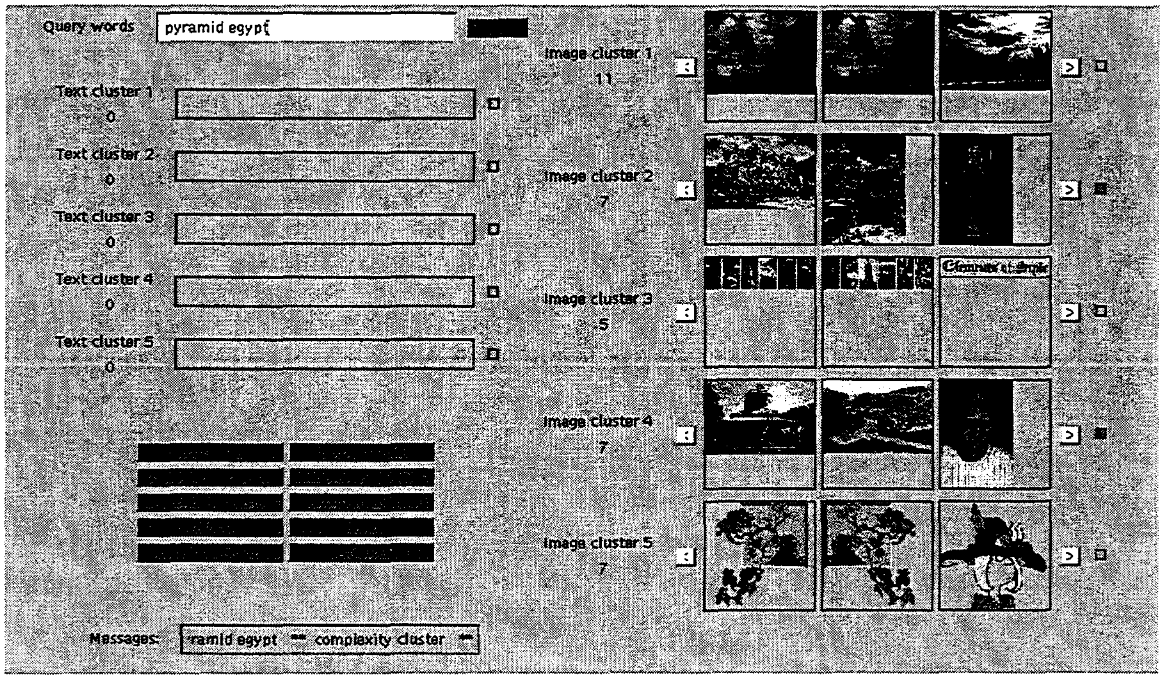
Image cluster 3: 0

Image cluster 4: 0

Image cluster 5: 0

Messages: ission ↔ textquery: pyramid egypt ↔

Figure 9. Text clusters returned in response to the query "pyramid egypt".



Query words: pyramid egypt

Text cluster 1: 0 □

Text cluster 2: 0 □

Text cluster 3: 0 □

Text cluster 4: 0 □

Text cluster 5: 0 □

Image cluster 1: 11 □

Image cluster 2: 7 ■

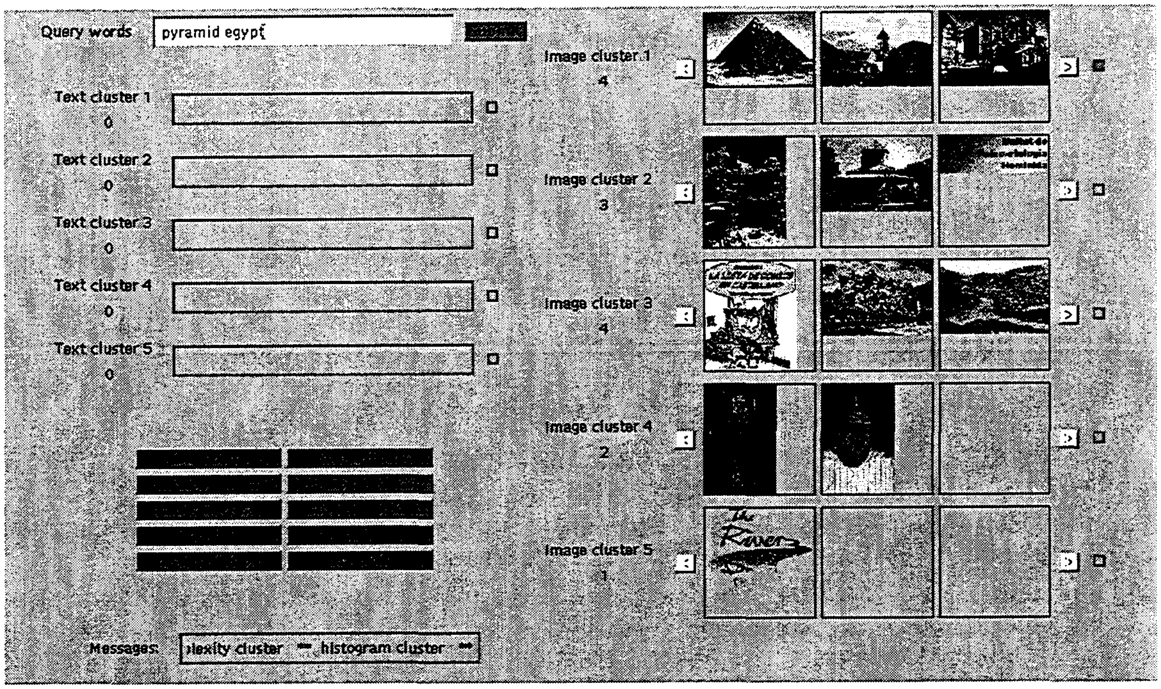
Image cluster 3: 5 □

Image cluster 4: 7 ■

Image cluster 5: 7 □

Messages: ramid egypt ↔ complexity cluster ↔

Figure 10. Image clusters returned after clustering based on the complexity feature.



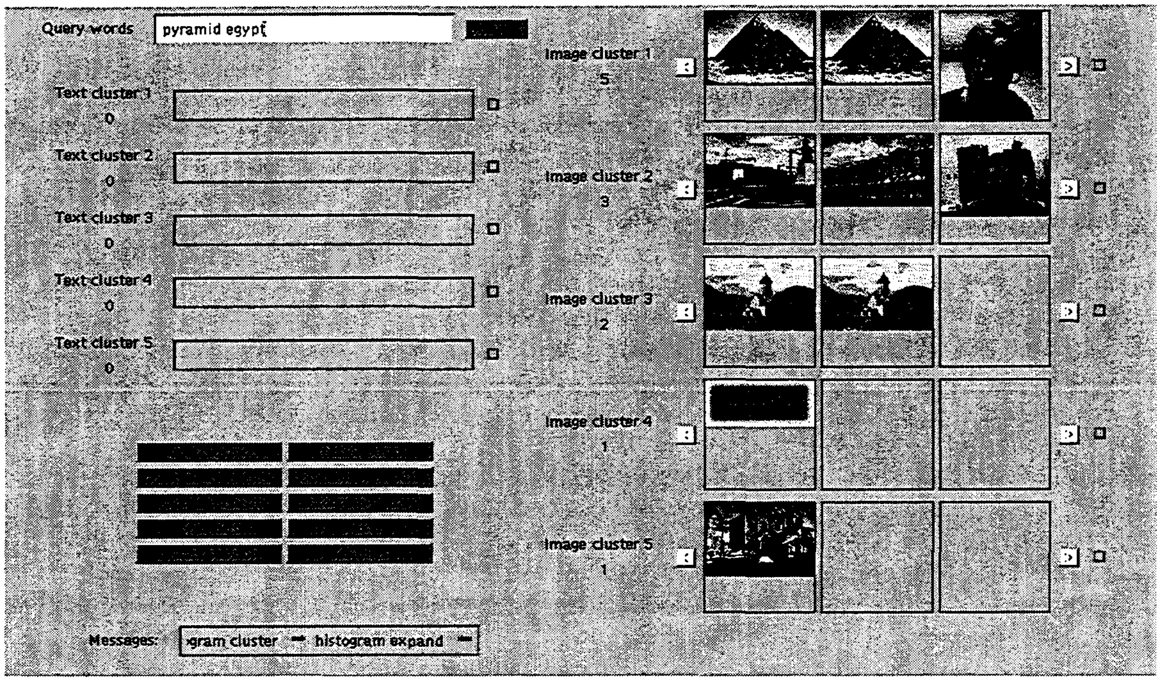
Query words: pyramid egypt

Text cluster 1: 0 □
Text cluster 2: 0 □
Text cluster 3: 0 □
Text cluster 4: 0 □
Text cluster 5: 0 □

Image cluster 1: 4
Image cluster 2: 3
Image cluster 3: 4
Image cluster 4: 2
Image cluster 5: 1

Messages: ›lexity cluster ➡ histogram cluster ➡

Figure 11. Image clusters returned after clustering based on the color histogram feature.



Query words: pyramid egypt

Text cluster 1: 0 □
Text cluster 2: 0 □
Text cluster 3: 0 □
Text cluster 4: 0 □
Text cluster 5: 0 □

Image cluster 1: 5
Image cluster 2: 3
Image cluster 3: 2
Image cluster 4: 1
Image cluster 5: 1

Messages: ›gram cluster ➡ histogram expand ➡

Figure 12. Text clusters returned after expanding the set of images in Fig. 11 and clustering the result based on the color histogram feature.

histogram feature. The resulting image clusters are shown in Fig. 11. Image cluster 1 contains four images, and the first image is of pyramids.

When Image cluster 1 is expanded to include similar images based on the color histogram feature, another image of a pyramid is identified, as shown in Fig. 12. This image occurs on a web page without any text and with a non-informative URL, and so it was retrieved on the basis of the color histogram feature.

In this example, the text query was used to reduce the size of the image collection, and the reduced collection was organized for presentation based on the image complexity feature. Additional images were obtained that were similar in the color histogram feature dimension.

In these examples, features in different modalities are used serially to help a user browse a set of images with associated text, using techniques of "scattering" and "gathering" subsets of elements in the corpus. A session begins with a text query to start with a more focussed initial set than the entire corpus. Clusters which are observed to contain one or more interesting elements can then be scattered to examine their content.

7. SUMMARY AND FUTURE WORK

We have developed a system for browsing a collection utilizing multiple modalities. Through an iterative process of "gathering" clusters and "scattering" the elements to examine the clusters, a user can find groups of images of interest. The expand function permits identification of elements in a collection that may be missing a value in one or more dimensions but are similar to other elements in some dimension.

In the future, we plan to enlarge the number of features and to investigate the utility of using the text features separately. The text feature can be enlarged by creating separate feature vectors for each term source (e.g., image URL, surrounding text, page URL), as described in the features section. An additional direction is to determine good methods for selecting subsets of features to combine at each step.

REFERENCES

1. D. Cutting, D.R. Karger, J.O. Pedersen and J.W. Tukey, "Scatter/Gather: A cluster-based approach to browsing large document collections," *Proceedings of the 15th Annual International SIGIR Conference*, pp. 318-329,1992.
2. M. D. Dunlop. *Multimedia Information Retrieval.* Ph.D. Thesis. Computing Science Department, University of Glasgow, Report 1991/R21, 1991.
3. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems,* 3, pp. 231-262, 1994.
4. E. J. Guglielmo and N. C. Rowe. "Natural language retrieval of images based on descriptive captions," *ACM Transactions on Information Systems,* 14, 3, May 1996, 237-267.
5. B.S. Manjunath and W.Y. Ma, " Browsing Large Satellite and Aerial Photographs," *Proceedings of the 1996 IEEE International Conference on Image Processing* Part 2 (of 3) 2 1996.
6. J. Marks, B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kan, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, S. Shieber, "Design Galleries: A general approach to setting parameters for computer graphics and animation", *SIGGRAPH97,* pp. 389-400, 1997.
7. T.P. Minka and R.W. Picard. "Interactive Learning With A 'Society of Models'," *Pattern Recognition,* 30, pp 565-581, 1997.
8. N. C. Rowe and B. Frew. "Automatic caption localization for photographs on World Wide Web pages," *Information Processing and Management,* 34, 1, 95-107, 1998.
9. Y. Rubner, L. J. Guibas, and C. Tomasi. "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," *Proceedings of the ARPA Image Understanding Workshop,* New Orleans, LA, 1997.
10. Y. Rui, T.S. Huang, and S. Mehrotra, "Relevance feedback techniques in interactive content-based image retrieval," *Proc. SPIE* 3312, pp. 25-36, 1998.
11. J.R. Smith and S.F. Chang, "An Image and Video Search Engine for the World-Wide Web," *Proc. SPIE* 3022, pp. 84-95, 1997.
12. R.K. Srihari and Z. Zhang, "A Multimedia Image Annotation, Indexing, and Retrieval System," *Proc. SIGIR Conference on Research and Development in Information Retrieval WWW Workshop*, pp. 29-45, 1998.
13. M.J. Swain and D.H. Ballard, "ColorIndexing," *Intl. Journal of Computer Vision,* 7, No. 1, pp. 11-32, 1991.
14. Arun Hampapur, Amarnath Gupta, Bradley Horowitz, Chiao-Fe Shu, Charles Fuller, Jeffrey R. Bach, Monika Gorkani and Ramesh Jain, "Virage Video Engine," *Proc. SPIE,* 3022, pp. 188-198, 1997.

Text Extraction from Video

Rangachar Kasturi **Ullas Gargi** **Sameer Antani**
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
kasturi@cse.psu.edu

We are involved in a research effort toward the automatic extraction of imaged text from general-purpose unconstrained video data. The goal is to build an efficient and robust system that will automatically detect the presence of text in a video stream, localize this text both spatially and temporally, and segment the text from the video frame in a form suitable either for presentation to a human or for processing by an OCR engine. We have achieved preliminary results in five main areas of this effort: ground-truth creation, testing of published algorithms, implementation of detection and localization algorithms, a method for robust tracking of text, and design of the overall system.

We are using the ViPER ground-truth creation tool created by the University of Maryland LAMP group to ground-truth a small set of varied video clips containing text of different styles. This ground truth will later be used for performance evaluation of the system with the ViPER-PE module. The ViPER interface is also used to generate an initial human-delinated bounding box for automatic tracking of text as described later.

We have studied the state of the art in text extraction from video and implemented and tested a number of algorithms published by other authorsfor text detection and localization in video and images. This has given us an insight into the problems still to be solved.

Further, we have implemented modified algorithms based upon published ones in addition to new methods that perform the localization task on MPEG-1 compressed video. The preliminary results indicate that no single algorithm is likely to work well for the various sizes, fonts, stroke thicknesses and other characteristics encountered in video text and that our approach of combining the decision of multiple methods is promising. However, individual algorithms are able to successfully detect and localize text of various sizes, contrasts, and stroke thicknesses. We are still testing our system on the ground truthed video testbed being created.

We have implemented a new tracking algorithm that works on MPEG-1 video. This accepts an initial bounding box and tracks its motion through the video, accounting for the appearance and disappearance of text from the frame and consequent modifications to the bounding box. This method appears to work robustly on our data. Modifications to make it robust to occlusion are ongoing.

The system design is still evolving. It can be broadly described as consisting of a set of modules (implemented as threads) that implement different facets of the task (multiple localization algorithms, tracking, high-level information, etc.) and that interact through a global queue of raw and processed video frames. The hope is that this design will allow more high-level logic to be built in that can account for, for example, varying partial occlusion of scene text that is never completely visible. We have used C++ and the public domain mpeg library and have implementations for SGI and Sun workstations. The algorithms have pseudo-random access to MPEG frames and operate in near real time.

Methods for segmenting text after localization will soon be added to the system. The final system prototype will allow text in MPEG-1 video to be automatially detected, localized, tracked and segmented. Alternatively, the tracking module can be used as a stand-alone tool to aid humans.

Next Generation Information Retrieval

Kathleen Romanik
powerize.com
901 Elkridge Landing Road, Suite 350
Linthicum, Maryland 21090
kromanik@powerize.com

Abstract

Powerize.com has developed a next generation information retrieval system called the Powerize Server™ to help workers extract relevant knowledge from distributed, heterogeneous information sources. With this product, a user can specify a topic of interest using a generic query language and can select multiple sources to search for information relevant to the topic. The Powerize Server then translates the generic query into a source-specific query for each selected source, connects to the source, and searches for relevant information. The results of the searches performed on diverse sources are filtered, ranked, merged, summarized and categorized using advanced artificial intelligence techniques, and they are presented to the user in a customized publication that can be viewed with a web browser.

We describe the architecture of the Powerize Server and key components of its design, including search wizards, power links and the distillation process. We also describe the underlying knowledge base that enables the server to provide users with a uniform interface for sending queries to diverse, distributed sources. Finally we describe the software development kit that allows the user to extend the capabilities of the server.

1 Introduction

The broad penetration of the Internet into corporate information environments, coupled with the growing use of corporate intranets, text retrieval systems, relational databases and professional online information services, has provided knowledge workers with a wealth of information at their fingertips. However, these information sources are distributed throughout the Internet and the corporate intranet, and each source has its own unique query interface for extracting relevant information. Knowledge workers are "drowning in info-glut" trying to find the information they need from the huge array of available sources.

Powerize.com has developed a next generation information retrieval system called the Powerize Server™ to help knowledge workers overcome the info-glut problem. The Powerize Server lets a user specify a topic of interest and select sources to search for information relevant to the topic. Interest topics are specified using customized search wizards that provide a simple, uniform interface for defining queries. Queries are represented internally using a generic query language, which the Powerize Server translates into source-specific queries for each selected source. The server connects to each source and searches it for relevant information. The search results are then filtered, ranked, merged, summarized and categorized using advanced artificial intelligence techniques, and they are presented to the user in a customized publication that can be viewed with a web browser.

2 Powerize Server Architecture

The Powerize Server architecture is illustrated in Figure 1. In the figure, boxes represent software components and circles represent information sources. A user accesses the server with a standard web browser and uses a search wizard to create a personal search profile by specifying topics of interest and selecting multiple sources to search for information relevant to these topics. These sources can include Internet search engines, specialized web sites, text databases, relational databases, real-time news feeds and professional online information services.

Using this search profile, the server generates specialized queries for each source selected by the user, connects to the source using a PowerLink (denoted by "P-Link" in the figure), and searches for relevant information. The results of the searches performed on these diverse sources are merged, summarized, ranked, and categorized by a distillation process that uses advanced artificial intelligence techniques. The results are then presented to the user in a customized publication that is also viewed with a web browser. The key components of the Powerize Server architecture are described in the following sections.

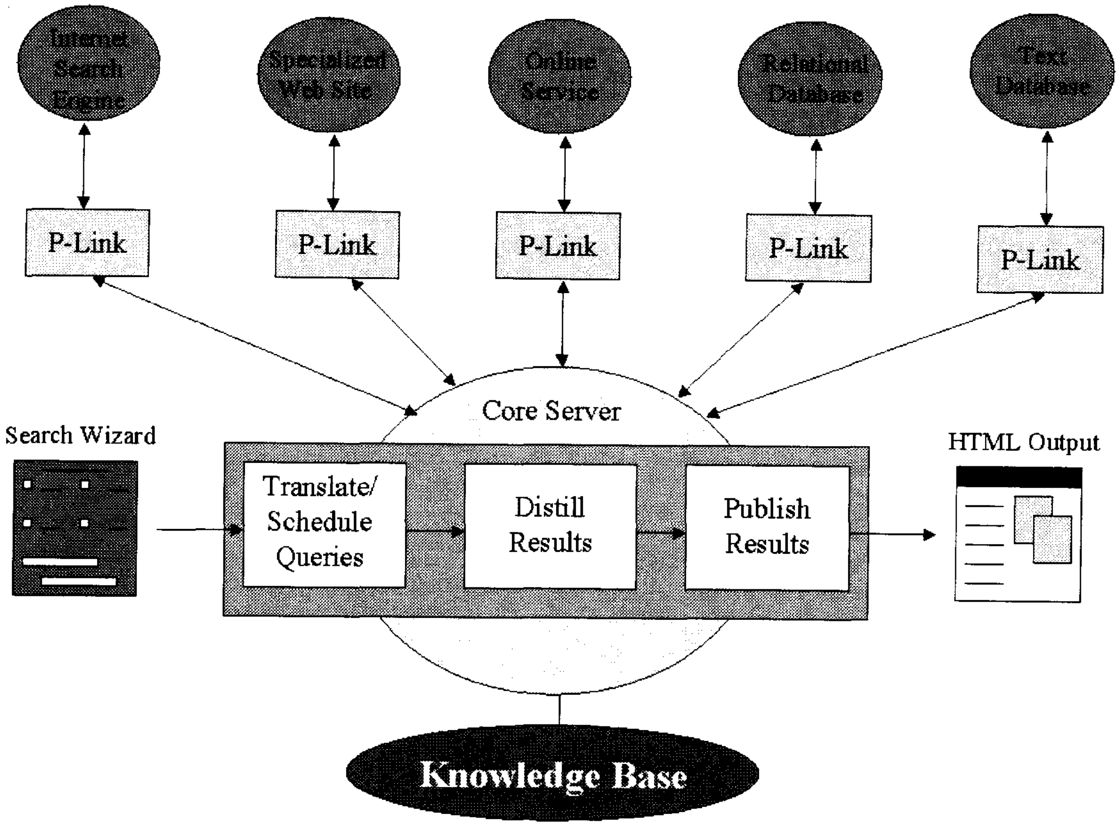


Figure 1: Powerize Server Architecture

3 Search Wizards

Powerize.com has developed specialized search wizards for several industries that work with the Powerize Server to provide an in-depth searching capability with a simple, intuitive interface. A wizard consists of one or more related topics, and each topic contains queries that are sent to multiple content sources. By entering a few terms into the wizard set-up screen to focus the search, a user can retrieve a large body of information relevant to several topics of interest without any knowledge of the sources being searched and without the time consuming task of creating a collection of search queries. For an example of a search wizard interface, see Figure 2. A partial description of the structure of this wizard is illustrated in Figure 3. The circles in Figure 3 represent sources that are searched for information on a topic.

The queries in a search wizard are expressed in a generic query language. The Powerize Server employs a knowledge base containing meta-information about each information source to translate the generic queries in a wizard to the native query language of each information source.

4 PowerLinks

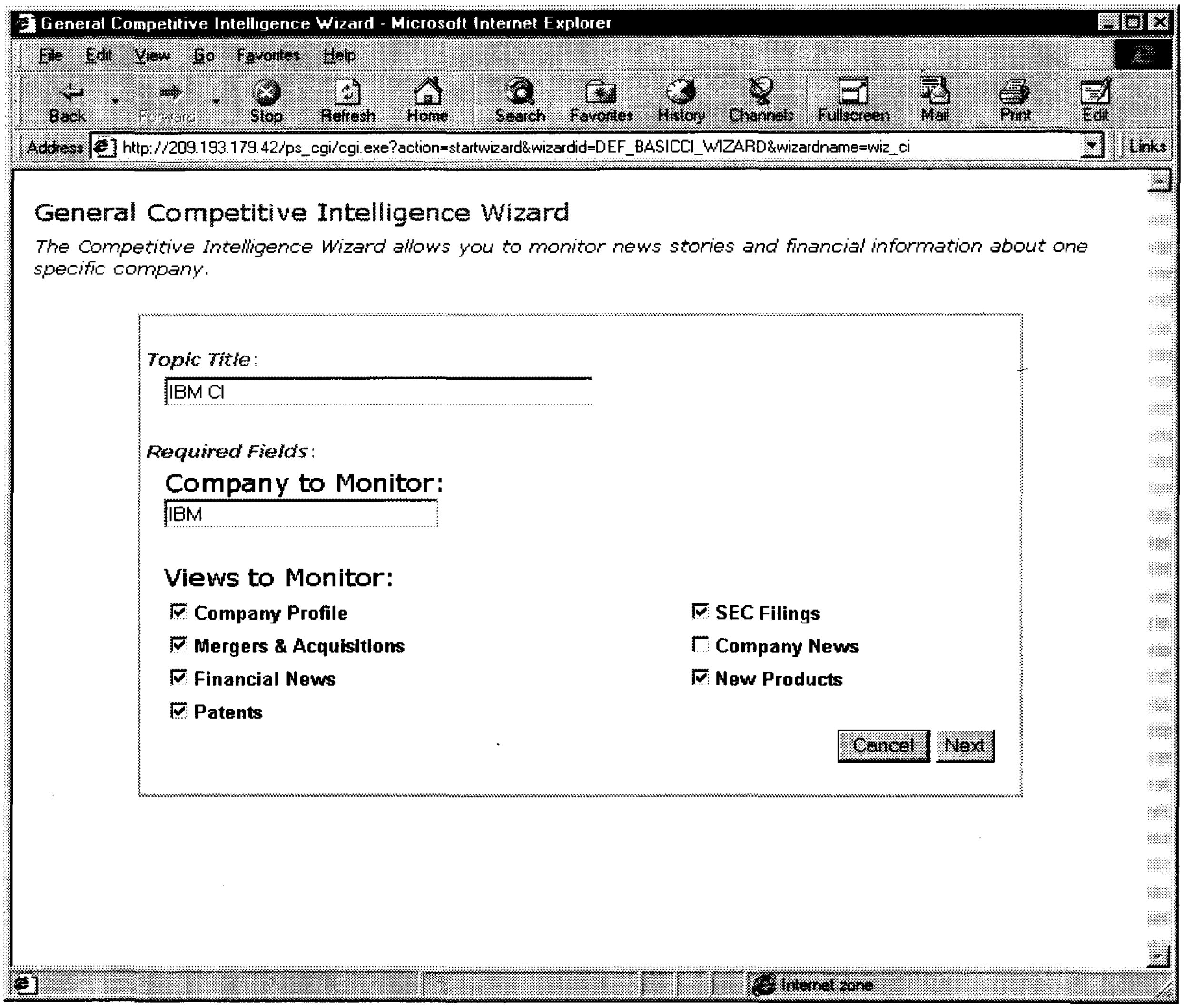
Each information source used by the Powerize Server is accessed via an intelligent software agent called a *PowerLink*. This software agent encodes information about the source such as protocols for connecting to the source and retrieving data from it, knowledge about how to effectively query the source for information, and rules for how to extract pertinent information from the data retrieved from the source.

PowerLinks have been developed for Dialog™, Lotus Notes™, Internet search engines, hidden web sites, ODBC compliant relational databases, NewsEdge™ real-time news, powerize.com Business Research Center™, Infonautics™, ProQuest™ and Lexis-Nexis™.

Using PowerLinks the Powerize Server allows users to retrieve information from multiple, distributed, heterogeneous sources using one seamless interface. PowerLinks also enable the core server software to be independent of source-specific issues.

5 Distillation Process

The distillation process in the Powerize Server includes methods for merging, summarizing, ranking, and categorizing documents returned from searches. After multiple sources are searched for information, the returned results are merged together and duplicates are removed. Each document is summarized using rules stored in the knowledge base for the source.


General Competitive Intelligence Wizard - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Links
Cancel
Next
Internet zone

Address http://209.193.179.42/ps_cgi/cgi.exe?action=startwizard&wizardid=DEF_BASICCI_WIZARD&wizardname=wiz_ci

General Competitive Intelligence Wizard

The Competitive Intelligence Wizard allows you to monitor news stories and financial information about one specific company.

Topic Title: IBM CI

Required Fields:

Company to Monitor: IBM

Views to Monitor:

- ☑ **Company Profile**
- ☑ **Mergers & Acquisitions**
- ☑ **Financial News**
- ☑ **Patents**
- ☑ **SEC Filings**
- □ **Company News**
- ☑ **New Products**

Figure 2: Search wizard set-up screen

Each document is given several rank scores based on Boolean, query-completeness, proximity, and number of hits measures. Several ranking algorithms using various combinations of these scores have been developed for sources accessed by different PowerLinks. These algorithms use morphology to expand a query before ranking a document against the query. For example, if the query contains the word "computing" the ranker expands it to also include the words "compute" and "computes".

In addition to being ranked, each document is categorized using a tool that classifies documents by organization, geographic location and people. It does this by first identifying proper names of people, companies, government agencies, organizations and geographic locations in a document and then placing the document into one or more categories based on the proper names found in it.

6 Knowledge Base

The seamless query interface of the Powerize Server is enabled with a knowledge base. This knowledge base stores meta-information about each source, such as the fields and operators used in its query language and the syntax for formulating correct queries to the source. The knowledge base also stores mappings from the generic fields, operators and values used in the server's generic query language to the corresponding source-specific components of the source's native query language.

As an example of how this knowledge base works, consider the task of someone searching for current contracts or research projects with the Army on decision-making software. Several databases potentially contain information about this topic. These databases index documents by the sponsoring defense agency and also contain fields for focusing on the subject of a contract or research project. However, the exact fields and values are different for each database, as well as the precise syntax for formulating a query. The knowledge base stores mappings from the generic field "Sponsoring Defense Agency" to specific fields such as "SP" or "SV" for each database. It also stores mappings of values for this field. The generic value "Army" may be mapped to "Department of the Army" for one database and to "Army Command" for another database.

The syntax for formulating correct queries to a source is specified using regular expression strings that guide the translation process. For example, one database stores the string "%F=(%V[Or %V]*)" with the mapping from the generic field "Sponsoring Defense Agency" to the specific field "SP". In this syntax string %F indicates a field name and %V indicates a field value. The notation "[]*" in the regular expression indicates that the part of the syntax string inside the square brackets should be repeated zero or more times, depending on how many values are present in the query term. Using this syntax string, the generic field-value pair "Sponsoring Defense Agency = Army" would be translated to "SP=(Department of the Army)". If the generic value "Army" were mapped to both "Department of the Army" and "Army" for this database, then the translated field-value pair would be "SP=(Department of the Army Or Army)".

7 Software Development Kit

Together with the Powerize Server, powerize.com has created a software development kit (SDK) that will allow users to develop their own search wizards and PowerLinks. Using the SDK users will be able to easily write PowerLinks for information sources that are not accessible via one of the many PowerLinks included with the Powerize Server, thereby making these sources available for searching by the Powerize Server.

Information experts, such as librarians, will be able to use the SDK to enter new sources into the knowledge base so they can be searched by any of the search wizards. They will be able to create mappings for new sources and to modify the mappings for existing sources. These mappings from generic entities (operators, fields, values) to source-specific entities will enable precise searching of these sources. Users will also be able to modify the generic query language used in the search wizards by adding new generic operators, fields and values to it.

The SDK will also let users create new search wizards and modify existing wizards. By creating their own generic queries, putting them into search wizards, and sending them to the sources that they most frequently search, users can quickly search for the information that they need without setting up an elaborate query each time they need this information.

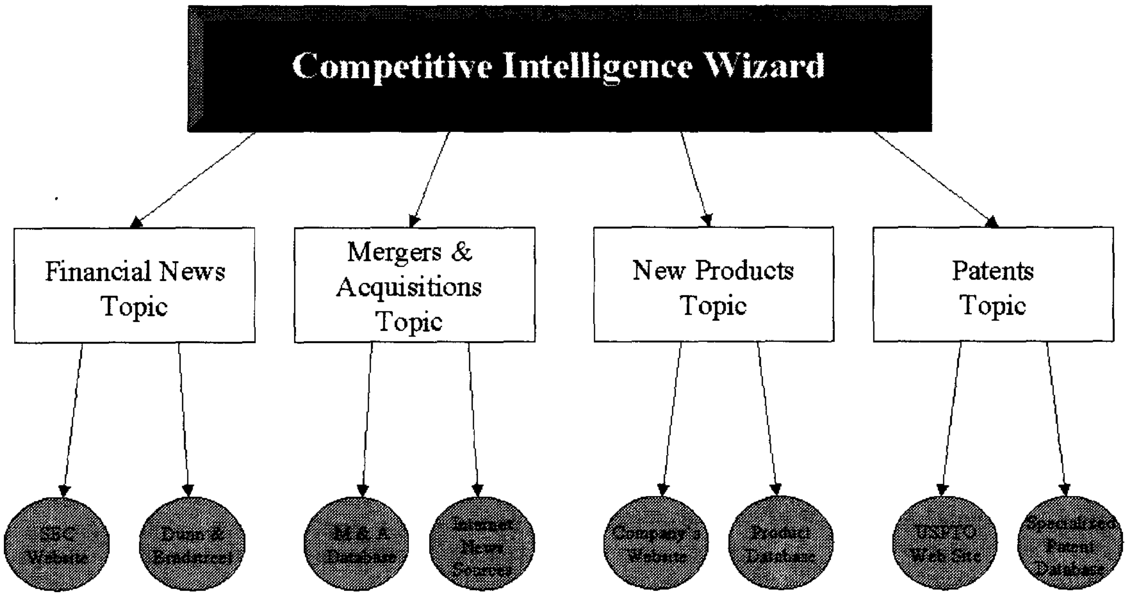


Figure 3: Structure of a search wizard

On Augmenting Documentation Reliability through Communicative Context Transport

Graziella Tonfoni

DPRC, The George Washington University, and BCN Group Inc.

Abstract

A mark up language for context sensitive packaging of information and a whole visual system for document annotation, will be presented.

The originating communicative context, in which a document was first generated, needs to be accessible, as to allow for full visibility of different communicative operations, which were performed or need to be performed upon the same document, both globally and locally, synchronously and asynchronously.

Operations that each document is likely to undergo during its life cycle are visually represented and finally conveyed as an attachment to the document.

Once a consistently interpreted and appropriately packaged document or piece of a document is available with its own originating context, it may then be reconfigured many times and more or less radically transformed, still without loosing the intended meaning.

Fuzziness and misinterpretation caused by lack of consistent clues for interpretation about originating contextual conditions, may this way be radically reduced and even eliminated, thanks to a consistently shared and robust system for accurate encoding of communicative contextual elements at play.

Introduction

A document comes from "somewhere in time and space and leads toward somewhere else": it may therefore be defined as a piece of information which has been derived from an information flow, dynamically evolving, then converted into a more stable form (Tonfoni 1996, 1998).

Enhanced encoding procedures for supporting accurate decision making are based upon conveying effectively those relevant clues, which altogether represent context throughout a combination of specific icons, which are of four kinds:

- **document annotation signs**: meant to indicate the communicative function or type of a document or piece of a document ;
- **document annotation symbols:** meant to indicate the communicative style of a document or piece of a document ;
- **document annotation turn taking symbols:** meant to indicate roles and interplay between the document producer and the document reader ;
- **document annotation amplifier symbols:** meant for constructing wider areas of documents, which show topical continuity and context consistency.

By topical continuity, we mean to indicate documents focusing on the same topic, which may be either literally extracted as linear sequences out of a document, or abstracted as a result of accurate interpretation and further adaptation at a conceptual level.

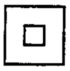
By contextual consistency, we mean to indicate documents showing the same communicative context, which is explicitly declared as to be easily retrieved.

A context enhancing mark up language

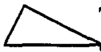
Document annotation signs, which represent the various communicative functions, a document may convey, paragraph by paragraph, are the following ones:

□ **Square**: for an informative document or piece of a document, which carries information about a specific event or fact, to be linked up with another document or set of documents made available, in order to extend topical continuity and context consistency.

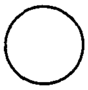
⧉ **Square within the Square**: for a summary of a certain document, which has been produced to reinforce contextual consistency between an original document and its own abstract.



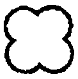
Frame: for a document or piece of document, which is found to be analogous in content to other documents and previously stored cases, is meant to reinforce contextual consistency between and among different documents.



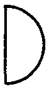
Triangle: for a memory and history generated out of a certain document, meant to establish topical continuity with background information, which has not been previously introduced, because not yet available.



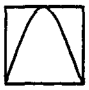
Circle: for a main concept conveyed by a certain document, which has been abstracted as to be linked to other documents, showing topical continuity. It is meant to reinforce topical keywords identification and to effectively link together documents, which show the same keyword.



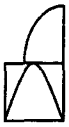
Grouped Semicircles: for main concepts, which are abstracted out of an originating document, meant to establish both topical continuity and context consistency between the originating document and a set of topical keywords.



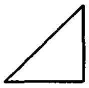
Semicircle: for a locally identified concept, abstracted out of a piece of document and meant to reinforce context consistency by establishing further links to other documents, which may be triggered by the same keyword.



Inscribed Arcs: for indicating the need for an upgrade and update of a certain document; it indicates that a revision process is likely to occur, though it does not declare if such revision will be a major or a minor one.



Opened Text Space: for indicating that an upgrade and update has indeed occurred within a certain document; it indicates that the document has now reached a new revision state; it does not declare if the revision has been a major or a minor one.



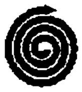
Right Triangle: for a comment made to a certain document or piece of a document, coming in, when more contextual information is needed, which has to be derived from other external sources, not previously available, based on topical continuity.

Document annotation symbols are meant to indicate communicative intentions and styles, locally within a certain document, sentence by sentence.

They are particularly useful as to show contributions made by individuals to the creation of a certain document and may be easily incorporated within the final document output, as to provide further clues, which may significantly add to clarity in interpretation.

Document annotation symbols, which represent different modes of information packaging, activated at different times or at the same time, may be combined and used dynamically for repackaging purposes. They effectively indicate transitional states within the same document, by declaring explicitly the nature of those changes, which have occurred or are likely to occur next.

They are the following ones :

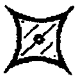


Describe: from Latin *describo*: write around.

It means complementing the original document or piece of a document with as much information as maybe found interesting to add, without any specific constraints.

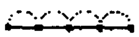
It is represented by a spiral, which starts from a central point –the middle point of the spiral indicating the original document- and proceeds toward expanding the document at various degrees, linking it with other documents or pieces of documents or information coming in from different sources and found to be relevant to facilitate the originating document interpretation.

Define: from Latin *definio*: put limits.



It means complementing the document with limited information about a very defined topic, which has been previously selected and identified as the most relevant one, which is represented by the middle point of the square.

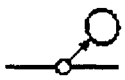
It indicates that there is a specific need to incorporate specific information about a relevant document or piece of a document, which is made available and implies accurate and most selective focusing on a very limited package of highly specific information.



Narrate: from Latin *narro*: tell the story.

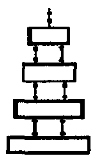
It means complementing the document with various facts and events, which have been referred to in the originating context, by following a logical and chronological order.

It indicates a set of major points or facts representing different diachronic stages, which are strictly linked up together in a sequence.



Point out: take a single point out of a story chain.

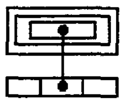
It means isolating a specific event or fact among those reported within a single document, focusing on just that one, and adding more detailed information, by expanding it significantly, as to have it linked with other documents, which have been found to be of relevance to that point .



Explain: from Latin *explano*: unwrap, open up.

It means that facts and reasons are given as to support interpretation of a certain event within a certain document or piece of a document.

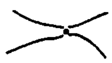
The document producer may start by indicating the originating cause and proceed toward showing the effects or start with effects and go back to the cause, according to what is found to be more significant.



Regress: from Latin *regredior*: go back.

It means that more information about a certain topic, presented within the document, is absolutely needed as to gain a deeper understanding.

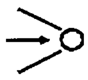
It represents a specific topic focusing process and an in depth information expansion, which is activated only for that precise topic. The document reader may want to consider if further information is needed on that and ask for availability of further resources.



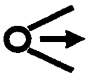
Inform: from Latin *informo*: put into shape, shape up.

It means that any document is the result of some information packaging and that the very specific document indicated is packaged in the most unconstrained way, therefore subject to many and various kinds of repackaging.

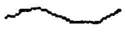
It leads toward two different kinds of further specification, which are respectively conveyed by the "inform synthetically" and the "inform analytically" indication:



"**inform synthetically**" means departing from a larger document or set of documents and proceed toward a summary related to a specific topic, identified as being the most relevant one emerging from the originating document .

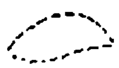


"**inform analytically**" means departing from a given document or limited set of documents as to expand toward further documents or add more information, which needs to be previously converted into the form of a document, which is not available yet.



Reformulate: from Latin *reformo/reformulo*: change shape and shape again.

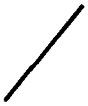
It means changing the kind of information packaging, which was adopted before and substituting a certain information request with a different one, still related to the same document. It may turn into a more or less radical transformation of the originating document, according to a precisely defined request or set of requests.



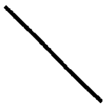
Express: from Latin *exprimo*: push out and press out.

It means adding personal opinions and individual feelings related to facts and events within a certain document; it indicates the most subjective mode of information packaging, which is openly seen as bound to very personal evaluations, judgements and emotional states.

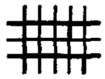
Document annotation turn taking symbols are meant to define the mode of accessing and reading the document, requested at each given time; they are suggested by the document producer to be followed by the document user; they are the following ones:



Major Scale: it shows that literal interpretation is needed and that those pieces of documents indicated and marked off, should be extracted and quoted literally, the way they were first packaged.

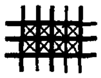


Minor Scale: it shows that accurate interpretation may need a further process of abstraction and that pieces of documents indicated and marked off, may undergo significant reconfiguration processes, up to abstraction.



Open or Unsaturated Rhythm: it shows that accessing the document may lead the user toward incomplete interpretation of those facts and events, which are presented.

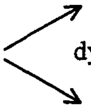
It is meant to suggest that the user access more documents and various kinds of sources, which are, made available.



Tight or Saturated Rhythm: it shows that accessing the document will lead the user toward complete interpretation of those facts and events, which are presented. It is meant to suggest that the user sticks to the interpretation provided, though access to other sources is still available, as to support evidence.

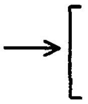
Document annotation amplifier symbols come last and may be added only after the previously illustrated ones have been used; they apply to larger documentation territories and indicate specific operations, which are to be performed as to connect sets of documents, which have been previously encoded and accurately stored.

They are the following ones:

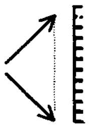


Choose: it is meant to represent the dynamic process of first identifying and then deciding between optional contexts for interpretation, which are mutually exclusive, given a certain set of documents.

Identify: it is meant to represent definition of a

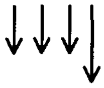


more specific context, within a broader context for interpretation of a set of documents; it naturally occurs before "search" and "select".

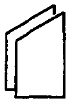


Search: it is meant to represent the dynamic process of choosing among different contexts for interpretation of a set of documents, which are many and compatible as to find the most appropriate one.

Select: it is meant to represent multiple contexts,



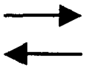
which may evolve either synchronously or asynchronously and may be modified, once a certain decision making process has been performed, as to be stored and kept as an example.



Copy/Replicate: it is meant to represent the dynamic process of duplication and repetition of a certain context, which, if lost, would affect understanding and accurate interpretation of a set of events and facts, described and explained by a set of documents.

Ahead: it is meant to represent the progression of a certain set of documents, which are linked together by context consistency or harmoniously shifting contexts.

Back: it is meant to represent the need to go back to delete and replace the originating context, which has radically shifted, in the course of various transition states, such that, if not eliminated, would indeed affect consistent interpretation of a whole set of documents.



Conflict: it is meant to represent an emerging inconsistency and incompatibility between various context attributions to a set of documents, which needs to be cleared as to proceed toward any further interpretation.

The whole document annotation system, here illustrated in its various components, may be applied at different layers and at various levels of complexity and is meant to underline the fundamental role and responsibility of the encoder individual and of the encoding team.

Conclusions

Just like geographic maps only show those features, which become relevant according to the nature and purpose assigned to the map itself, the same "way of thinking" may be extended to documentation mapping, according to various packaging and repackaging priorities, in continuously shifting contexts.

Energy and time dedicated to quite an expensive process, such as enhanced encoding is, may this way become time and cost effective, because each encoded document will provide an enormous amount of examples and in-house knowledge, which will remain extensively available.

Basic references

[1] **Tonfoni, G.**, 1996, Communication Patterns and Textual Forms, Intellect, Exeter,U.K. .

[2] **Tonfoni, G.**, 1998, Information Design: The Knowledge Architect's Toolkit, Scarecrow Press, Lanham, Maryland, U.S.

A Statistical Approach to Corpus Generation

A.E.M. Brodeen F.S. Brundick
U.S. Army Research Laboratory
Aberdeen Proving Ground, MD

M.S. Taylor
University of Maryland
College Park, MD

A quantitative measure of the efficacy of an optical character recognition (OCR) product is most often pursued through submission of an appropriate collection of groundtruth source-language documents to the OCR device for interpretation. Specialized scoring software then compares the OCR output with the corresponding groundtruth to produce descriptive statistics: character accuracy, word accuracy, and a confusion matrix* are common measures. An accessible database of groundtruth documents, accepted as authoritative by the research community, would enable the evaluation of an OCR product to proceed from a common benchmark. Unfortunately, such databases do not widely exist, making the comparison of OCR products more tentative.

Collection of a corpus that is sufficient for evaluation of an OCR product is in most instances a burdensome task. Access to a sufficient number of source-language documents representative of the document classes of interest may not be feasible and, even if obtained, an expensive and time consuming process of preparing groundtruth remains. To address this problem, we are investigating a statistical approach to corpus generation based on a small set of source-language documents.

Consider the block of Cyrillic text (Fig. 1) and its digital (or more precisely, codeset) representation (Fig. 2). In Figure 2, the first 80 letters of the Cyrillic text are displayed; the vertical dashed lines mark the location of inter-word spaces. The x-axis indexes the order of occurrence of the characters and the y-axis measures the corresponding codeset values. Figure 2 bears a striking resemblance to a discrete time series and, if we consider a situation in which the characters are processed sequentially, then we can assign to each character an associated time epoch. With this observation Figure 2 can, without loss of generality, be considered a time series representation of the block of text (up to character 80). The scale of measurement for the y-axis is nominal; another choice of codeset would lead to a different graph, but with no attendant loss of information.

In order to build a corpus, we would like an authentic document to serve as a basis from which to generate additional pseudo-documents or, consistent with the time series model, we would like to use the authentic time series as a basis from which to generate additional time series.

This situation has arisen in the analysis of time series data and has been addressed with some success through the use of a computer-intensive resampling plan known as the *bootstrap.* The bootstrap is a robust data-based simulation method for statistical inference that finds widespread use in applied statistics. While conceptually simple, and on the surface appearing as little more than sam-

*A confusion matrix shows the requisite number of character insertions, substitutions, and deletions required to reconcile the groundtruth and OCR output files.

"Нецес остати пусто Невесиње равно, но цес бити оно
сто си вазда било: расадник Српства и колевка лава!"
"Србија мора постати велико радилисте и родилисте!"
Ово су само два изватка из скорасњих говоранција Вука
Драсковица. Анахроницна, срцепарајуца реторика,
примерена политицарима осамнаестог века, представља данас
најбољи пример лази-говора или језика-маске. А онај ко на
себе стави маску лази-говора, пре или касније, изабраце и лаз
као основни политицки принцип. Нико није изрекао толико
лази, подвала и лазних доказа о Косову као г. Драсковиц
и његова телевизија. Раније смо ту анахроницну реторску
маску примали као некакав његов особењацки избор, као
сто примамо нецији цудацки стил у одевању. Требало је за
његову реторику реци оно сто је одувек и била - да је обицан
киц.

FIGURE 1.—*Serbian Text*

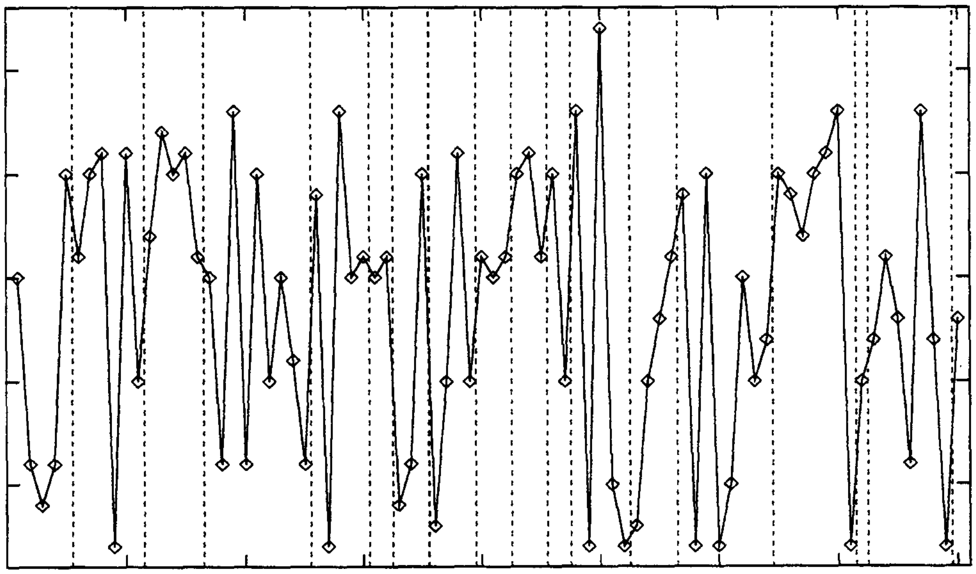


FIGURE 2.—*Serbian Time Series With Interventions*

pling with replacement, its mathematical foundation is quite deep and has attracted the attention of researchers for at least the last two decades. Efron[†] has been a vigorous advocate and to large extent is credited with the growth in popularity of the procedure. The name bootstrap alludes to the phrase regarding someone who, commencing from a position of woeful inadequacy, goes on to "pull themselves up by their bootstrap." It is not the same as the computer science expression meaning to "boot" a computer from a set of core instructions, although it shares a similar concept.

We will present here only an abbreviated outline of the bootstrap procedure applied to the time series model. At the onset, it is helpful to notice that the time series has an inherent structure. The time series represents a block of text—it is not a random sequence. Moreover, the words themselves are subject to lexical constraints and hence the patterns that they assume in the codeset representation have meaning. The interword spaces play the role of interventions in time series modeling. As a consequence, the time series has local structure contributed by the word patterns but little global structure due to the high frequency of interventions. A fundamental requirement of the bootstrap procedure applied to these data is that the characteristics of this structure be retained.

Denote the time series as a sequence of ordered pairs $(x_1, y_1), \ldots, (x_n, y_n)$. We begin the bootstrap procedure by choosing at random a location within the time series, say x_r. Starting at x_r, we copy a subsequence $(x_r, y_r), (x_{r+1}, y_{r+1}), \ldots, (x_{r'}, y_{r'})$ and store it as the start of a new sequence; the length of the subsequence, $r' - r + 1$, is determined by sampling from the empirical distribution of word-lengths contained in the original document. A second point of entry, x_s, is determined and a second subsequence $(x_s, y_s), (x_{s+1}, y_{s+1}), \ldots, (x_{s'}, y_{s'})$ is copied and appended to the previous subsequence, separated by an intervention, and so on, until a stopping rule terminates the process. At that point, a bootstrapped time series with interventions (Fig. 3) has been produced; reversing the mapping from codeset values back into letters and performing syntactic refinements produces a bootstrap document (Fig. 4).

Can the document displayed in Figure 4 be "read" by an individual? Of course not. Recall, however, that our intent was to produce a document image (or character string) sufficient to assess the character recognition ability of an OCR device. If the OCR device has incorporated decision aids to support character segmentation the bootstrap document will likely reduce the effectiveness of those procedures. Clearly, spell-checkers will not be of value. Lexical analyzers will likely be inhibited, but not rendered completely ineffective, since some local structure has been retained by the bootstrap sampling procedure.

The bootstrap procedure brings no added value if an adequate groundtruth database is available. In the more likely situation where the database is inadequate, to include no database at all, the bootstrap offers a rapid and inexpensive way to respond to this deficiency. Most research questions regarding the effectiveness of the bootstrap procedure for corpus generation remain unanswered at this juncture. We are only now beginning to accumulate empirical evidence. Theoretically, we have cause for optimism; practically, another time-worn expression cautions that "The proof of the pudding is in the eating."

[†] B. Efron and R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.

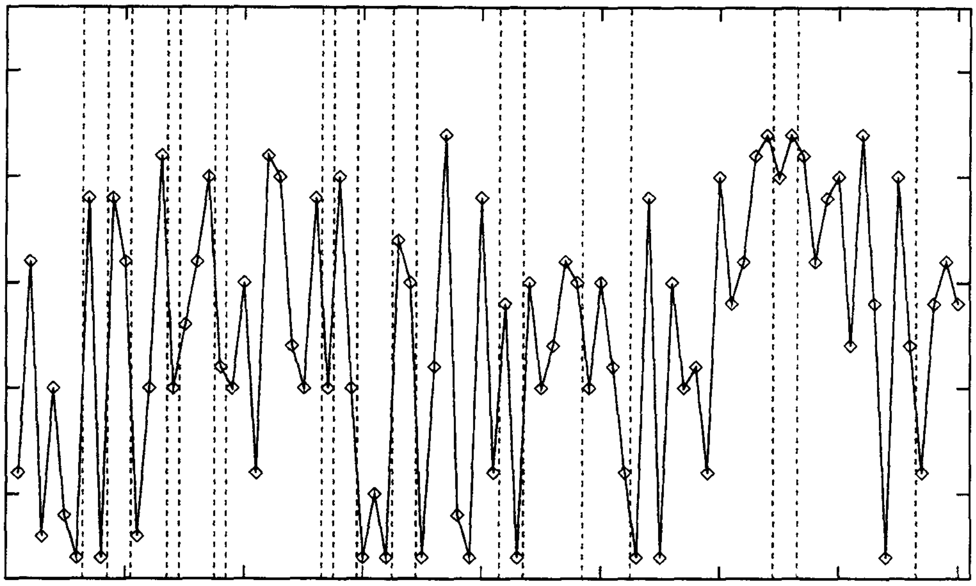


FIGURE 3.—*Bootstrapped Time Series*

Еобица ра Ро бит. И лос ј" инецкир И си ада Пн ајуцаре ма никон иње аранијесмоту су торскумаск емом драсковицињ оно. Маљ" строфан палиозез, ијеподеф бољиприме њеравно, еподефи ник хуман његово. Ихуманиста цесбит примецен надозбун есоста а ацк роф трпљ илазнихд? Ајеобиц тастроф "амаскеа о изарпре зика ов аизватк уцутипр ст. "Бестави кадра тицар аос обењацк есљава еговат Срцеп азец оруцива то. Аз их а цкипри го илазка астиље иљ р рску. Стинерас икаон анау, ог јесм икаприме, татибел се Ињ" рикаприме, кцијом" джун оц његовурето апре, ика? Лоцан цисесељ ацкииз телевиз мо такође Тојeдав палиозези, ци Имс стоје ву крволоцан. Имамоне аскулазиго остоје м зва" роц инецкиратн иг ебест л вима ихекспеди овор сост радасусеосец ицсесељев ика. Ц ребалојеза вљад цесоста мсилином и ијара хтеваодасе и ле јом цсе олитицари торикап, Бити стотог.

FIGURE 4.—*Bootstrapped Text*

Automatic Groundtruth Assignment for Text Image Databases

Andrew Gillies, Ralph Krug

Nonlinear Dynamics Incorporated
123 N. Ashley Street, Suite 120
Ann Arbor, MI 48104

ABSTRACT

At the heart of many document understanding systems is the text recognizer: a module for converting images of text into ASCII, Unicode, or some other form of machine coded text. One of the most laborious stages in the development of a text recognition system is the collection and processing of training data. The training data often consists of images of text, along with groundtruth files. A groundtruth file contains an ASCII (or other) representation of the text which appears in an associated document image. In preparation for recognizer training, each character of text from a groundtruth file must be associated with the location of that character in the image file. We refer to this process as groundtruth assignment. In many development efforts, groundtruth assignment is performed manually, in a truth assignment editor. This adds greatly to the cost of developing a new text recognizer.

This abstract describes an automatic system for groundtruth assignment based on hidden Markov models (HMMs). The system, called and *autotruther*, analyzes image and groundtruth files to produce image coordinates for each character from a groundtruth file. The system has been used in the development of an Arabic text recognition system.[1] The approach is amenable to a large variety of languages, scripts, and text recognition methodologies.

The system begins by breaking text images into single lines of text. Each line from the image is then associated with a line from the groundtruth file. Because errors in line assignment lead to many unrecoverable character assignments, we use a manual review process at this stage. Because the line assignments are coarse grained, the time to do manual review of this kind is not a problem given an appropriate viewer. The result of review is a set of line images correctly aligned with lines from the groundtruth file.

The autotruther uses a feature extractor which assigns a 96 element vector to each column in the line image. Each element of the vector is associated with a character stroke contour at a given orientation, curvature and y–location in a 5 pixel window around the column of interest. The vectors are then mapped into an integer in the range 0 to 99 by a vector quantization process. The vector quantization codebook is produced by a simple k–means clustering of a sample of feature vectors extracted from line images. The result of feature extraction and vector quantization is a representation of the line image as a sequence of integers.

The heart of the autotruther is the line alignment module. This module uses an HMM to perform line alignment. The HMM is based on a model of the typographic conventions which are applied in transforming the textual representation found in the groundtruth file into the graphic representation found in the image. In the case of Arabic text, the model is used to account for the strikingly different forms which Arabic characters take in different contexts. This includes both the four normal forms of each character, and the alternate forms for character pairs which have been combined into a ligature character. In general, the model might also be used to capture in the assignment process such things as varying type faces, italics, and the like.

At the core of the text model is a representation of two character sets. The *truth set* contains all characters which may appear in the groundtruth files. This may be a subset of Unicode, for example. The *glyph set*, on the other hand, is a representation of all the character shapes which are to be discriminated by the autotruther, and which are ultimately to be discriminated by the recognizer. In some cases, a finer distinction may be drawn in recognizer space than is actually required for recognizer output. Thus, one may design an autotruther to distinguish italics, and train a recognizer to distinguish italics, but then throw away the italic attribute for a text file output. In the case of the Arabic system, the four forms of a character are recognized as four separate shapes, but the output file simply produces a sequence of unicodes which do not distinguish the forms.

The autotruther creates a directed graph for each line image generated from the corresponding line in the groundtruth file. For simple text models this graph can be just a linear sequence of model states, one for each character in the truth file. For Arabic, this model contains branches for the pairs of characters which may be represented either as ligatures or a pairs of simple characters. Each state of the text model corresponds to short sequence of states in the HMM. In the Arabic autotruther, each glyph corresponds to from five to fifteen Markov states.

The line alignment itself is accomplished by the Viterbi algorithm within the hidden Markov model. This algorithm computes the maximum likelihood sequence of model states visited as the HMM as it traverses the sequence of integer codes produced by the feature extraction process. Once this sequence is known, the boundaries between adjacent characters can be computed to the pixel level. Note that the HMM not only locates each character, but also chooses between alternative glyph representations of the truth text.

The output of the HMM is a character location file which gives the glyph number, left x–coordinate and right right x–coordinate for each character in each line image. This is a relatively simple and universal format which should be of use to recognizer developers using a variety of character recognition mechanisms. In the development of the Arabic recognition system, this level of truth is input to a second level of truthing which is much more recognizer–specific.

The autotruther HMM does, of course require training. It is trained using the sequence/model pairs. Initial model parameters are estimated from a small amount of previously aligned character data. Once the HMM is trained, it reprocesses the sequence/model pairs to arrive at the ultimate assignment of model states to columns in the image.

The approach used by the Arabic autotruther is amenable to a large variety of languages and scripts. Some of the code would have to be modified for languages written in columns rather than lines, and the modeling module might have to be extended for typographic conventions not occurring in Arabic or roman scripts, although we do not know of any conventions which could not be accounted for within the framework already developed.

The evaluation of a groundtruth alignment is in general difficult to define. Our experience with the autotruther, however, suggests that its performance is well above anything we have used before. We consider it to be one of the key technologies used in the development of a text recognition engine.

[1] A. Gillies, E. Erlandson, J. Trenkle, S. Schlosser, "Arabic Text Recognition System", in this volume.

Government Support

Document Image Analysis Technology for Information Dissemination and Retrieval

Stephen J. Dennis
U. S. Department of Defense
9800 Savage Rd., R522
Fort George G. Meade, MD 20755-6000
sjdenni@afterlife.ncsc.mil

Abstract

The U. S. Defense Department scans a large volume of foreign language paper documents in order to identify information of interest to the Government and to identify information that can be released to the public. Content-based access to these paper documents facilitates information analysis for a distributed collection of scanned document images. To address the requirements for rapid information access, the DoD has developed a fully automated document image analysis system that processes a large number of pages per day.

The automated indexing system operates on a principle of rapid information diagnosis. It first discovers the script and languages of machine-printed text in the scanned image. This information is used to direct image data to the appropriate language character recognizer, if available, or otherwise to humans with appropriate language skills. Any text derived from the OCR process is represented in an information retrieval system that is enhanced with detected graphical objects indexed by a feature signature. The system relies on static queries for dissemination and dynamic user queries for archive retrieval; each involving text and graphics. Duplicate copies of the same document are also clustered.

User studies indicate that the resulting performance of the system is more efficient than simply combining OCR and text retrieval technologies. Hybrid search involving both text and graphics provides a better ranked list of documents returned from a query containing text and graphic information requests. De-duplication of document image data can save a significant amount of time in reviewing document images returned from a query, and improves the timeliness of information analysis.

Issues regarding language independence of the system are important, and continue to be addressed. Multi-lingual text retrieval technology is new, and there are few robust techniques available. We continue to develop new techniques for rapid image diagnosis and focus on automatic extraction of high-level information to represent the content of a document image.

Document Image Working Group (DIWG) Workshops and Evaluations

Ivan Bella (ivan.bella@lmco.com)
Lockheed Martin
7100 Standard Drive
Hanover, MD 21076

Abstract

Over the course of 1998, several workshops were organized by the Document Image Working Group (DIWG), a collaborative effort across several Government organizations. These workshops were to discuss evaluating several document image understanding technologies, including Duplicate Document Detection, Document Segmentation/Page Decomposition, Meta-data Extraction, and Information Retrieval. This paper provides a summary of the results of these workshops. One significant outcome was that the DIWG was to prepare several datasets to evaluate one or more of these technologies. Duplicate Document Detection was decided to be the top priority and a dataset is being put together to be distributed. The details of the dataset and evaluation results gathered to date are included in this paper.

1 Overview

The Government has organized a collaborative effort across several Government organizations to discuss the evaluation of document image understanding technologies. This group, the Document Image Working Group (DIWG), organized several workshops with the assistance of Lockheed Martin to discuss the issues surrounding such evaluations. This paper documents the results of these workshops and the evaluation efforts underway.

2 Document Image Working Group

The Document Image Working Group (DIWG) is composed of representatives from the Department of Defense, the Army Research Lab, the Central Intelligence Agency, the Department of Energy, and the Gulf War Declassification Program. This group was assembled to collaborate on document image understanding technologies including digitized document collections, document management systems, and the advancement of automated document processing. Several extended areas of interest include speech processing, text processing, video processing, multimedia processing, and information retrieval involving very large databases.

Early in 1998, with the help of Lockheed Martin, several workshops were organized to focus on the evaluation of three specific areas of Document Image Understanding: the detection of duplicate documents, the segmentation and extraction of meta-data from documents, and Document Image Understanding as it relates to Information Retrieval. The motivation for these evaluations was to assess the current capabilities in the general research community in order to make investment decisions involving research and system implementations.

3 Workshops

Two workshops were organized, the first on the West coast and the second on the East coast. The workshop on the West coast was held at the Stanford Research Institute (SRI) in Menlo Park, CA. on the 29th of July. The East coast workshop was held at Lucent Technologies in Murray Hill, NJ. on the 12th of August.

The expectations coming into the workshops were to share the Government's application requirements, to establish the requirements for various evaluations, and to organize and run the evaluation of various document image understanding tasks. The short term goal was to start on a modest scale for the preparation of representative datasets and the definition of the evaluations. The workshops would allow the investigation of new approaches and alternatives, and to achieve specific short term results in terms of performance measures. The long term approach was to look to others within the community to increase the size and scope, and to build a program to sustain the research and evaluation efforts.

4 Workshop Results

The workshops were generally very informative in terms of the requirements for evaluations. The results of the discussions are organized in terms of the three areas of interest: Duplicate Document Detection, Document Segmentation and Meta-data Extraction, and Document Image Understanding as related to Information Retrieval.

4.1 Duplicate Document Detection

The commercial interest in the area of Duplicate Document Detection includes form detection and copyright violation detection whereas the Government's interest is primarily for the declassification program. The general problem in this area is to automatically determine whether two document images are of the same (or close to the same) document.

The main problem surrounding this area is the definition of a duplicate. There appear to be two main classes of duplicates: *exact* and *near* duplicates. An *exact* duplicate is usually considered to be the same document copied or distorted in some manner that does not change or add to the semantic value of the document. A *near* duplicate is a copy of a document with semantic changes or additions such as handwritten annotations or different drafts of the same document. Note that these categories can change depending on the actual application.

It was noted that the data, as with any evaluation, needs to be representative of the application area. However, in terms of the number of duplicates, triplicates etc., it was thought that an over-sampling would perhaps be sufficient and perhaps even preferred in order to get significant results from an evaluation of algorithms. This works well for the declassification program as an initial sort of the documents is done based on the title and date of the documents. The resulting document groups are then sent through a duplicate document detection process. In any case, a statistical method needs to be performed to determine the percentage of duplicates for an application to ensure a valid evaluation can be performed.

It is significant to state that identifying duplicate *documents* is a slightly different problem than identifying duplicate *pages*. However the general consensus was that we should proceed with the duplicate page problem before handling the duplicate document problem as one feeds into the other.

The most intuitive metrics for an evaluation of this problem area was thought to be hits versus false-alarms. However, there is normally some sort of parameter associated with these algorithms that can change the probability of correct detection with a resulting change in the probability of false-alarms. This parameter needs to be taken into account when reporting the results.

It was also suggested that a useful approach to evaluating this problem area would be to look at only the disagreements between the algorithms being compared. This would be a good approach especially if it is not guaranteed that all the duplicates in the dataset are known. In the Text Retrieval Evaluation Conferences (TREC), if the ground truth was unknown for a given dataset, there was a way to bootstrap a confidence measure by looking at the union of all of the returned results. This method may be useful for this problem area. It was noted that a small dataset which is completely ground truthed may be used to measure the precision and recall, and a larger dataset in which the ground truthing process is too costly may be used to evaluate the differences between the algorithms.

Additional evaluation results that may be useful would be the correlation between Duplicate Document Detection and image differences and the correlation with the human categorization of duplicates.

A final interesting comment was made about the Library sciences where it was found that humans were only about seventy percent (70%) consistent in determining how to index or classify book topics. It was suggested that a similar measure should be performed for the duplicate document detection problem to see how consistent humans are in classifying duplicate documents. We may not be able to expect better from an automated process.

4.2 Document Segmentation and Meta-data Extraction

The commercial interest in this area was thought to be very large as it is a required step in almost every Document Image Understanding task such as Information Retrieval and Document Routing. As a result there are many existing datasets that may be used in the evaluation of this area:

1) MEDLINE index
2) IEEE scanned documents
3) ICDAR proceedings
4) UNLV-ISRI data

Since this area of interest has so many applications, the meta-data to be extracted can be fairly large and very costly to ground truth. Therefore the types of meta-data to be extracted need to be carefully determined. Some of the suggested meta-data included:

1) document type or source: letter, magazine, newspaper, etc.
2) multi-column versus table
3) text content
4) reading order
5) page segment type: author, footnote, headline, title, abstract, table, graphic, etc.
6) noise categorization
7) font type

One of the main problems in this area was thought to be the data format. Formats that may be considered are that of the University of Washington datasets [1,4], the DAFS format [2], the RDIFF format [3], or an XML standard format. The main problem is that the tools used to create the ground truth need to be supported and enhanced as the problem area becomes more mature.

4.3 Document Image Understanding and Information Retrieval

The commercial interest in this area is also fairly large, for example digital libraries and the World Wide Web. It was a general consensus that the Government needs to regroup and assess the real requirements in this area. In general, three Document Image Understanding processes were thought to be significant in terms of relating the results to Information Retrieval: meta-data extraction, OCR, and machine translation.

In terms of the meta-data extraction process, the questions to be answered are:

1) Can better information retrieval be done using meta-data?
2) Is meta-data required to succeed?
3) Will the navigation speed be improved?

In order to begin answering these questions, examples of queries that would require the use of meta-data need to be determined.

In terms of OCR and machine translation, the main question is how does the accuracy affect precision and recall.

It was imagined that a unified dataset should be created for both the machine translation community and the OCR community. This may be achieved by supplying the OCR engines and the required machine translation engines in addition to a dataset produced for the segmentation and meta-data extraction problem. However, an Information Retrieval evaluation usually requires significantly more data than can be reasonably produced for the segmentation and meta-data extraction problem.

4.4 General comments

Many general comments were made involving the evaluation of these Document Image Understanding tasks.

First, for any dataset that is produced, an appropriate evaluation toolkit needs to be supplied with multiple metrics and comparison routines. In addition, these routines as well as any general data format management routines (including format conversions) need to be well supported and maintained as metrics and required meta-data evolve over time. It is hoped that the Government may be able to supply this service.

Second, whatever datasets are produced, future image domains need to be considered such as greyscale, color, and multiple resolutions. In addition, multilingual data is becoming more important.

Third, in order to reduce the amount of time and cost related to creating ground truth data, existing datasets should be extended whenever possible. For example, publishers may be able to provide existing datasets. Digital libraries may also provide datasets to be extended.

Finally, there were several lively and lengthy discussions on how evaluations should be conducted. Some of the views expressed during those discussions were as follows:

1) The final results of any evaluations should be kept anonymous; or there should be a signed (legal?) agreement that the results will not be passed on to prevent the promotion of one participant over another. Steps need to be considered that will discourage the evaluations from becoming an economic (marketing) competition. However, the technologies that would be evaluated in the various document understanding tasks discussed are not necessarily as mature as the OCR vendor products that were used in the UNLV-ISRI tests. Hence, the concerns for keeping results anonymous may not be the same or as relevant as those expressed during the previous OCR tests.
2) To allow these evaluations to mature properly, one organization needs to be consistently funded over time to facilitate consistent data preparation and evaluations.
3) When the statistics are gathered, comments about the data and the metrics need to also be solicited and reported with the results.
4) During the UNLV-ISRI tests the train versus test issue was seriously abused, and hence this led to the decision to do independent testing. The purpose of the evaluations needs to be absolutely clear such that a similar situation does not reoccur.

5 Datasets

After the results of the workshops had been gathered, it was decided by the DIWG that the Duplicate Document Detection and the Document Segmentation and Meta-data extraction evaluations would be pursued. To this end, several datasets are being worked on.

5.1 Duplicate Document Dataset

For the Duplicate Document Detection problem, actual document page images have been extracted from a DoD declassification effort. A set of over 1000 images have been obtained and are waiting final approval to be released as of the writing of this paper. Half of these images will be used in the first dataset release.

As these documents have been through the declassification process, selected sections have been removed. It was agreed upon in the workshops that this was not a significant problem in the evaluation of Duplicate Document Detection.

In addition, duplicates of the documents have already been removed by a semi-manual method within the declassification effort. Therefore duplicate document images will be simulated. Actual duplicates found in the declassification process have been visually examined to determine how to simulate reasonable duplicates. Exact duplicates will be simulated using various distortion methods and near duplicates will be simulated using annotations.

A concurrent effort is underway to extract actual duplicate document images from the declassification effort. The initial dataset described here is intended to allow the research community to get an initial feel for the problem faced in the declassification program while a more significant dataset is being obtained.

The initial release of documents are over 20 years old. The documents appear to be created using mostly typewriters. The duplicate documents were generated using the technologies of the time which include carbon paper, mimeographs, and manually retyped copies. Many of the documents are on colored paper.

The scanning process is performed by dedicated personnel who care that the document content is readable. Therefore most of the effects of colored paper, blurred content, and other natural distortions have been compensated for in the scanning process by manually adjusting the scanning. In addition, most of the document skew has been removed by both manual and automatic methods. In the case where multiple pages are scanned at once as with a book, the skew may be removed for one half of the image, but not the other. The initial dataset does not include many (if any) such examples.

The resulting images are 300 DPI, binary images as space was a concern in the design of the declassification program. The images will be delivered in TIFF format. The ground truth data will be delivered in an XML format; one file per image. The ground truth data will contain the results of an OCR engine, the results of Michael Cannon's quality measures [7], pointers to any duplicate documents, and all information about the duplication process for those images that are generated to represent duplicates. In addition, a file containing an NxN matrix will be included which denotes which documents are duplicates of other documents.

Finally, an evaluation toolkit will be delivered which will take as input the ground truth NxN matrix along with a set of NxN matrices generated by a duplicate detection algorithm. The tookit will produce various statistics and graphs of the results including hits versus false-alarms. It is intended that the results of this tookit will be used to demonstrate the results of any duplicate detection algorithm that is run against this dataset. This will allow anybody to compare the results of one algorithm against another even if the results are presented independently of one another. As this will be the first dataset of several to be released, the tookit and data formats can be improved as the research community provides feedback.

Since the dataset and the evaluation tookit were still being created at the time this paper was written, the contents of the initial delivery may vary slightly from the description given. Any discrepancies will be noted with the dataset delivered.

Additional datasets exist or are being created which may be used to supplement this data including data from the Gulf War declassification program (40K+ pages), data from the Department of Energy (144 pages) [7], and data being gathered by George Washington University (Dr. Richard Scotti) for the same purpose.

5.2 Page Segmentation Dataset

A dataset of 3000+ multi-lingual document page images gathered from the Library of Congress are currently in the process of being ground truthed. The languages include English, French, German, Spanish, Russian, Ukrainian, Japanese, and Chinese. They were scanned at 200 DPI using Fujitsu and HP scanners. For most of these images, the ground truth text has been entered and verified by linguists for the various languages. A breakout of the entire dataset can be seen in Appendix A.

A subset of these images (111 English document pages) were delivered in DAFS format [2] which included the text ground truth data with segmentation data at the page, line, and character levels. All of the page segments were tagged with their type such as title, text, figure, graphic, etc. All documents were tagged with their type such as magazine, newspaper, letter, etc.

This dataset was provided as a demonstration of the amount of detail that can be provided with a dataset to be used for page segmentation evaluation. There has been a good deal of research in the area of page decomposition [3-6] along with the creation of various datasets using various formats. In addition, the cost of creating the segmentation data for the 111 English document pages was on the order of a couple hundred hours of labor. Therefore, before the remaining data is provided along with the segmentation data, additional work will be done to determine exactly what meta-data is of interest.

Additional datasets exist which may be used to supplement this data including the NIST METTREC dataset [8], data collected by MathSoft (200 pages), and other well known datasets such as those created by the University of Nevada Las Vegas (UNLV), the University of Washington (UW), the Center of Excellence for Document Analysis and Recognition (CEDAR), and the University of Maryland (UMD).

6 Evaluations

Unfortunately, as of the date this paper was written, no evaluations have been completed using these datasets as they have not been completed. Initial evaluations will be done by DoD for several duplicate document algorithms. Evaluation results will be presented in a future paper.

Acknowledgements

Many thanks go to the members of DIWG, and to all of those who participated in the workshops. Special thanks to those who hosted the workshops, Jeff DeCurtins from the Stanford Research Institute (SRI) and Dan Lopresti from Lucent Technologies.

References

[1] I. Phillips, S. Chen, J. Ha, and R.M. Haralick. *University of Washington English/Japanese Document Image Database II CDROM.* (1995).

[2] T. Fruchterman, DAFS: A Standard for Document and Image Understanding. *Proceedings 1995 Symposium on Document Image Understanding Technology.* (1995) 94-100.

[3] L. Vincent and B. Yanikoglu, A Complete Environment for Ground-Truthing and Benchmarking Page Segmentation Algorithms. *Proceedings 1995 Symposium on Document Image Understanding Technology.* (1995) 70-83.

[4] R.M. Haralick, I. Philips, S. Chen, and J. Ha, Document Structural Decomposition. *Proceedings 1995 Symposium on Document Image Understanding Technology.* (1995) 27-38.

[5] D. Doermann, Page Decomposition and Related Research at the University of Maryland. *Proceedings 1995 Symposium on Document Image Understanding Technology.* (1995) 39-55.

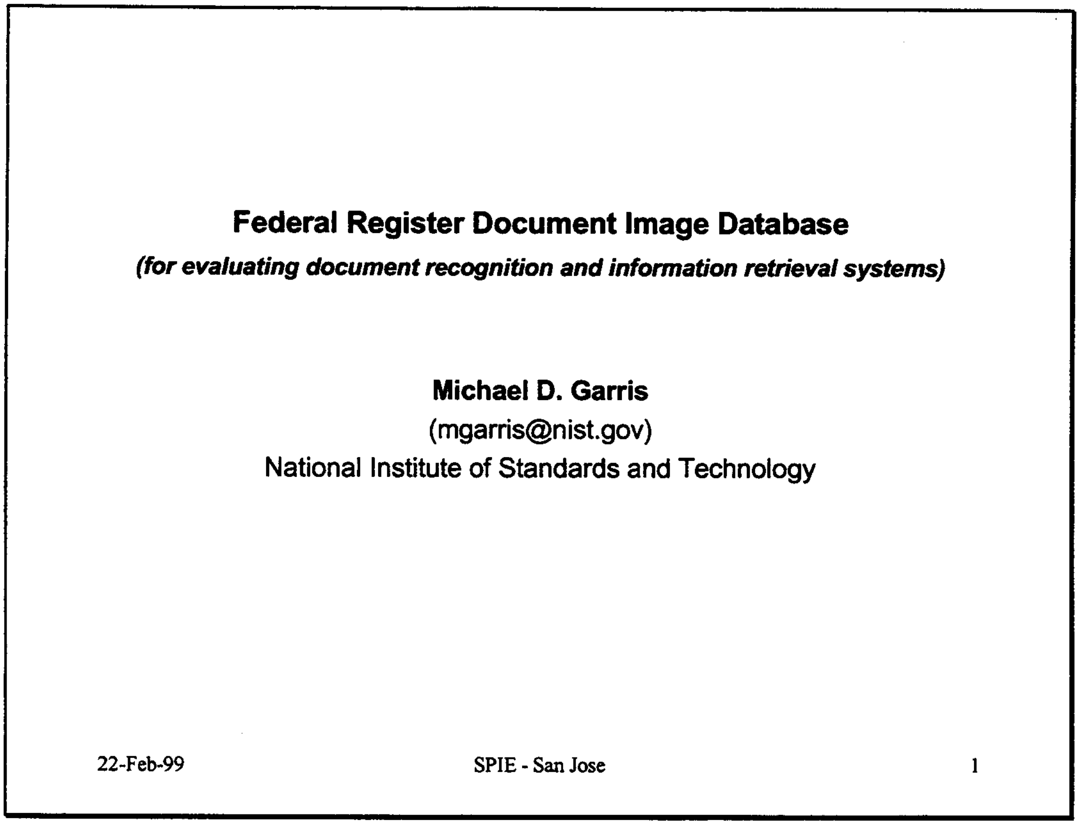
[6] S.L. Taylor, M. Lipshutz, and R.W. Nilson, Document Classification and Functional Decomposition. *Proceedings 1995 Symposium on Document Image Understanding Technology.* (1995) 56-69.

[7] M. Cannon, J. Hochberg, P. Kelly, and J. White, An Automated System for Numerically Rating Document Image Quality. *Proceedings 1997 Symposium on Document Image Understanding Technology.* (1997) 162-170.

[8] M.D. Garris and W.W. Klein, Creating and Validating a Large Image Database for METTREC. *NISTIR.* **6090** (Dec. 1997).

Appendix A

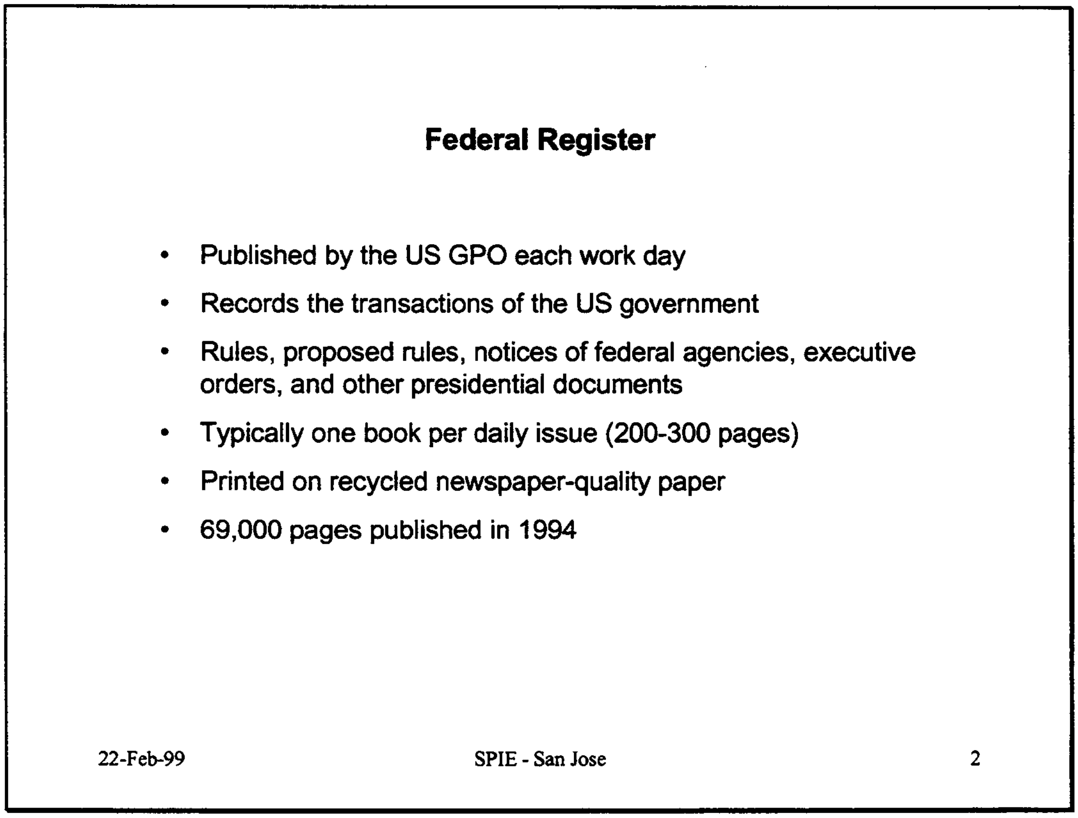
	Book	Letter	Mag.	Man.	News	Perio.	Other	Mix	Total
Eng.		1	60	79	268		1	22	431
Fr.	2	85	53	31	761		19	40	991
Sp.		112					36	52	200
Ger.		45	48				10	88	191
Ukr.	130			154	152	120			556
Rus.			125					74	199
Pol.		1							1
Ch.			68	26	264		5	142	505
Jap.			29						29
Total	132	244	383	290	1445	120	71	418	3103



Federal Register Document Image Database

(for evaluating document recognition and information retrieval systems)

Michael D. Garris
(mgarris@nist.gov)
National Institute of Standards and Technology



Federal Register

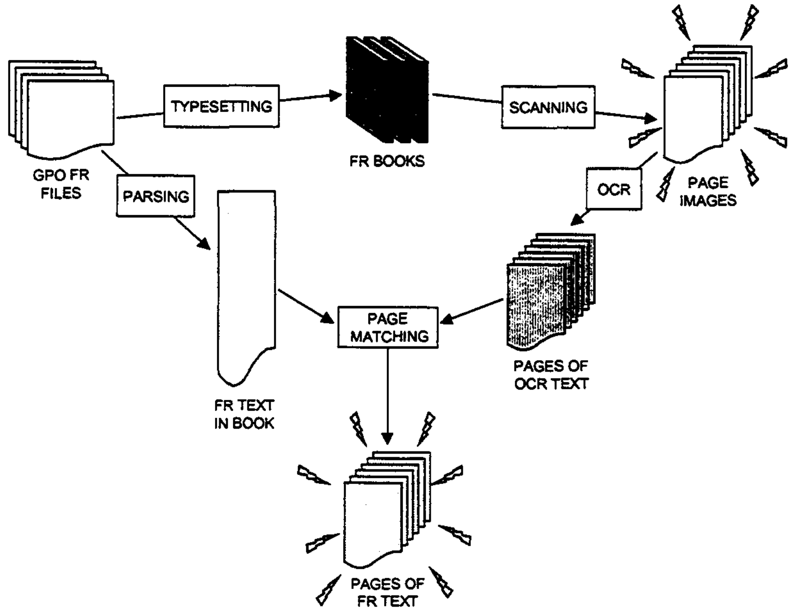
- Published by the US GPO each work day
- Records the transactions of the US government
- Rules, proposed rules, notices of federal agencies, executive orders, and other presidential documents
- Typically one book per daily issue (200-300 pages)
- Printed on recycled newspaper-quality paper
- 69,000 pages published in 1994

Federal Register Selected

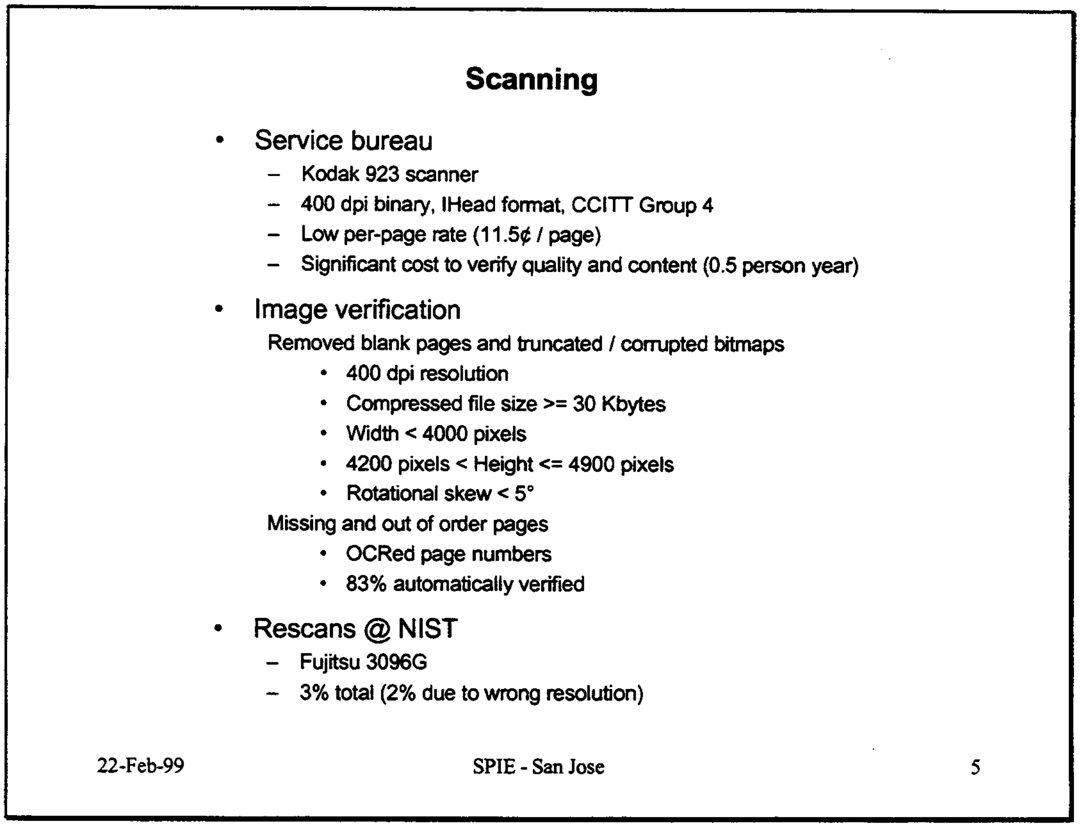
- Complete set of documents in the public domain
- Large collection (250 issues; 69,000 pages)
- Structured layout
- Significant variations in print and image quality
- Text stored in electronic typesetting files (ground truth source)

22-Feb-99 SPIE - San Jose 3

Ground Truth Production



22-Feb-99 SPIE - San Jose 4



Scanning

- Service bureau
  - Kodak 923 scanner
  - 400 dpi binary, IHead format, CCITT Group 4
  - Low per-page rate (11.5¢ / page)
  - Significant cost to verify quality and content (0.5 person year)
- Image verification

  Removed blank pages and truncated / corrupted bitmaps
    - 400 dpi resolution
    - Compressed file size >= 30 Kbytes
    - Width < 4000 pixels
    - 4200 pixels < Height <= 4900 pixels
    - Rotational skew < 5°

  Missing and out of order pages
    - OCRed page numbers
    - 83% automatically verified
- Rescans @ NIST
  - Fujitsu 3096G
  - 3% total (2% due to wrong resolution)

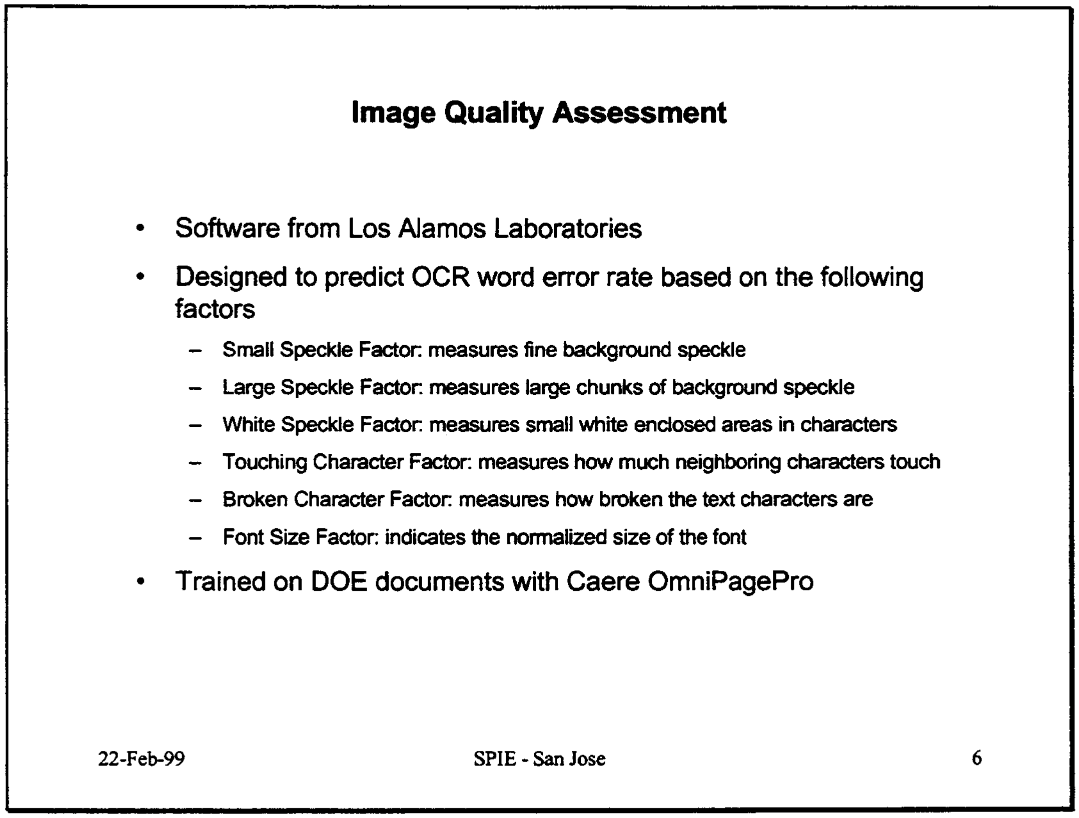
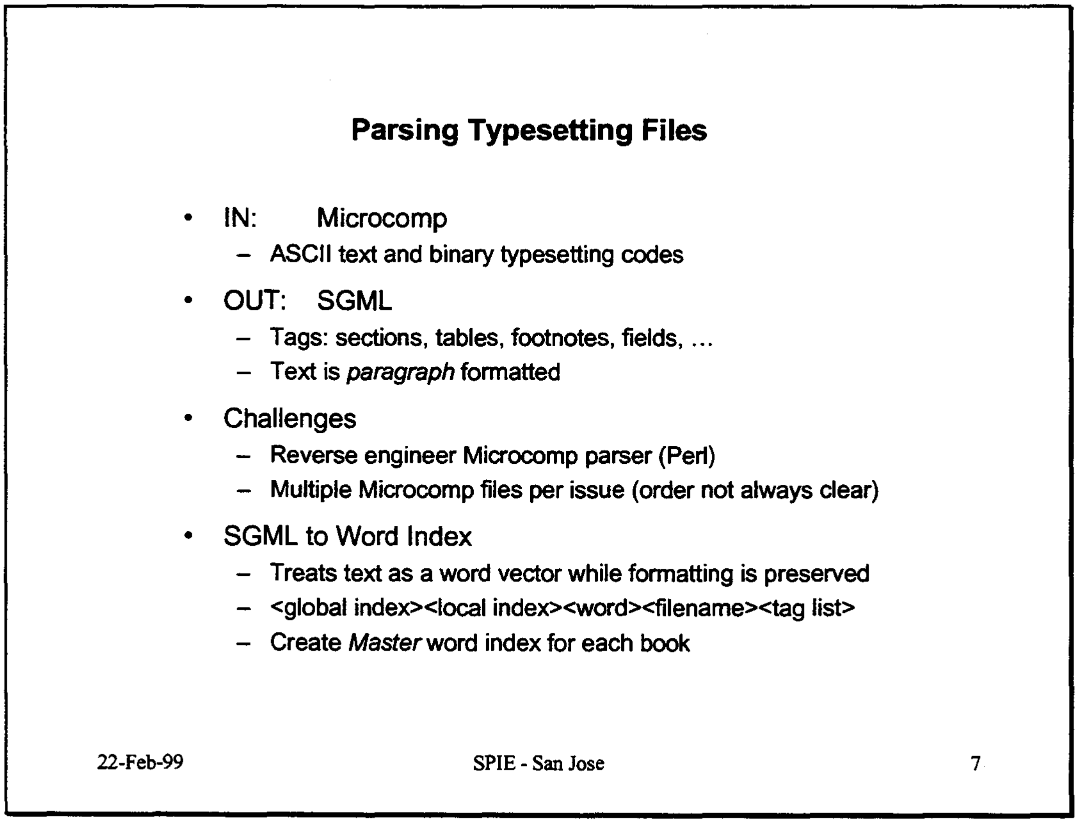


Image Quality Assessment

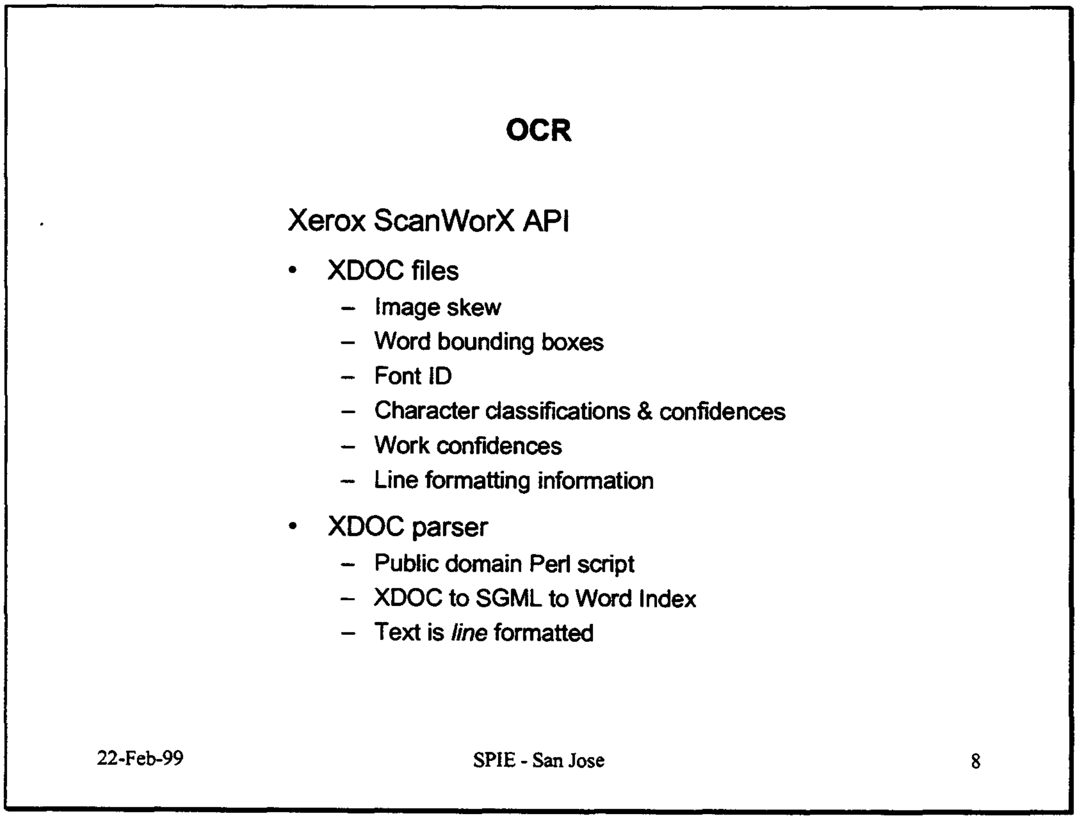
- Software from Los Alamos Laboratories
- Designed to predict OCR word error rate based on the following factors
  - Small Speckle Factor: measures fine background speckle
  - Large Speckle Factor: measures large chunks of background speckle
  - White Speckle Factor: measures small white enclosed areas in characters
  - Touching Character Factor: measures how much neighboring characters touch
  - Broken Character Factor: measures how broken the text characters are
  - Font Size Factor: indicates the normalized size of the font
- Trained on DOE documents with Caere OmniPagePro



Parsing Typesetting Files

- IN: Microcomp
  - ASCII text and binary typesetting codes
- OUT: SGML
  - Tags: sections, tables, footnotes, fields, …
  - Text is *paragraph* formatted
- Challenges
  - Reverse engineer Microcomp parser (Perl)
  - Multiple Microcomp files per issue (order not always clear)
- SGML to Word Index
  - Treats text as a word vector while formatting is preserved
  - <global index><local index><word><filename><tag list>
  - Create *Master* word index for each book

22-Feb-99 SPIE - San Jose 7



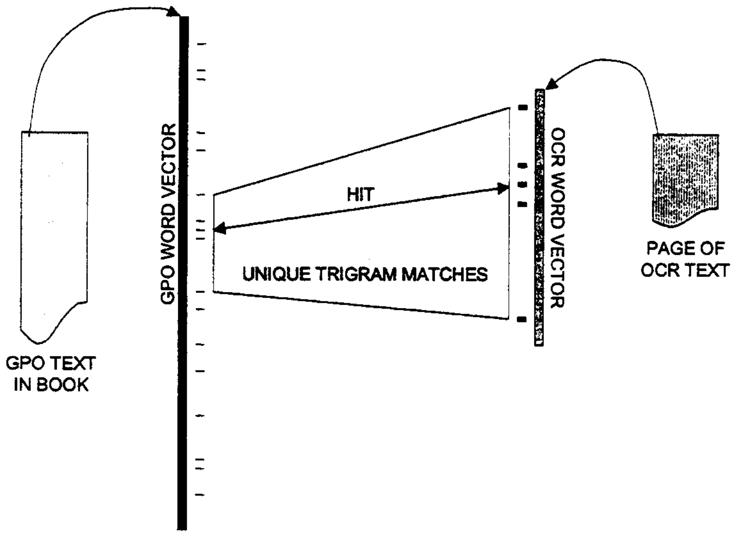
OCR

Xerox ScanWorX API

- XDOC files
  - Image skew
  - Word bounding boxes
  - Font ID
  - Character classifications & confidences
  - Work confidences
  - Line formatting information
- XDOC parser
  - Public domain Perl script
  - XDOC to SGML to Word Index
  - Text is *line* formatted

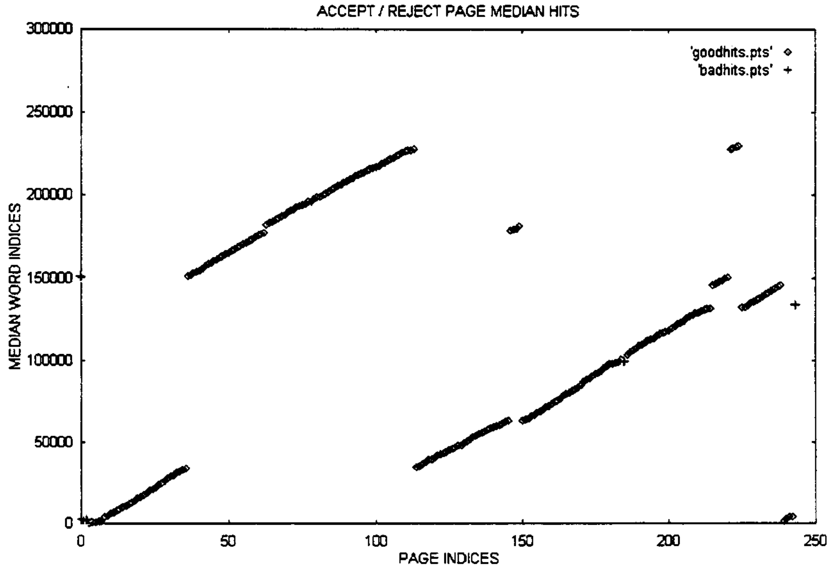
22-Feb-99 SPIE - San Jose 8

Locating Page in Typesetting Text



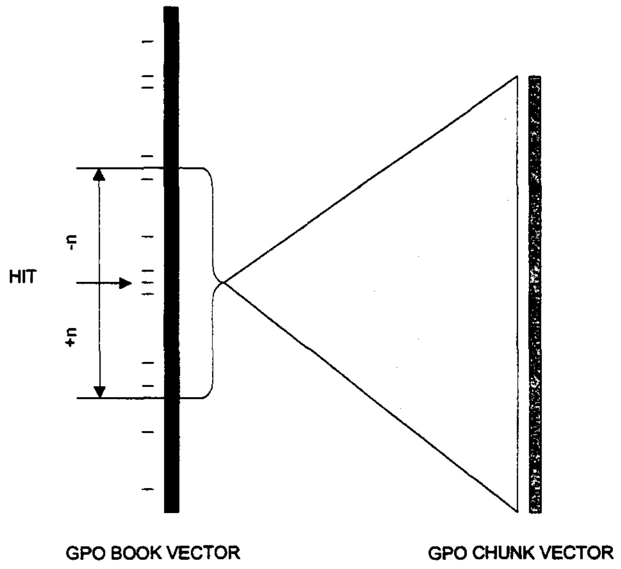
22-Feb-99 SPIE - San Jose 9

Rejecting Bad Page Locations



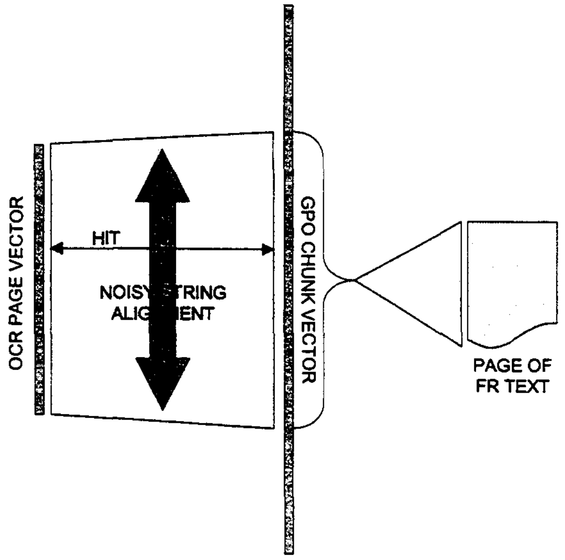
22-Feb-99 SPIE - San Jose 10

Extracting Ground Truth Chunk



22-Feb-99 SPIE - San Jose 11

Trimming Ground Truth



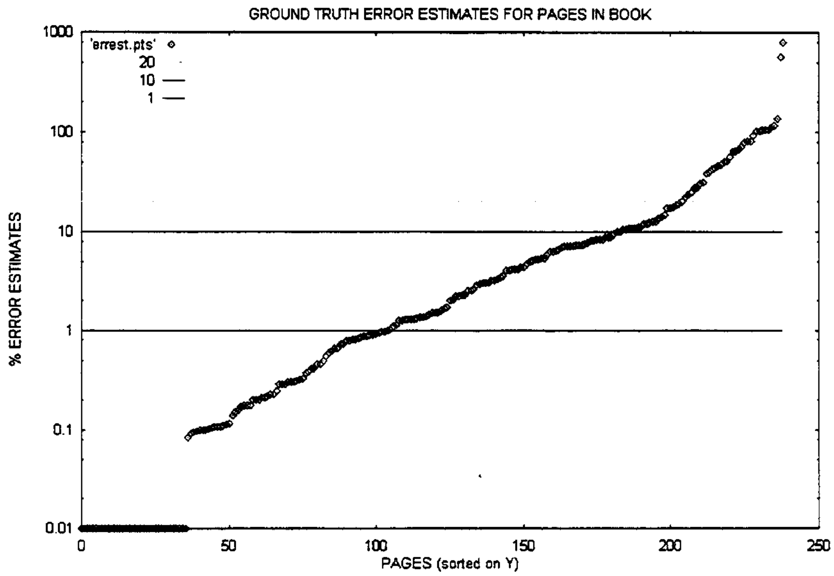
22-Feb-99 SPIE - San Jose 12

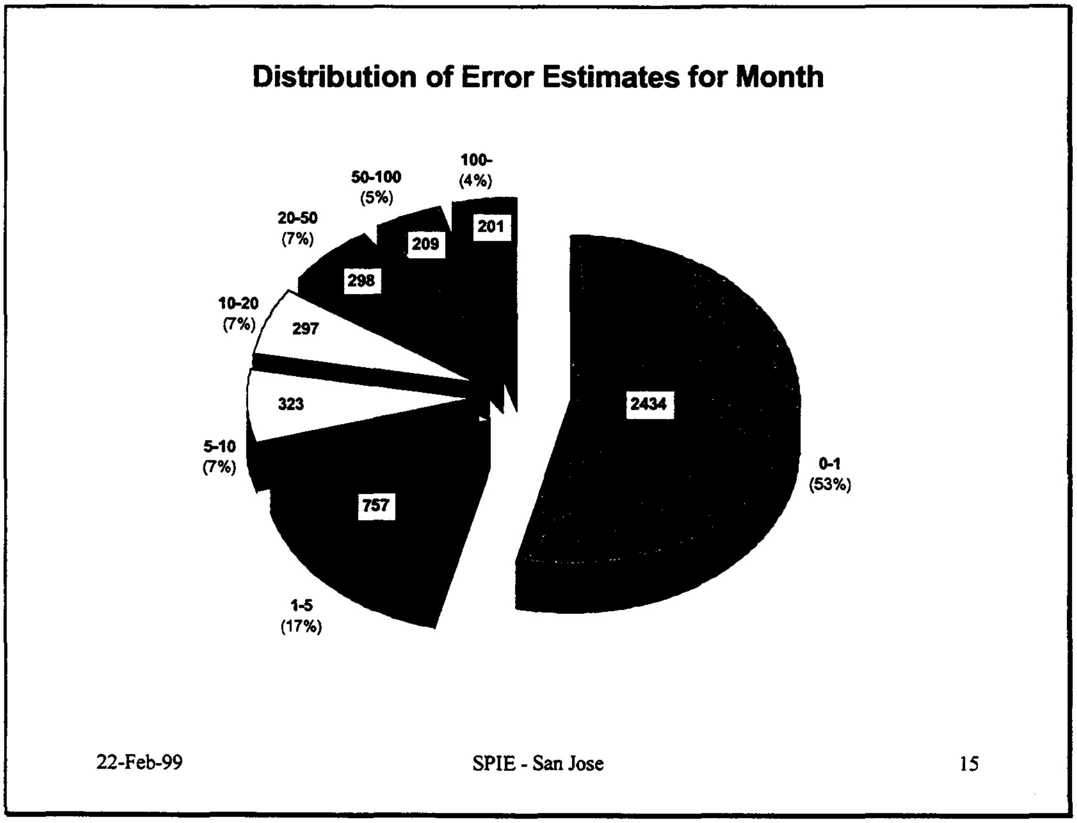
Estimating Error

- Accurately determining page boundaries is difficult
  - Text ordering, missing text, OCR quality, ...
- Reject mechanism unsatisfactory
- Conservative error estimate
  - Realign trimmed ground truth with OCR text
  - Match out of order blocks of text
  - Remaining inserted and deleted words represent how well the ground truth *covers* the OCR text

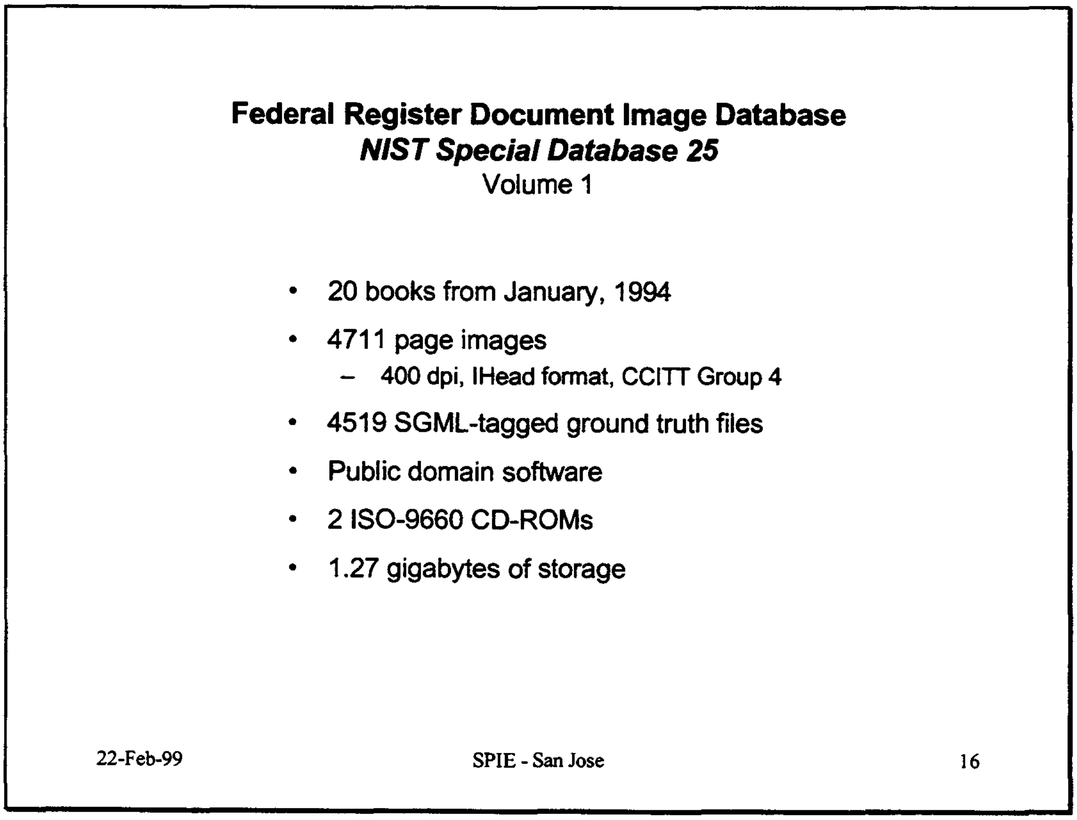
$$error = \frac{(ins + del)}{nwords}$$

Error Estimates for Book



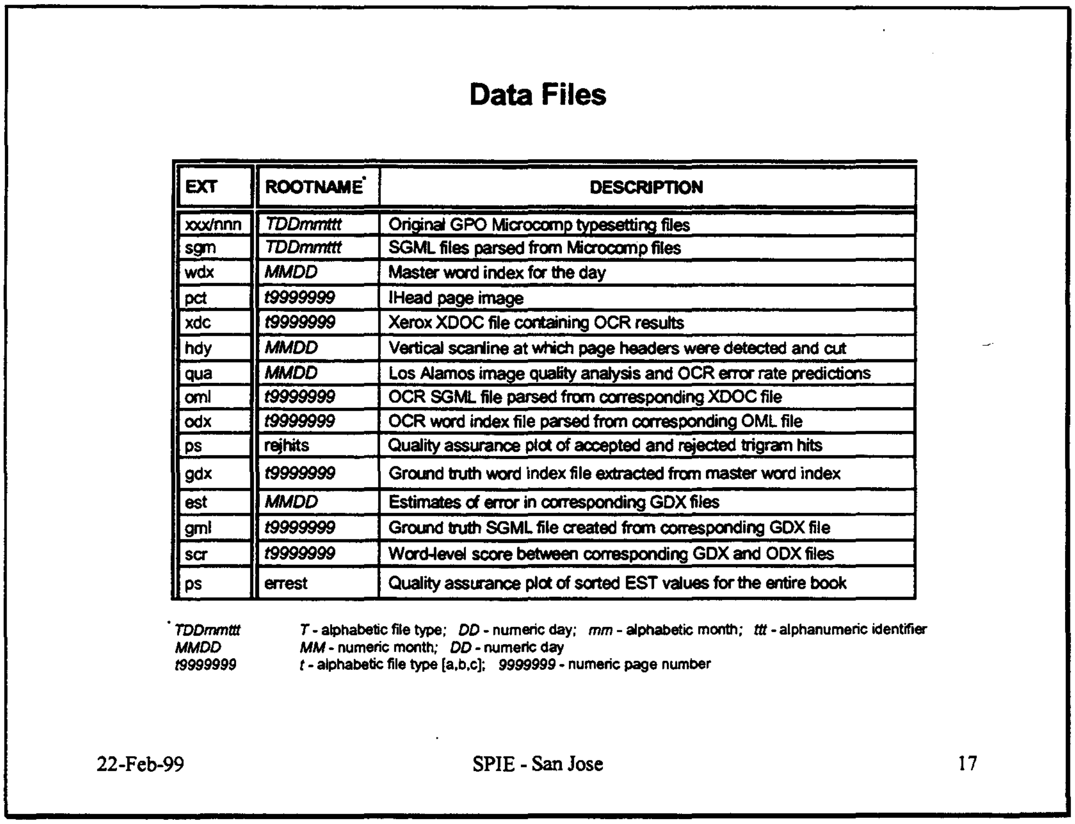


Distribution of Error Estimates for Month



Federal Register Document Image Database
NIST Special Database 25
Volume 1

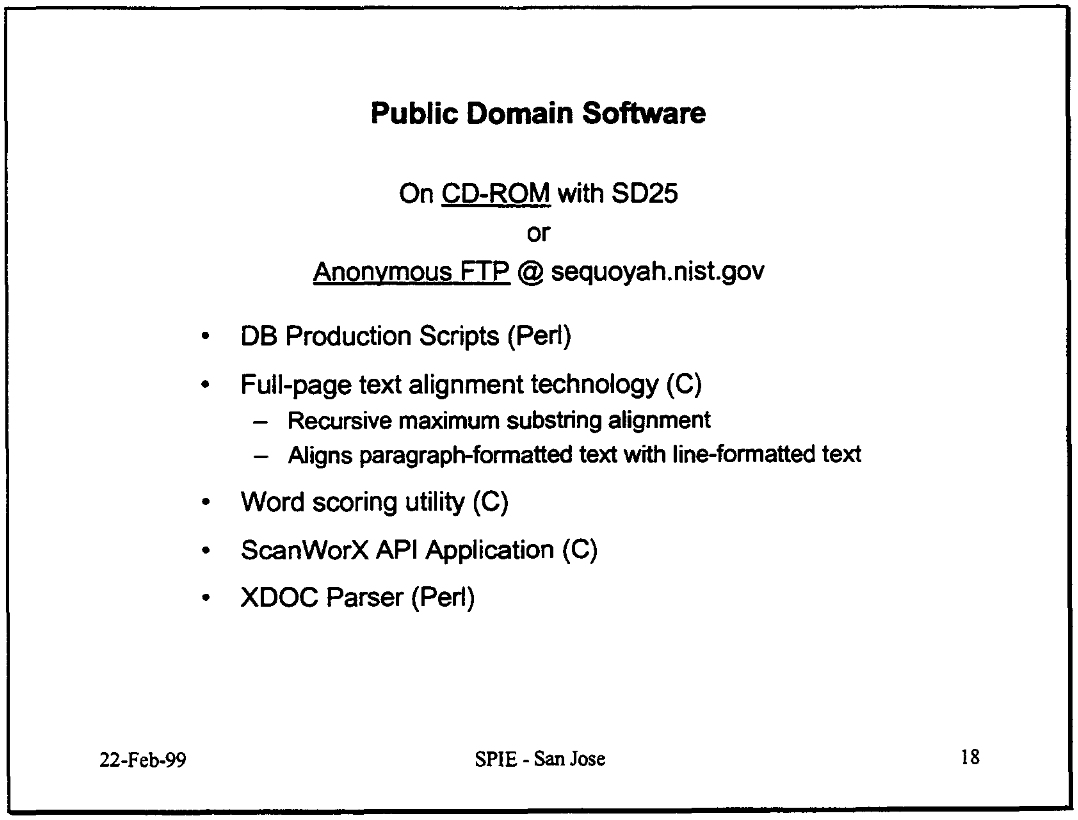
- 20 books from January, 1994
- 4711 page images
  - 400 dpi, IHead format, CCITT Group 4
- 4519 SGML-tagged ground truth files
- Public domain software
- 2 ISO-9660 CD-ROMs
- 1.27 gigabytes of storage



Data Files

EXT	ROOTNAME*	DESCRIPTION
xxx/nnn	*TDDmmttt*	Original GPO Microcomp typesetting files
sgm	*TDDmmttt*	SGML files parsed from Microcomp files
wdx	*MMDD*	Master word index for the day
pct	*t9999999*	IHead page image
xdc	*t9999999*	Xerox XDOC file containing OCR results
hdy	*MMDD*	Vertical scanline at which page headers were detected and cut
qua	*MMDD*	Los Alamos image quality analysis and OCR error rate predictions
oml	*t9999999*	OCR SGML file parsed from corresponding XDOC file
odx	*t9999999*	OCR word index file parsed from corresponding OML file
ps	rejhits	Quality assurance plot of accepted and rejected trigram hits
gdx	*t9999999*	Ground truth word index file extracted from master word index
est	*MMDD*	Estimates of error in corresponding GDX files
gml	*t9999999*	Ground truth SGML file created from corresponding GDX file
scr	*t9999999*	Word-level score between corresponding GDX and ODX files
ps	errest	Quality assurance plot of sorted EST values for the entire book

* *TDDmmttt*	*T* - alphabetic file type; *DD* - numeric day; *mm* - alphabetic month; *ttt* - alphanumeric identifier
MMDD	*MM* - numeric month; *DD* - numeric day
t9999999	*t* - alphabetic file type [a,b,c]; *9999999* - numeric page number



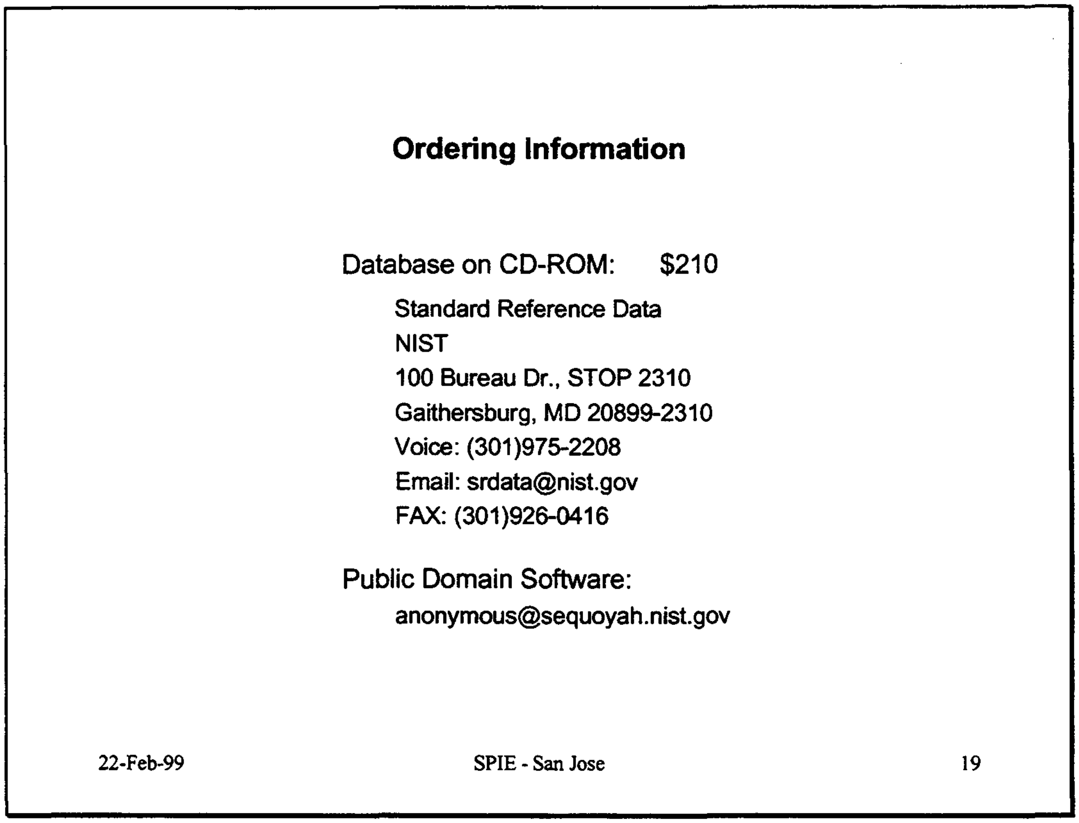
Public Domain Software

On CD-ROM with SD25

or

Anonymous FTP @ sequoyah.nist.gov

- DB Production Scripts (Perl)
- Full-page text alignment technology (C)
  - Recursive maximum substring alignment
  - Aligns paragraph-formatted text with line-formatted text
- Word scoring utility (C)
- ScanWorX API Application (C)
- XDOC Parser (Perl)



Ordering Information

Database on CD-ROM: $210

Standard Reference Data
NIST
100 Bureau Dr., STOP 2310
Gaithersburg, MD 20899-2310
Voice: (301)975-2208
Email: srdata@nist.gov
FAX: (301)926-0416

Public Domain Software:

anonymous@sequoyah.nist.gov

Additional Submissions

Similarity Analysis and the Mosaic Effect

Dr. Paul S. Prueitt
Senior Scientist, NetBase Corporation
and Director, BCN Group Inc.

Abstract

A preliminary theory and notation for similarity analysis is outlined with application to the reduction of a mosaic effect observed in declassified collections.

1.0: Definition of mosaic effect

The syntactical mosaic effect occurs when structural parts of a single image or text unit are separated into disjoint parts, each part judged not to have a certain piece of information but where the combination of two or more of these units is judged to reveal this information.

The semantic mosaic effect occurs when structural parts of a single image or text unit are separated into perhaps overlapping parts. Each part is judged not to imply a certain concept but the combination of two or more of these units is judged to support the inference of this concept.

As a general rule, an increase in similarity analysis causes a decrease in the mosaic effect.

This relationship, between the quality of similarity analysis and the occurrences of mosaic effects, is one of three fundamental relationships that argue for a change in the nature of the discussion about computer mediated knowledge management and the use of computers to provide situational analysis. It's application to the management of classification by the Federal Government would make tractable tasks mandated by Executive Order 12958.

Additional fundamental relationships open the possibility of defining notational systems that provide a control language for the declassification process, or more generally for a new "horizontal" information technology supporting "conceptual checkers". These new tools are applied to text and are based on the identification and interpretation of conceptual substructures referenced by the text. Conceptual checking will work like a spell checker or a grammar checker.

The relationships between concepts, their similarities and dissimilarities, are essential to such a technology. However, context limits the scope of notation in a concept space, and thus it is essential to account for context in the notation. The interplay between substructural similarities and properties of interpretation is thus to be seen as an evolutionary process, that can be controlled via an annotation language such as the one developed by Tonfoni [1].

1.1: Similarity and inference: There is a fundamental relationship between similarity analysis in text and images and the mosaic effect seen in declassification releases. The effect reveals information that, when gathered together, support inferences and strong evidence that certain things are true.

For example, an agent may feel that a spy is working in a certain area. A mosaic effect might provide sufficient evidence from declassified material for conjecturing the exact identity of the spy. Whereas the declassified material does not explicitly identify the spy, the material does lead to an identification that might not have otherwise occurred.

At the heart of the mosaic effect there are different types of similarity.

The type of similarity that we address below is a relationship between causes of, or the properties of, a class of two or more situations. This type of similarity depends on a part to whole relationship that is exploited in various bi-level voting procedures and is discussed in various literatures (including the work by D. Hofstadler and his students) [2]. The voting procedure was developed, by the author, from an interpretation of J. S. Mill's logic and a Russian cybernetic system [3].

The following is some preliminary notes on the grounding of a generalization of the Duplicate Document Detection (D3) formalism [4]. It shows the importance of both connectionist and evolutionary programming and the paradigms that support their analysis. The development of the notation should proceed with the peer review and contributions of several scholars in related areas of research.

1.2: Conceptual foundation, based on bi-level mathematics: Let S2 be the sign system for all basins of attraction for an "integrated" system of oscillating point sources of magnetic flux [5]. These basins of attraction are a structural cause for a number of system behaviors such as when a group, of source points, comes to share the same phase of oscillation. Phase locking is called entrainment and is a manifestation of events where two of more point sources act together as an integrated whole. The sign system S2 is made specific with a one to one correspondence between symbols and basins.

The emergent phenomena are basins of attraction in a manifold that develops either in simulation or in physical reality. In one class of simulated systems, the basins of attraction of a simple layered artificial neural network represents emergent phenomena.

However, a model of weakly coupled oscillators [5] is a clearer model of bi-level computation since the basic elements are each something like a physical pendulum. Linkage relationships, between point sources, can be specified in either the physical systems or in the simulations. Systems of weakly coupled oscillators give us a means to verify hypothesis about physical systems with emergent properties.

The mosaic effect can be studied in these simulations, since a conjecture about a fact corresponds to the formation of a basin signed by the sign system S2. The related theories on deduction and induction may be grounded in neurophysiology [6] and thus has long term validity as a motivation for basis research on text and image understanding.

Let S1 be the sign system for the set of point sources. These point sources provide a model of a simple type of substructure. The bi-level model considers the elements, modeled by the notation in S2, to be an aggregation of elements from the sign system S1. However, the "entanglement" of the elements of cognition, memory substructure, is not nearly so dominate as the bi-level model suggests.

One finds that removing elements from S1 does not modify only one or a few basins – but rather modifies many or all of them. The "entanglement" in this model is not natural. The bi-level model is not powerful enough. A 'top-down" environmental level is needed to establish context.

In a bi-level model, the sign system S2 represents the "ultrastructure" of the emergent manifold, but this ultrastructure does not have ontological referent outside of the happenstance of the manifold. It is merely an artifact of the binding of the elements of substructure into one whole. We need a top down fitness function.

The fitness functions, in evolutionary programming, cause the configuration of basic elements to evolve towards some implicit representation of an ecosystem. With the bi-level model, we see that a fitness function is indeed defined, but in an ad hoc fashion. The fitness function fits the basin – without putting pressure on the basin to change. There is no "action-perception" cycle.

The evolutionary, or adaptive, pressure must come from a measurement process that actually is open to the complex nature of the environment. In the case of declassification annotation and judgments, the required openness is to the cognitive processing of the analyst.

In the bi-level model, each emergent phenomenon is not co-selected by substructure and ecosystem affordance, as is the case in natural systems having a specific chemical distribution in an environment, or a set of behaviors in a real behavioral space. We need a third level to supply context and which puts pressure on basins to change.

The way to extend the bi-level model to a tri-level model is developed in C. S. Peirce's "Unifying Logical Vision" (ULV), as interpreted by modern research on situational logics, knowledge representation and neuropsychology [3, 5, 6].

1.3: Unifying Logical Vision (ULV): The ULV is stated in the following way:

"Concepts are like chemical compounds, they are composed of atoms"

This vision was instrumental in the development, by D. Pospelov and V. Finn (1970-1995), of the theory of situational control. The issue addressed by this Russian research team was the role of environmental factors in determining which of many possible basins are actually manifest in a natural system at any one time. This role introduces a third level to the organizational stratification of the theory.

These levels are "real" levels separated by "gaps", not the hierarchical levels seen in subsumption graphs and tree data structures.

For example, since the third level of analysis is about things that are at a higher time scale, the environmental role is seen in incomplete sets of rules of behavior rather than all at one time.

The properties and relationships between basins, in so far as they are understood, are represented in the situational logic produced for the sign system. The Russian logic has the form of five notational languages, two inner languages and three outer languages that progressively build up the situational logic having special properties related to an openness to change in the axioms and inference rules. In our interpretation of the Russian work, the two inner and three outer languages are with respect to a gap that necessarily separates structure from function in natural systems. It is a gap that separates S1 from S2.

2: The Tri-level notation

After the grounding of similarity analysis in the previous section, we now develop a specific notation for similarity analysis in document and image collections.

The relationship between this notation and the 4 by 4 D3 formalism [4] is one of generalization from four absolute levels { page segment, page, document, collection} to three relative levels,

{ substructure, middle, contextual }.

If one considers the four levels of the D3 formalism, we see that each of the middle two levels are each between two levels. In this case, any analysis of similarity has three levels. For example, the page's substructure is non-overlapping page segments, and the page's context is held within the document. Page segments can be given a n-gram substructure, but this was not done in the 4 by 4 D3 formalism.

Four types of similarity metrics, {exact, near-exact, non-near exact but similar, different}, are also generalized to a class of N similarity metrics,

{exact, near-exact,
similar through relationship r, different}.

The cardinality of this class of metrics depends on the number of relationships, { r }, that are active in the collection. We suggest that this number is not constant from context to context, and that this fact represented a primary obstacle that has not been addressed in any of the large-scale projects. Since a shift in context requires a shift in time, the formalism is called the ***N(t) by 3 SA formalism***.

2.1: On the issue of descriptive enumeration and relations

Let

$$A = \{ (a, r, b) \}$$

be the set of active relationships in S1 at time t_0. These active relationships are determined empirically to be some subset of all potential relationships

$$P = \{ (a, r, b) \}.$$

Potential relationships are also to be determined empirically. The requirement that relationships be determined empirically is a harsh one, but one required by the nature of phenomenon.

A method for determination of sets of relationships can be advanced and thus the harshness of requiring empirical determination of all possible relationships is tractable. This method is descriptive enumeration.

The set of active prototypes in substructure can be determined by a descriptive enumeration of invariance. By invariance we mean those patterns that form equivalence classes seen from the perspective of the middle level. From descriptive enumeration, each invariance is assigned a symbol. As this set is being developed, it is possible to consider all subsets of size 2. Each of these subsets {a, b} define a certain number of potential relationships (a,r,b). These relationships are identified through a process of descriptive enumeration, this time about the class of relations.

The process of descriptive enumeration requires strong support from human decision makers, since any formal method relying entirely on computational process is likely to be intractable.

Active relationships, between substructure, are instantiated in the context of building an ensemble at the next higher level of organization. Of the three levels, this ensemble level is the middle level. The formation of judgement is normally about the objects and relations in a middle level.

The middle level is defined by a set of temporal invariance; e.g., objects having permanence in some interval of time, interacting with each other. Substructure and context do not interact with ensembles, by definition. In fact, a "level" is properly defined to be the whole class of all objects that interact with each other. From this definition, it follows that each level is separated by an "gap'. The complex systems research community calls this separation an "epistemic" gap. Physicists call a class of such gaps Heisenburg gaps, and mind – body philosophers call certain type of gaps Cartesian gaps. In each case, it is commonly thought that it is not possible to fully formalize the gap's nature.

Seen from the perspective of the middle level, substructure is a statistical artifact where individual substructure invariance is treated as a member of a prototype class. This principle is illustrated by the regard that a chemical compound has for an individual atom, or a factory has for an individual worker. It may be appealing to say substructure has the relationship of "is a part of" to the ensemble. However, this use of language is simply not correct. There is an incompleteness of description that can not easily be overcome.

Likewise, context has only incomplete relationships to ensembles since context is only partially a function of the environment. Again, we have an incomplete description, but of a different kind. Now the incompleteness has to do with the waiting time required for patterns to complete. Sometimes the incompleteness of description is not a problem, but at other times it leads to errors.

The definition of level comes from an appeal to the physics of complex systems, particularly the physics of quantum events. However, the model of three levels {substructure, middle, contextual} is relative. Any specific substructure level might be a middle level seen from a different perspective and the same may be said for the level of context.

2.2: Temporal dimension

Each "real" object has a period of consistency where the object maintains it's "temporal invariance". In consideration of the properties of a level, the temporal dimension is essential. The formation event introduces a new object within a level that existed prior to the emergence. Once created, the object has a stable existence before suddenly losing its cohesiveness.

Once formed, the whole may be modified by new internal emergence. However, the level shapes the emergence and thus the temporal state of the level is reflected in the properties that the new object has.

There is an assumption that the set of all objects that have an active relationship is invariant over some period of time starting at t_0 and lasting until t_1.

Let t_0 be when an ensemble is first formed. We need for the sign system, S2, to have a one to one correspondence to those objects in the level that have active relationships to the new ensemble.

The selected set, S2, is

$$\{ o_1, o_2, o_3, \ldots, o_n\},$$

over the period from t_0 to t_1. Each of the objects in this set have a set of properties { p } and relationships { r } to other objects. These properties and relationships are to be discovered.

2.3: Tri-level notation

The tri-level notation must account for three issues.

1) The first issue is the emergence of a level, or of a new object into an existing level, that is seen as an interaction between substructure and context.

2) The second issue has to do with the entanglement of substructural invariance in the objects of a level.

3) The third issue has to do with the interpretation of signs as referential to specific concepts.

These three issues are treated in the following three subsections.

2.3.1: The emergent manifold and its invariance: Intellectually, the problem we face is explaining how levels come into existence in the first place. In section 1, we grounded our discussion of similarity in a model of weakly coupled oscillating point sources. This model has several unique qualities. First, it is possible to perform experiments with either a physical apparatus or within a computer simulation. Second, the bi-level nature is illustrative of the ensemble behavior of network models of neural networks or of evolutionary programming like genetic algorithms. Thus we have several ways to motivate a deep and empirically grounded discussion about the emergence of a level.

If our model was only bi-level then there would be no level, only an emergent "object". However, any object implicitly defines a level as the set of all objects that it interacts with. In many formal theories, the emergent object is considered alone – without taking into account the implicitly defined level that forms the "full" environment of the object. We feel that this practice is motivated by the avoidance of "necessary" logical paradox.

Sometimes this paradox can be stated: "there are things that do not exist". In particular, the paradox comes up when one talks about the existence of memory when the memory is not actually, at that moment, the contents of an awareness state.

We use the term ‘full environment' here to remind us that the level is not statically defined and has a temporal dimension.

2.3.2: Entanglement of substructural invariance in the objects of a level: Perhaps the best example of entanglement is seen when we attempt to represent the referent concepts signed by text or discourse. The boundary of a concept referentially is just not as crisp as classical logic would have us believe. It is not always possible to say that a concept is either present or not present in a passage of text.

One reason for non-crisp delineation comes from the nature of physical mechanisms that produce mental images. These mechanisms may operate in a three-tiered modality, again due to the underlying qualities producing physical stratification of temporal processes into three levels with gaps.

2.3.3: Interpretation of signs as referential to specific concepts: Presumably concepts were occurring within the mental images of humans before natural language came into it's modern form. One imagines that specific types of behavior were coincident with the presence of a concept. Later, these specific types of behavior were part of the substance of sign systems and then language systems.

Interpretation of signs then became the basis for communication of information. The tri-level model of complex processes, including concept formation, is thus seen in the light that Peirce and Popper used in their analysis.

For Peirce the three levels correspond to an interpretation within context, the sign system itself as the middle level, and the objects of the world as the substructure.

For Popper the three levels are a subjective experience of reality in a present moment context, the collective knowledge that has come to exist as a historical heritage, and the objects of the world.

Conclusion

The author has described the minimal complexity required to understand a technical solution to the mosaic effect in large bulk declassifications. It is argued elsewhere that this minimal complexity is required, given that the Nation acquire the technical capability to manage the national secrets in accordance with statute and with the Constitution.

References

[1] Tonfoni, G. (1999). On Augmenting Documentation Reliability through Communicative Context Transport. In SDIUT 99 Proceedings, this volume.

[2] Hofstadter, D. (1995). Fluid Concepts and Creative Analogies. Basic Books.

[3] Prueitt, P. (1998). An Interpretation of the Logic of J. S. Mill, in IEEE Joint Conference on the Science and Technology of Intelligent Systems, Sept. 1998.

[4] Prueitt, P. (1999) The 4 by 4 Duplicate Document Detection Algorithm, in this volume.

[5] Kowalski, J, Ansari, A, Prueitt, P, Dawes, R and Gross, G. (1988). On Synchronization and Phase Locking in Strongly Coupled Systems of Planar Rotators. *Complex Systems*, 2, 441-462.

[6]. Prueitt, P. (1997). Grounding Applied Semiotics in Neuropsychology and Open Logic, in IEEE Systems Man and Cybernetics Oct. 1997.

Automatic Reformatting of OCR Text from Biomedical Journal Articles

Glenn M. Ford, Susan E. Hauser, George R. Thoma
National Library of Medicine
Bethesda, Maryland 20894

Abstract

The goal of the Medical Article Record System (MARS), being developed by the National Library of Medicine, is to reduce the manual keyboard entry of bibliographic citation fields for the MEDLINE database by automatically identifying and converting information from bit-mapped images of biomedical journal article pages to ASCII data. An important element of this automatic conversion requires reformatting the title, author and affiliation fields from the output of the Optical Character Recognition (OCR) process in MARS to the formats specified by MEDLINE conventions. This paper outlines the methods developed to implement the reformatting process.

1 Introduction

The MARS system in its first version now operating at the National Library of Medicine automatically extracts article abstracts from the bitmapped images of journal articles, but relies on the manual keyboard entry of all the other fields required in the MEDLINE database. A second generation MARS system is being designed to automate the entry of other fields, focussing primarily at present on the article title, author names and their institutional or organizational affiliation. Following the scanning stage, the OCR system converts the image contents to text, and algorithms segment the page image (autozoning), and automatically label the zones. Reformatting follows the zone labeling stage so that the zone contents adhere to MEDLINE's syntactic conventions.

1.1 Institutional Affiliations

Institutional affiliations of the authors are reformatted by finding the best match between the OCR text and a list of about 130,000 correctly formatted affiliations obtained from the current production version of MARS. Simple string matching is not promising because of the myriad arrangements in which affiliations can be expressed. Most journals show the affiliations of all authors, but by convention only the affiliation of the first author is entered into MEDLINE. However, the text string corresponding to the first affiliation may be scattered throughout the OCR text for the affiliation field. As an example, when multiple authors are affiliated with different departments within the same institution, the printed affiliation may be "Department A, Department B, Department C, Institution XYZ," while the correct MEDLINE entry is "Department A, Institution XYZ." The problem is further confounded by OCR errors, especially errors in detecting superscripts and subscripts. To find a match, the entire OCR text of the affiliation field is compared with every entry in the list of existing affiliations. A matching score for each of the existing affiliations is calculated on the basis of partial token matches, distance between token matches and customized soundex matching. The three highest scoring candidates are presented to the "reconcile" (verification) operator for selection. In preliminary tests, our current version of affiliation field reformatting successfully identifies the correct affiliation over 80% of the time when the affiliation is represented in the list. This success rate is expected to improve with parallel efforts to reduce OCR errors and the expansion of the list of affiliations from ongoing production data.

1.2 Article Titles and Authors

The reformatting of author and title fields is implemented by predefined rules. Based on journal title and field identification (author or title), the software selects a subset of rules from the inclusive set of all rules. The selected rule set and the OCR text are passed to the implementation algorithm. As each rule is applied, the OCR string is modified. Rules for title fields involve initial-letter capitalization and all-letter capitalization. Rules for author fields include characters used to delimit authors in a multiple-author list; tokens to be removed, such as Ph.D.; tokens to be converted, such as II to 2nd; and particles to be retained, such as "van." For example, Eric S. Van Bueron, Ph.D.

becomes Van Bueron ES. Our preliminary version of title and author reformatting correctly reformats more than 97% of the authors and titles from a test set of 1857 processed articles. We expect performance to improve with the addition of rules derived from production data.

2 Reformatting the Author field

Reformatting the author field uses *forward chaining*[1] rules based deduction. The reformat module can have many rules defined for a particular field. Each rule has a number of requirements among which are that it must

- Be associated with a specific ISSN number (Journal Title)
- Fall into one of eight categories. The categories are pre-defined in the reformat module and are required to help in our conflict resolution strategy, which in our case is *specificity ordering*. Whenever the conditions of one triggering rule is a superset of another rule, the superset rule takes precedence in that it deals with more specific situations. An example of this is shown later. The eight categories and examples are listed in Table 1 shown in Appendix A.

The example column in Table 1 shows the complete reformatted field. Note that a single rule or category does not necessarily complete the reformatting, but may need to be combined to achieve correct reformatting of the author field.

With the eight categories defined, the first step in using the reformat module for a given ISSN is to define which rules are appropriate for a particular ISSN (or journal title), since the printed format varies widely among journals. As an example, in one journal the authors appear as:

> Glenn M Ford, MD, John Smith, PhD, and John Glover

This can be difficult to parse with a default set of rules, such as ', and' and ',' so that other rules need to be defined. By defining, in the database, the rules for a specific Journal Title over a specific period[2] of time we can customize the rules to work for unusual or specific cases.

The above example fails in the default rule set that only has ',' and ', and' as the Author Delimiter because this would incorrectly identify 'MD' and 'PhD' as author names. To accommodate this journal (and others like it) a high priority rule trigger list was created for Author Delimiters such as ', MD', ', PhD', 'Mr.', 'Dr.', and other formal titles.

To avoid conflict among rules each word chain is passed through all the categories recursively until no more rules are triggered. As long as we have an antecedent with consequences we continue to process the word chain. Using the forwarding chaining method, when an if statement is observed to match an assertion, the antecedent (i.e., an if statement) is satisfied. When the entire set of "if statements" are satisfied, the rule is triggered. Each rule that is triggered establishes, in a working memory node, that it was executed. During conflict resolution the reformat module decides which rules take priority over others via specificity ordering. An example would be:

Reduce category executes on 'John Smith II' and makes this 'J S II'

Convert category executes 'John Smith II' and marks Smith as convert pre-word and 'II' to '2nd'.

Our conflict resolution specifies that the convert category is more specific than the reduce category, thus keeping the word 'Smith' and '2nd'. In addition, the pre-word convert flag in this particular example signals the conflict resolution manager to keep 'Smith', initialize 'J', and append '2nd '. This is possible because we have retained our original text and the converted text. The text did not change and an integrated rule has informed us that the word 'Smith' remained the same and by examining all words, we deduce that this is the last name.

Example Before/After:
Before - John Smith II
After - Smith J 2nd

[1] Forward Chaining is the logical construct in which the number of conclusions reached is small, but the number of ways to reach a particular conclusion is large.

[2] Journals often change formats over the years to accommodate new publishers or printers. Therefore the rules may need to change even though the Journal Title remains the same.

The conflict resolution strategy at the category level is that of specificity ordering. There is also a conflict resolution strategy within a given category: priority list rule ordering. Rules within a given category are assigned a priority level to avoid conflicts. An example of this is the following:

Glenn Ford, John Smith, and David Wells

We have the following Author Delimiter rules defined

',' and ', and'

However, the ',' is assigned priority 1, and the ', and' is assigned higher priority 2. If we did not give a higher priority to ', and' we could end up with 'and' as part of the author name or create a null value.

In our latest ground truth testing of the author reformat rules system we tested 1857 authors from OCR data. Of those 1857, 41 were reformatted incorrectly, for a 97.29% correction rate. Of those 41, all 41 were missing rules defined for a given case. An example of a missing rule is given in the case of an author field that reads:

Glenn M. Ford, Jr., John Smith.

By adding the rule [', Jr. ' Author Delimiter priority 2] to our test set, with just a new rule created and no changes in code required, we achieved 100% correct reformatting in the test set.

3 Reformatting the Affiliation field

The reformatting strategy for the affiliation field is quite different from the above. The OCR data for an affiliation field could contain many affiliations, since each author may have a different affiliation. This data is often difficult to reformat. One reason is that only the affiliation of the first author is to be retained, in line with MEDLINE conventions. Another reason is that the desired data is spread out over the entire field and not contiguous. For example, in a 30 word affiliation zone, we may only want to retain words 1-8, 12-14, and word 30. Our method is to do probability matching to historical data of ~130,000 unique affiliations collected to date.

The first step is to read all these unique affiliations into memory and create a Ternary Search Tree [1, 4] for each affiliation. We then create a soundex word list [2, 3] for each affiliation.

When a zone is identified at the labeling stage as an affiliation field, the OCR data is first processed through a partial-matching algorithm. Low confidence characters are replaced with wildcards.

> Example: Uniuersity. The 'u' is actually a 'v' but the OCR engine assigned it as a 'u' with a low confidence level. The partial match algorithm replaces the 'u' with a '.' signifying that this character is a wildcard, and that any word in our search tree that has the pattern Uni<any letter>ersity is considered to be a match.

The first step is to determine if a word in the affiliation zone matches one in the affiliation list. Ignoring implemented performance optimizations[3] we perform a partial word match for all the words in the OCR list and build up a chain of those words that do match. We also track distances between chains.

Consider the example of trying to find the affiliation "Department of Computer Science, University of Maryland" in the affiliation list. The OCR input string looks like: "Department of Computer Science, Department of Engineering, University of Maryland, Department of Computer Science, Johns Hopkins University."

Since only the first affiliation is to be retained, there is considerable data that is irrelevant. The problem is to retrieve just the data needed. By word chaining we can find chains of words that exist in both the OCR text and in an affiliation zone and then use these to derive weighted probabilities.

In this example there is a chain of 4 words that match, followed by 3 that do not match, followed by 3 more that match, and finally 7 that do not. Our probability algorithms compute chain word matches and distances between chained words.

[3] Optimizations such as: if the first word does not exist in the affiliation listing entry 1, go to entry 2 instead of looking at every OCR word.

The next step in our process reverses the partial word match. The ~130,000 affiliations are matched to the OCR affiliation.

Using the same example, "Department of Computer Science, University of Maryland" has 7 words and all 7 occur in our OCR word list. It is likely there is another affiliation entry that looks like "Department of Computer Science, University of Delaware". This would give a high match of 6/7 words. By comparing and weighting word matches from OCR to Corrected Affiliation and Corrected Affiliation to OCR, and using information such as the number of words matched, total number of words, chain of words matched, and chain of words unmatched, we arrive at a probability between 0 and 1. Note that partial matching is used to help cover OCR errors that would ruin a literal string pattern matching as the affiliation field is often in a smaller font and might incur higher than normal OCR error rates.

In addition to a partial match search algorithm, a soundex algorithm is used with the addition of OCR substitution. For the example in which 'Uniuersity" has the 'u' as low confidence, a substitution table developed lists common OCR errors where a u == v == y. All three letters are substituted in the low confidence 'u' position, and if a word matches with a soundex hash it counts as a match.

In our ground truth testing with affiliation zones, if the OCR affiliation exists in our affiliation list of 130,000 entries, the probability that the affiliation match is the correct one is 88%. The affiliation reformat module picks the top 5 candidates which are presented to the reconcile operator who can choose the correct one in the 5, or pick the nearest match and type in any missing data, usually a room number, zip code, or an email address.

4 Reformatting the Article Title field

The title field uses the same principles as in the author rules system, but requires very few rules or categories. Of the 8 categories mentioned in the author reformat section, only 3 are used: Uppercase, Lowercase and First Letter Upper.

5 Current Work

Current research focuses on the correct detection of superscripts in both the author and affiliation fields to help improve reformatting algorithms. With this information available, correct affiliation matching is expected to reach the middle 90 percent range.

6 Summary

This paper has described the field reformatting stage in the automated data entry process being designed at the National Library of Medicine. The rules and rule categories applicable to reformatting the author, title and affiliation fields have been given.

References

[1] Bentley JL, Sedgewick R. Fast algorithms for sorting and searching strings. Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms. Jan 1997.

[2] Baase S. Computer Algorithms, Addison-Wesley, 1988, pp 242-4.

[3] Hall PAV, Dowling GR. Approximate string matching. ACM Computing Surveys (1980).12:381-402.

[4] Bentley J., Sedgewick B. Tenary Search Trees. Dr. Dobb's Journal, April 1998, pp 20-25

Appendix A

Table 1: Categories of Author Reformat Rules

Category	Description	Example
Particle Name	Many names contain "particles" forming an integral part of the family name and possibly bearing significance to the family. A particle is retained as part of the reformatted author name.	*Etienne du Vivier* becomes *du Vivier E,* where 'du' is a particle and is retained as is and preceding the last name Vivier. The first name is initialized.
Compound	Compound family names are preserved in the form given and are often difficult to detect. We	*L.G. Huis in 't Veld* becomes *Huis in 't Veld LG*

	use a mix of rules to deduce it as a compound name. Most compound names use a hyphen. Those that don't can often use particle name rules to help preserve the compound name.	*H.G. Huigbregtse-Meyerink* becomes *HuigBregtse-Meyerink HG*
Convert	Convert is a broad category that deals with general requirements to convert one pattern of text with another.	James A. Smith IV becomes Smith JA 4th
Religious	Religious titles include Mother, Sister, Father, Brother. Names with surnames are handled differently from those that have no surnames.	Surname example: *Sister Mary Hilda Miley* becomes *Miley MH* No-Surname example: *Sister May Hilda* becomes *Mary Hilda Sister* For translated articles, e.g., from the French, *Soeur* becomes *Sister.*
Reduce	Reduction rules cover the elimination of text with a single author name. It also handles the Reduction of a person's given name and marking of the Surname if present.	*Mr. John Smith* becomes *Smith J* *John Smith MD* becomes *Smith J*
Lowercase	Some fields present all data uppercase. This rule simply converts to lower case all text that is uppercase.	JOHN SMITH becomes *Smith J*
First Letter Upper	Title and Author at times will require that the first letter of a specific word be uppercased, depending on other rules.	JOHN SMITH becomes *Smith J*
Author Delimiter	Many articles are by multiple authors who contributed to the paper, such as this one. This rule takes an OCR stream of text and creates a word list, a chain of words, and delimits where a particular author begins and ends in the complete chain of words.	Example1: *Glenn M Ford, John Smith* becomes: *Ford GM* *Smith J* (, is the delimiter here) Example 2: *Glenn M. Ford, John Smith, and Susan O'Malley* becomes: *Ford GM* *Smith J* *O'Malley S* (', and' is the trigger, which must precede in priority ',' as a triggered rule)

Manual Verification and Correction of Automatically Labeled Zones: User Interface Considerations

Glenn Pearson[1] **George R. Thoma**[2]
National Library of Medicine (NLM)
Bethesda, Maryland 20894

Abstract

A system for automatic extraction of bibliographic information from scanned document pages is being developed at NLM. A goal of this system is the automated labeling of rectangular image zones as title, authors, abstract, etc. Software to help achieve this goal has two broad roles. The research role contributes to the extraction of salient features from image or OCR data and their subsequent analysis by AI systems being tuned for zone label categorization. The production role embeds such functionality within the planned production workflow. Furthermore, since automated labeling will not be 100% reliable in the near term, an important part of the production role is manual verification and correction of zone segmentation and labeling to achieve full accuracy.

A software-development case study is presented of "Zone Checker", a software component that contributes to both roles. This duality of roles introduces multiple classes of users, with implications for menu structure and other aspects of graphical user interface (GUI) design. Additional design issues are explored, particularly those specific to visualizing, verifying, and correcting zones and their labels. An example is how to best convey the relative reading order among zones for image text that flows from one column to the next.

1 Introduction to MARS, the Medical Article Record System

1.1 Keeping MEDLINE Up to Date

One of the major accomplishments and services of the National Library of Medicine (NLM) is the creation and maintenance of MEDLINE, a database of journal article citations and abstracts covering much of the world's biomedical and related scientific literature. This resource, originally accessible by researchers, doctors and other providers, and medical librarians, has been available since 1998 to anyone with web access [3]. MEDLINE's data is also incorporated into a number of public and private value-added databases.

The traditional method of data entry into MEDLINE, as into most bibliographic databases worldwide requiring high accuracy, is by typing in all information in duplicate. More recently, two additional methods have been implemented that are less labor-intensive:

- Direct electronic submission from publishers; and
- Scanner-based journal imaging, with optical character recognition (OCR). This is the MARS approach.

The portion of journals handled by each of these two new methods continues to grow.

1.2 The Origins of MARS I

MARS I was developed, starting in late 1996, by NLM's Communications Engineering Branch (CEB) in collaboration with other groups within the library. Since installation, MARS I has seen substantial incremental improvements in both software and operations. Throughout 1998, a sustained throughput of over 600 journal articles per weekday was achieved, one third of MEDLINE's total data entry requirements.

1.3 The Origins of MARS II

In parallel with improvements to MARS I, CEB began research on and development of a successor system with three major goals:

- Diminution of manual work per article, by the incorporation of both document image understanding techniques and improved user interfaces;
- Replacement of the intricate MARS I file-based mechanisms with a database system, to provide better reliability, data integrity, and potential for throughput growth;
- Replacement of DOS and Windows 3.1 16-bit client applications with Windows 32-bit ones.

[1] To whom correspondence should be addressed: Glenn_Pearson@nlm.nih.gov. Dr. Pearson is a computer scientist with Management Systems Designers, Inc., a provider of on-site software development services.

[2] Chief, Communications Engineering Branch; thoma@nlm.nih.gov

[3] See NLM's homepage, www.nlm.nih.gov, for no-cost MEDLINE access via PubMed and Internet Grateful Med.

Associated with this was a migration from C to C++, along with selective incorporation of OCX componentware.

Since the start of 1997, MARS II has been a major CEB project. At the outset, the project's software developers, generally adept in C, were more heterogeneous with respect to experience with C++ and the DevStudio/ Microsoft Foundation Classes (MFC) environment.[4] Nevertheless, by fall of 1998, a first prototype of client stations working with the database was tested. First production deployment should occur this year.

2 Zone Checker As First Conceived

2.1 Beyond MARS I

In MARS I, the first page of each appropriate journal article was scanned. Next, within each bi-tonal (black and white) image on screen, a few fields were manually "zoned". For each, a rectangle was drawn on the bitmap image by positioning two opposite corners, and what we will call here a "zone label type" (specifically, either Title or Abstract) implicitly assigned by entry order. Only the areas within these zones were OCR'd. Separately, all fields except the abstract entered by typing. The title was entered both ways in order to allow the two sources to be matched and merged.

In MARS II, as part of the goal to minimize or eliminate manual steps, manual prezoning would no longer be done at the scan station. Instead, scanning would be followed by full-page image segmentation (to locate paragraphs) and extensive auto-labeling. In order for the latter step to become highly reliable, multiple sources of information (or analysis techniques or rules) would need to be consulted, some of which are general and others of which are journal-specific. The latter, in aggregate, represent a "journal profile".

2.2 The Research Role – Capturing Journal Profiles

Early in the MARS II research program, a need was identified for the development of a software tool to help capture certain components applicable to a journal profile. Such information, once captured, would be analyzed and generalized to form profiles. This tool would read TIFF images generated by the MARS I project. It would perform these steps:

1. Open a scanned image;
2. Algorithmically locate zones. This boundary-detection process could be informed by image segmentation, artificial intelligence (AI), historic journal-specific information about layout and font styles, and, with some limited OCR-like capability, and keyword recognition;
3. Display the zoned image, along with a table of zone label types;
4. Allow the operator to manually assign a label type (e.g., author, title, etc.) to each zone;
5. Store the resulting data (journal, image number, zone sequence number and position, label type) in tabular form as part of a journal profile.

Step 4's verification establishes the ground truth, an essential element when building journal profiles either manually or using AI techniques requiring feedback or learning (e.g., weight adjustments to an expert system or neural net). It also allows quantified assessment of the quality of automatic zone finding.

2.3 The Possible Production Role – Using Journal Profiles

Contemporaneously and independently, an early database design identified certain client modules in the MARS II workflow [Table 1]. This design didn't yet clearly incorporate automated labeling. To do so, auto-labeling must happen no earlier than "Segmentation" and no later than "Zone Validation".

Consider the three validation stages shown. The last two of these, OCR and record validation, are handled in MARS I by a single "Reconcile" module (but uploading is a separate process). The first stage, that of "Zone Validation", is new. With automated labeling available, that step was seen as the subsequent inspection and correction of both zone boundaries and labels. Its inclusion is necessitated by the belief that the automated segmentation and labeling systems will not be 100% reliable, due to imperfect OCR data, or the use of AI systems with initially sub-optimal performance. Potential sources of AI difficulty include undersized training sets, incomplete coverage across the large, stylistically-heterogeneous collection of journals, or inadequate zone-feature recognition and categorization. In any event, errors in auto-labeling, if not caught early, propagate downstream to cause more corrective work for the final record validation or reconciliation operator.

Comparing Zone Validation with the tool design in

[4] C++ was chosen over Java because of the easier skill migration, more mature development tools, and faster run-time performance. Also, Java's strength is in creating distributed remote systems, but MARS requires the operators to be in physical proximity in order to circulate journals, since relying upon bitmap page images alone is sometimes inadequate for character-level inspection. At least this is true for the 300 dpi images that give us the best OCR results; others [10] have also found 300 superior to 400 and 600 dpi with commercial OCR systems. As for choosing MFC, the only real Windows-centric alternative, Borland's OWL, was no longer seen as competitive for new projects not requiring Windows 3.x support.

Table 1. Early MARS II Workflow Design [1]. Manually-operated modules (non-daemons) are in **bold**. While details of the number, names, and order of modules have since evolved, and a number of processing aspects refined, the overall concept remains generally valid.

Station	Purpose
Scan	Enter a new journal into the MARS II system, then scan the first page of each relevant article within.
Segmentation	Locate paragraph zones within each scanned image.
OCR	On each image, perform per-zone optical character recognition (OCR). As with MARS I, the Prime Recognition 5-engine OCR system is employed, but writing to the database, not to ".pro" files.
Spell Check	Remove doubt about low-confidence OCR characters if they appear within words found in a special dictionary
Zone Validation	Associate a type, such as "Title", with each zone of interest.
OCR Validation	Typed correction of OCR errors. This could be per image and zone, as in MARS I, or per character across images using "carpet" correction.
Record Validation	Handle all remaining data entry or correction tasks. Then upload to the pre-MEDLINE machine (for further specialist indexing, then entry into MEDLINE)
Archive	Periodic off-loading of production database

the previous section, it was quickly seen that this proposed module shared a preponderance of goals and features, particularly since one type of "tabular form" was database tables. However, since the form of the database was still in development, saving data in tabular form to the file system was also needed.

The main conceptual change required by the production role of Zone Checker was to Step 2, which now must also allow the simpler alternative of reading in zone location (generated by Segmentation) and OCR data, instead of performing its own calculation.

2.4 Two Roles, One Tool

Combining the two roles seemed both parsimonious of development effort and functionally synergistic. Thus, from the outset, Zone Checker was shaped towards becoming a possible MARS II component. This duality of roles had drawbacks and advantages. One advantage was that CEB researchers, as users of this tool and more accessible than the production personnel, provided early and on-going informal usability testing, and a stream of suggestions for enhancements.

2.5 To a New Step 2

The research role's Step 2 called for incorporating border-finding and segmentation; the thought was to upgrade and integrate some existing libraries. When exploratory efforts during the first implementation period revealed severe technical and legal hurdles, this goal was dropped, substituting:

2a. Read in zone locations and corresponding OCR data (characters, attributes, and bounding boxes) from a source external to Zone Checker.

The initial source for all this information was " .pro" files generated by the Prime Recognition OCR Server, with "auto-zoning" enabled, so as to include the segmentation task.

From the point of view of the production role, this information would instead be found in the database, generated by the OCR software daemon. CEB's current version of the latter, "Prod", also uses the PR OCR Server with auto-zoning.

As mentioned earlier, the early database design did not indicate where auto-labeling would occur. For convergence, it was decided to add another optional step to Zone Checker:

2b. Algorithmically guess each zone's label type.

For MARS II production, doing the (2b) work within Zone Checker was seen as a provisional convenience, predicated on the assumption that it would take no more than a second or two per image, and thus wouldn't slow down an operator significantly. A quite recent direction is to relocate the auto-labeling to a new unattended process earlier in the workflow, which also would incorporate a new zone-boundary correction phase.

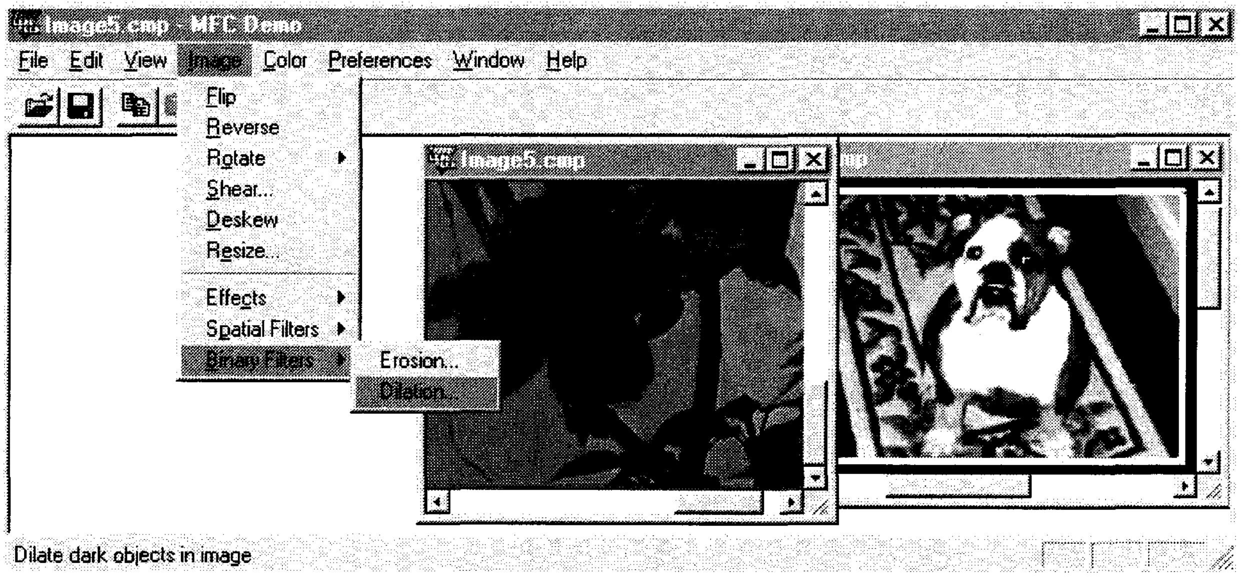


Figure 1. The MFC Demo Sample Program, MFCdem32. Shown are two open general-purpose images, and a few of the available image processing operations.

3 Early Design Choices and First Implementation

3.1 Selecting an Imaging Library and Starting Application Framework

Beginning with the foregoing concept of Zone Checker (basically a TIFF viewer and label editor with both file and database storage), a review of commercial Win32 imaging libraries was undertaken, looking particularly for components that support annotations atop scrollable images[5]. Lead Technologies' "Lead Tools Pro" [2] in dynamic-link-library form was selected, which advantageously had C++ wrappers (albeit thin ones) around its C API. In addition, a significant amount of sample source code was provided. In particular, "MFC Demo" [Figure 1], a standard DevStudio-wizard-generated code skeleton that had been fleshed out to encompass most of the Lead Tools API, became the starting point for Zone Checker. It was attractive because it immediately provided an extensive core of image file processing functionality. However, it lacked annotation features, which were hand-merged from another sample program [Figure 2].

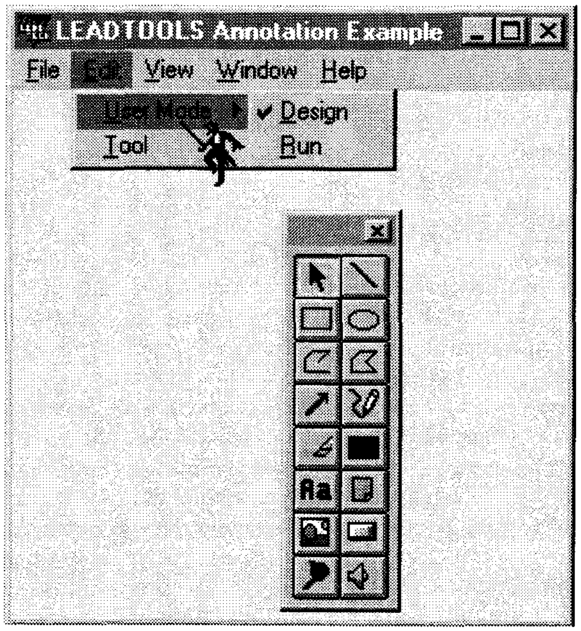


Figure 2. The Annotation Sample Program, Annot32. The floating toolbar shown for drawing annotations is discussed in a later section. Most of these annotation types are specializations of rectangles, such as hotspots, buttons, text boxes, highlights, and redactions (black-outs). The "run" and "design" modes became the internal basis for Zone Checker's "Adjust Labels" and "Select Zone" modes, respectively.

[5] In-house TIFF reader/writer code from another project was also considered as a starting point, but passed up because its conversion from 16- to 32-bit Windows was not complete in early 1997, nor was there annotation support. Note that Zone Checker development predated availability of Wang Imaging for NT, used in OCX form by some other MARS II modules.

3.2 Going with the Flow

With a compendium of third-party code such as the Lead Tools library, there are always some things that are made trivial or straightforward for the programmer, and other things that are difficult. Part of initial exploration is to identify these fairways and mine fields, and try to drive the design in such as way that the needs of the application can be satisfied by operations within the friendly landscape. This is as in contrast to a design philosophy, perhaps more appropriate for mass-market commercial software, in which the aesthetics and affordances[6] are prespecified in the absence of implementation considerations.

3.3 The Trade-offs of Featurism

Zone Checker's dual roles entailed supporting both an emerging database and existing file-based storage. As a "Swiss Army knife", this flexibility allowed it to be used as a go-between, for instance, to load historical file-based data into the database. The cost is that of software bloat [3], not just in terms of memory and hard drive footprint, but in terms of presenting users with a rich but complicated feature set, some of which are unused by certain user classes. An on-going effort involves eliding bloat wherever possible through feature re-packaging, hiding, disabling, or deletion.

For instance, the "Annotations" toolbar is very helpful for the developer in exploratory try outs of different on-screen representations. But, because marks drawn with it have no tie-in to OCR data, it is not helpful for most end-users. A new menu item was therefore introduced to toggle its presence[7]. This toggle is inaccessible to production users, for whom the toolbar is always hidden.

3.3.1 How Many Page Views?

"MFC Demo" provided a multiple-document user interface (MDI), so the first question was whether to reimplement it as a single document interface (SDI). It was left as MDI, mainly to minimize up-front development time and risk. Also, it was known that occasionally both the first and second pages of a journal article were scanned, because the abstract ran over to the second page. In such cases, the ability to see and touch both pages at the same time could be helpful. However, sticking with MDI caused later development time penalties, in working with the more complex doc-view internal structure and in making the seldom-needed multiple-document aspects less intrusive to the user. Most other MARS II modules that display images use form-based SDI.

3.3.2 The Triage of Imaging Features

The broad outline of MFC Demo's user interface was that provided by the DevStudio/MFC wizard, and initially retained by Zone Checker (until the later make-over of Section 5.1). But a few wizard-provided features, such as the most-recently-used file list, were removed because they seemed hard to extend to database interactions.

While MFC Demo ties "File/New" to a TWAIN-compliant local scanner, this capability was dropped from Zone Checker as unneeded and a potential support headache. Instead, the early Zone Checker user invoked a standard File Open dialog to choose an existing MARS I image, read-in via Lead Tools TIFF decompression. To enhance bitmap readability, the default display mode of scalable images was changed to scale-to-gray; the bitonal file image itself remained unchanged in normal usage.

Of the many image processing algorithms found in MFC Demo, those clearly irrelevant to Zone Checker (e.g., artistic effects and "slide show" multi-image transitions) were dropped. A few that seemed particularly germane (e.g., deskew, flip) were made more prominent in the top menu structure. Conversely, ones anticipated to be of infrequent use were buried deeper into the menu hierarchy under a general title of "Specials", an example of a bloat-hiding strategem.

MFC Demo had zoom menu items, but they weren't well matched to our needs. An early change (that matured through a number of improvements) was to introduce a separate "Zoom" bar.

3.3.3 Juggling Multiple Purposes with Setup Property Pages

Since Zone Checker had both research and production roles, it was obvious that configuration control was important. This first took the form of File/Open Setup and File/Close Setup menu choices. As its name suggests, Open Setup controlled what additional processing steps occur associated with opening each image. Initially, Open Setup provided a single dialog for locating the corresponding OCR file (e.g., 1.pro for 1.tif), the only source of zone-specific OCR information at the time. This file might be in the same directory as the image, or in a directory location given in the early database. The contents of the property page was soon expanded to hold database log-on and database vs. file-system-only choices. Then a second tabbed property page appeared for the feature discussed next. Close Setup saw similar incremental growth.

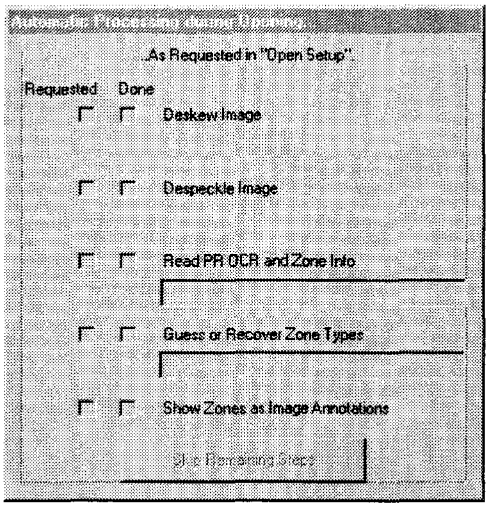
3.4 Communicating Transient Operations During Image Opening

As each image was opened, a configurable series of processing steps, of perhaps several seconds total duration, need to be applied to it. What should be displayed while this occurred? While a modal

[6] That is, GUI ways of offering control of functionality.

[7] This toolbar refuses to be hidden in a Windows API sense, so it's moved past the screen edge to hide it.

hourglass cursor, possibly-modeless progress bar, or other animation were considered, a more informative display was sought. A common alternative, text in the status bar, would mutate too quickly to be discernable. Instead, a custom dialog briefly appears [Figure 3].



Automatic Processing during Opening

...As Requested in "Open Setup"...

Requested Done

□ □ Deskew Image

□ □ Despeckle Image

□ □ Read PR OCR and Zone Info

□ □ Guess or Recover Zone Types

□ □ Show Zones as Image Annotations

Figure 3. Steps during Image Opening. The left column shows requested steps, the right those that are completed, with processing in top to bottom order. This shows the current version of this dialog; the original didn't have the extra text fields, which indicate, for instance, whether labeling is done by the built-in rules or by reading an external file. There is a setup property page of similar appearance, with a single column of checkboxes, in which steps may be requested or not. The despeckle and deskew steps are seldom needed; MARS II images will be already deskewed. It is the next-to-last labeling step that is most frequently toggled in research activities.

3.5 Enumerating Zone Types – Rich versus Minimal

MARS I reports a dozen-odd document field types to MEDLINE. On the other hand, one alternative method of document input to MEDLINE, electronic submission of structured documents directly from publishers to NLM, defines about fifty SGML tags. As part of developing Zone Checker, a rich zone descriptive system was defined, closely modeled on the latter but with some additional distinctions. This is flexible and allows AI training to categorize types that may be more important for exclusion than inclusion. Nevertheless, a minimal set is usually of most immediate interest.

3.6 How should a Zone Appear? The First Zonescape

Early on, the Lead Tools "rectangle annotation" was chosen as the representation for a zone. This is aligned with the image edges (although programmatic rotation is possible). Since the zone boundaries given by OCR were also so aligned, this was convenient from a structural standpoint. From the GUI point of view, the rectangle is much easier to manipulate than the main alternative, a polygon annotation. The latter is more flexible, allowing, for instance, picking out a particular few sentences from a paragraph, but positioning all the control points would be burdensome for the user, and tricky for the developer in relating to underlying OCR data. Instead, there could be multiple rectangles of the same type, linked implicitly by relative order or explicitly. Furthermore, a zone denoting a small snippet of text might be overlaid on a larger zone.

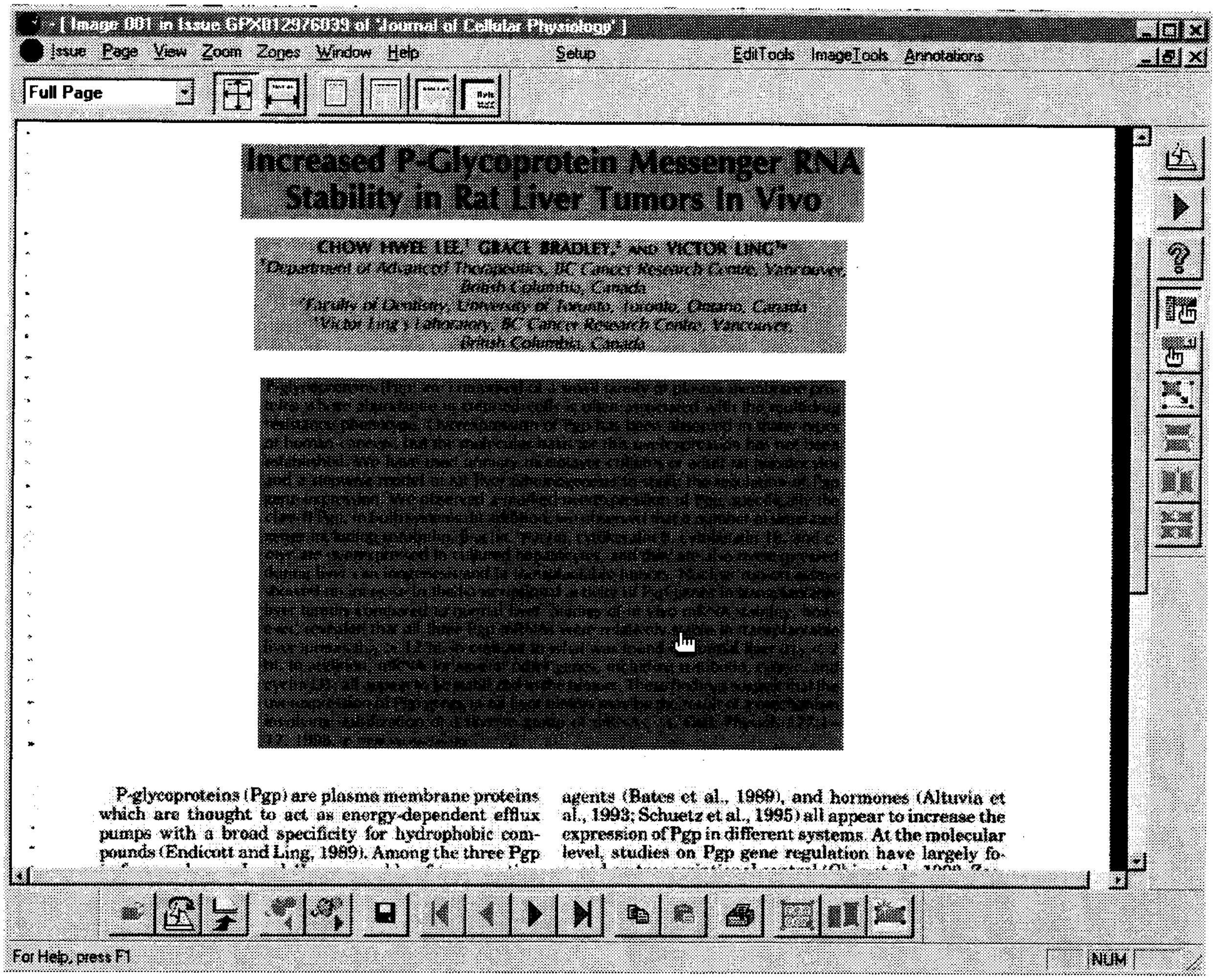
It was decided that each zone could be only of one type, and zone splitting or overlays employed as needed to work around any restrictions this might impose. Each zone type is either "major", represented by a particular solid color ("translucent" in Lead Tools parlance), or "minor", with "clear" interior and a colored border, most suitable for overlaying on major zones. An example is shown with major zones [Figure 4]. A fixed palette of colors relating to particular zone types (later called a " zonescape") was developed. Colors selected were spread out in a spectrum, with zone types varying from red to purple corresponding to where they most commonly appeared on a page. Most colors were fairly bright. For differentiation, colors for unknown zone type or body text were pastel.

Zones that were most likely garbage were specially rendered to be unobstrusive. For instance, type "White-out Blem (nontext)" means that the zoned portion of the original page has no actual text, and typically no actual graphical content either. If the OCR did report any text, it is a misrecognition; for instance, it is not uncommon for zones around border or gutter shadows to have phantom characters like " i" or "I" in them. Zone Checker visualizes this distinction with a subtle hint: With no OCR text, the zone is shown in light gray crosshatches on opaque white; otherwise, the crosshatches are golden.

3.7 Implementing Zones

A great deal of early effort was being able to read in the ".pro" data, associate it with internal data objects, and represent it on screen using the rectangle annotation.

Atop a visible "gell" zone is a totally transparent "hot spot" zone of the same size. A click on the hot spot routes an event (including a hot spot ID) to a Lead Tools-specific callback function within application code. In our case, the hot spot ID allows discovery of the corresponding "Zone Object", the overall manager of a specific zone's annotation, OCR data, and other information. This information is used to position the zone-type selection popup menu, and is revised if the user changes the zone's type, which causes an immediate change to the zone's color.


- [Image 001 in Issue GPX012976039 of 'Journal of Cellular Physiology']
Issue Page View Zoom Zones Window Help
Setup
EditTools ImageTools Annotations
Full Page
For Help, press F1
NUM

Increased P-Glycoprotein Messenger RNA Stability in Rat Liver Tumors In Vivo

CHOW HWEE LEE,[†] GRACE BRADLEY,[‡] AND VICTOR LING[*]
[*]Department of Advanced Therapeutics, BC Cancer Research Centre, Vancouver, British Columbia, Canada
[†]Faculty of Dentistry, University of Toronto, Toronto, Ontario, Canada
[‡]Victor Ling's Laboratory, BC Cancer Research Centre, Vancouver, British Columbia, Canada

[illegible]

P-glycoproteins (Pgp) are plasma membrane proteins which are thought to act as energy-dependent efflux pumps with a broad specificity for hydrophobic compounds (Endicott and Ling, 1989). Among the three Pgp

agents (Bates et al., 1989), and hormones (Altuvia et al., 1993; Schuetz et al., 1995) all appear to increase the expression of Pgp in different systems. At the molecular level, studies on Pgp gene regulation have largely fo-

Figure 4. The Current Appearance of Zone Checker. Zone coloring is discussed in the main text. Note in particular the small "White-out Blem" zones near the left edge, with their faint crosshatches. The frame shows three toolbars, "Zoom" at top, "Main" at right, and "Specials" at bottom. The latter is usually hidden. The buttons in Main are, from top, Open Issue, Next Page, Help, the three mutually-exclusive modes ("Adjust Label", "Set Reading Order", and "Select Zones"), and additional buttons that become enabled in select-zones mode: Split Horizontally, Split Vertically, and (not yet implemented) Merge.

4 Under the Hood

As with many software projects, some time-consuming aspects of Zone Checker development had a relatively modest impact on the user interface.

4.1 Automatic Guessing of Zone Labels.

Zone Checker served as a research testbed for "first generation" rule-based algorithms for automated labeling of zones [Appendix]. If enabled, these built-in C++ rules would activate as each image was opened and the corresponding OCR data automatically read in.

4.2 Saving Zone Features and Labels

It became valuable to be able to save the zone labels, as well as certain of the calculated features, to files, respectively called ".lab" and ".zon" files. These could then be used to train external AI labeling systems, such as a neural net [4]. Later, it became possible for Zone Checker to read .lab files as well; and analogous read/write actions using the database as the persistent label store were added. The writing of this information occurred as each image was closed, which necessitated adding more property pages to Close Setup.

4.3 Interfacing to the Prototype Database

To get images and store results, Zone Checker connects automatically on start-up to a MARS II-specific database, unless configured to use only the file system. The initial test database for Zone Checker was a local MS Access one, accessed via MFC's ODBC class wrappers. But within six months, Zone Checker had migrated to a small SQL Server 6.5 database elsewhere on the LAN, populated with MARS I images and information, suitable for testing and initial profile development. ODBC was phased out in favor of RogueWave's DBTools.h++ [5] with its associated SQL Server driver. Subsequently, the design and content of the database continued to evolve as many complicated design issues were resolved. Rogue Wave/C++ wrapper classes for most database tables were coded and incorporated into a shared library used by Zone Checker and other modules. Windows NT became the platform target for both database and clients.

5 Recent GUI Improvements

A number of usability enhancements have occurred within the last six months.

5.1 Redesign of Main Menus and Workflow

A GUI redo [Figure 5] streamlined and refocused the application towards production. The items within the main "File" menu were parceled out into separate new "Issue", "Page", and "Setup" topics. The latter merged File's setup options [Section 3.3.3.] with those (seldom appropriate to change, such as scale-to-gray) of "Preferences", and sequestered them from production operators. To reflect their non-production status, "Edit" and "Image" were renamed " EditTools" and "ImageTools" and moved to the right side of the window frame; EditTools got the various print-related functions from "File". The new "Zones" menu (used as an example in Figure 6) absorbed general-use operations from "Annotations", leaving behind researcher-only functions. The main toolbar was split into "Main" and "Specials" (as shown in the earlier figure), and a show/hide feature for the latter added to "View". Subsequently, View became the repository for general-user options, such as mini-icons [6] on the menus [Figure 6] and toolbar button size.

An important workflow improvement was to move away from generic {File/Open, File/Close} processing to an {Open Issue, Go to Next Page} paradigm for both file-system-only and database configurations. But the original random-image-access method is still available for special research purposes, now as Page/Open.

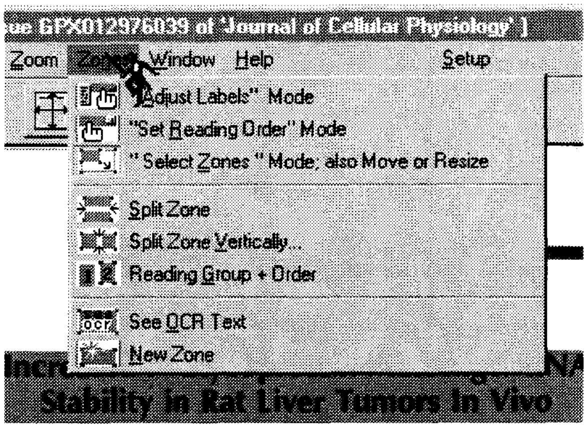


Figure 6. Mini-Icons on Menus. Every menu item that has a corresponding toolbar button is decorated with a small version of the same image. The user exercises View/Options to turn this feature on or off.

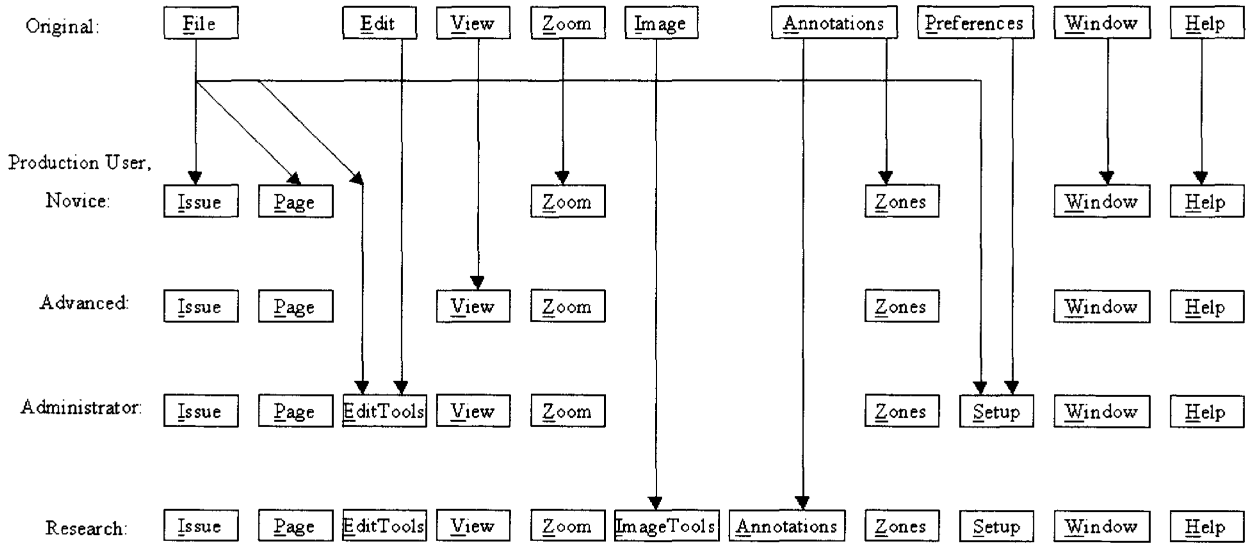


Figure 5. Redesigning the Top Menu Structure and its Accessibility by Class of User. The top row has the original left-to-right menu order. The "novice" versus "advanced" production distinction is so far only theoretical. A domain group was established for research users, and the current user's membership within it checked at Zone Checker startup. Appropriate features are disabled for non-researchers. In addition, the Setup menu solicits a special administrative password. This approach was taken so that a production user with a setup problem could ask the local manager to intervene without requiring the user to log off. Note that operations with the MARS II database require user pre-registration.

5.2 Ways of Visualizing Zone Types

For a long time, the single fixed zonescape (label palette) discussed earlier was the only feedback to the user as to zone type. The total palette mapping could be viewed within on-line help or as a color print.

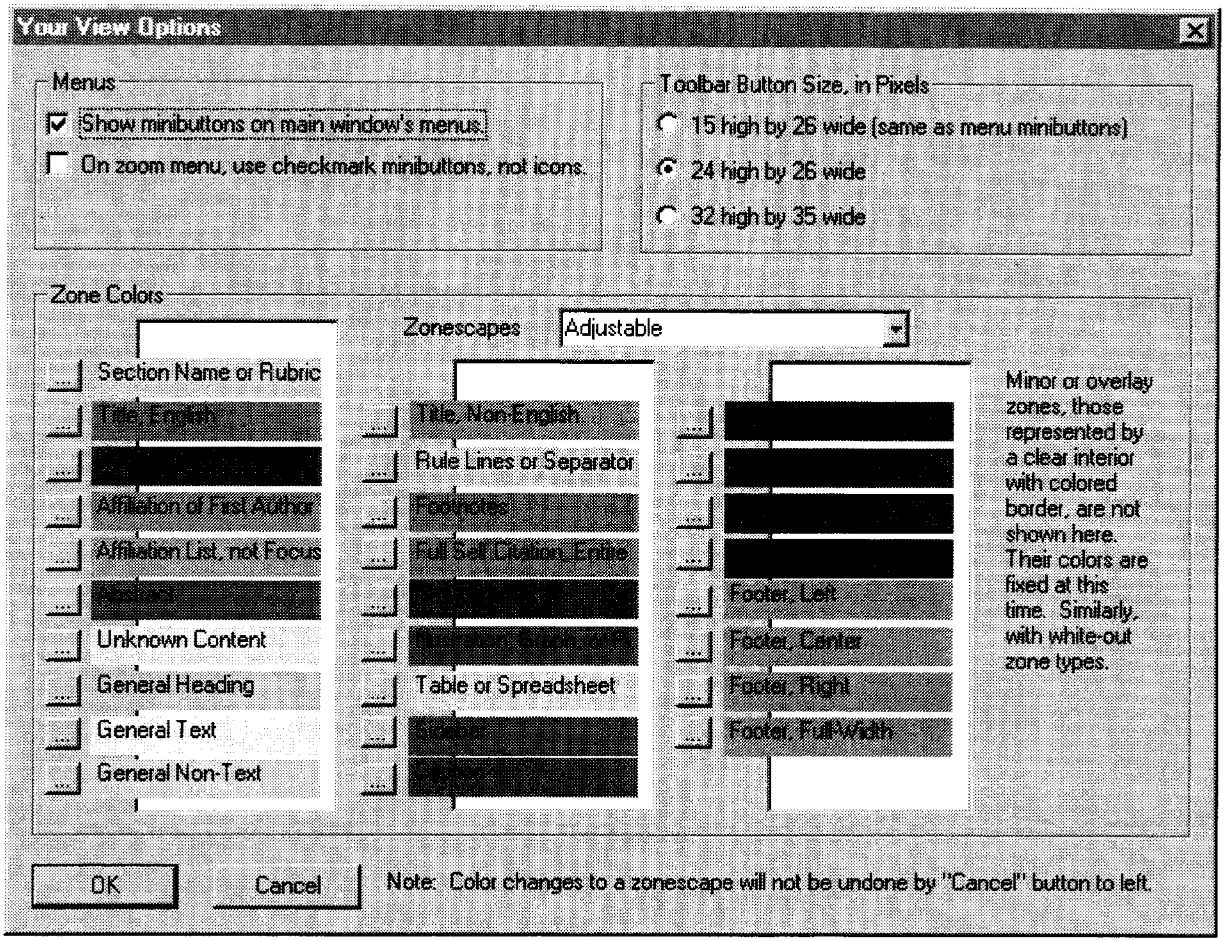
5.2.1 A User Designed Zonescape

Users may differ in their color acuity and preferences. To accommodate this, as well as facilitate usability testing to optimize the default zonescape's colors, a second, user-definable zonescape was invented. Management of these two zonescapes occurs in the new View/Options dialog [Figure 7]. Further zonescapes may be added in the future.

5.2.2 Zone Types Shown as Text

When in "adjust label" mode, it was felt that textual feedback in addition to colors would be helpful. Figure 8 shows one very early idea. Another was placing a check next to the zone-side popup menu item (but this is less useful with a multilevel menu). A third possibility was having a toolbar palette of color swatches and zone labels, analogous to our zonescape-setting property page, but always visible.

The direction pursued was instead to have "fly-over" help when moving the cursor over a zone. While the text, e.g., "Abstract", could have been put into the status bar, it was decided that a tool tip rectangle immediately below the cursor would lessen eye transversals. Since this appears and disappears automatically, and is repositionable by moving the cursor, it doesn't get in the way of the underlying bitmap text.



Your View Options

Menus

☑ Show minibuttons on main window's menus.

□ On zoom menu, use checkmark minibuttons, not icons.

Toolbar Button Size, in Pixels

○ 15 high by 26 wide (same as menu minibuttons)

● 24 high by 26 wide

○ 32 high by 35 wide

Zone Colors

Zonescapes: Adjustable

Section Name or Rubric

Title, English

Affiliation of First Author

Affiliation List, not Focus

Abstract

Unknown Content

General Heading

General Text

General Non-Text

Title, Non-English

Rule Lines or Separator

Footnotes

Full Self Citation, Entire

Illustration, Graph, or Pi

Table or Spreadsheet

Sidebar

Caption

Footer, Left

Footer, Center

Footer, Right

Footer, Full-Width

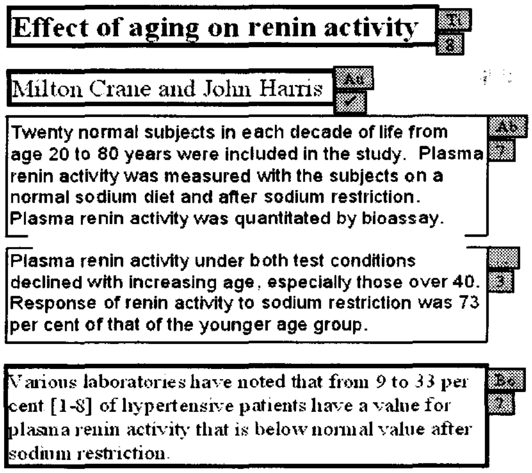
Minor or overlay zones, those represented by a clear interior with colored border, are not shown here. Their colors are fixed at this time. Similarly, with white-out zone types.

OK

Cancel

Note: Color changes to a zonescape will not be undone by "Cancel" button to left.

Figure 7. Zonescape Management. Only the set of 27 "major" zones (those with solid colors) are shown as swatches that can be manipulated at this time. Pressing a button to the left of a zone brings up the standard Windows "color chooser". A new color can be selected therein via several ways, unless the zonescape is the read-only default one. The color chooser has a set of "custom color" boxes, which here is used as a most-recently-visited list. This makes it easy to copy colors between zone types or between zonescapes.



Effect of aging on renin activity

Milton Crane and John Harris

Twenty normal subjects in each decade of life from age 20 to 80 years were included in the study. Plasma renin activity was measured with the subjects on a normal sodium diet and after sodium restriction. Plasma renin activity was quantitated by bioassay.

Plasma renin activity under both test conditions declined with increasing age, especially those over 40. Response of renin activity to sodium restriction was 73 per cent of that of the younger age group.

Various laboratories have noted that from 9 to 33 per cent [1-8] of hypertensive patients have a value for plasma renin activity that is below normal value after sodium restriction.

Figure 8. The Early "Ears" Concept. In this mockup, each zone was decorated with a two-button "ear". One button holds a zone type abbreviation: "Ti" for title, "Au" for author, "Af" for affiliation, "Bo" for body text; etc. A suffix digit could also appear, for instance, "Au1", "Au2" if there were two author zones separated by a non-author one. "..." was used to indicate, "this zone is a continuation of the previous zone's label." Continuations were further indicated by leaving the abutting top and bottom borders open, except at the corners. (This style is not part of the available Lead Tools repertoire; perhaps it could be drawn by overlaying an opaque white line on a rectangle's border.) The other ear button showed a labeling confidence level, in the range 0..9. (Labeling confidence values are not yet available.) An ear is also a hotspot, so that the operator can override the guessed value. In such a case, the confidence digit is replaced by a checkmark. Current Zone Checker handles labels and reading order as more distinct modalities than the ears concept. But the implemented reading order display, discussed next, is stylistically similar.

5.3 Reading Order

It's often necessary to determine the "reading order" among certain zones, particularly those of the same label type whose OCR text is to be combined downstream. This is particular true for same-type zones that are not adjacent, such as those containing text flowing from one column to the next, and thus not candidates for merging. The normal OCR system does not deduce full-page reading order. There is an optional mode to use Xerox's Textbridge engine to attempt such a sort, but this is reportedly problematic. In general, even with content understanding, it is difficult for humans to always agree on the proper reading order encompassing all text elements on a page.

Zone Checker would instead support manual partial ordering, by letting the user define one or more zone groups, and set the order within each. Zone Checker would automatically provide a default full-page group. Within this group, ordering is top-to-bottom by top zone edge, then left-to-right order by left edge. This ordering calculation is performed when zones are first read in, and on subsequent pertinent events such as zone splits. Placing a zone in a user-specified group in effect hides the default group ordering.

5.3.1 Initial Interaction Approach

The first implementation was as a new function within the existing "select zones" mode. The user selects a zone, then hits a "reading order" button on the main toolbar. This brought forth a dialog box, with two fields to view and set a "group number" (greater than "1" for user-defined) and order within group. As this proved extremely tedious, a third mode, "reading order", was conceived. Here, every zone would have its own widget. This would display on its face the value of its current reading order, and could be clicked upon to alter it. Displaying reading order as text seemed more straightforward to present to users than alternatives such as drawing arrows between zones.

5.3.2 Widget Choices – On the Button

As widget candidates, Lead Tools provides a number of annotation types, such as stamp, note, or the familiar rectangle. But most compelling seemed the button annotation with the usual Windows beveled-button look. The opaqueness of the button is helpful. Another advantage is that it intrinsically responds to click events, unlike rectangles for which one must maintain a hotspot separately. (Internally, button events would be associated with a particular zone rectangle by using a programmer-defined numeric "tag", just like the existing zone hot-spot/gell implementation.). A button's text and the text color can be changed, but the font and its size are that of the current system font. The background color is the current Windows frame color.

Like all annotations, the button object is inserted into the "annotation container" that holds the zone rectangles. Thus, zooming the image, which re-scales all container objects, resizes the button itself. But, as noted, not the text. Thus, the button size must be chosen to be large enough so that centered text won't be clipped at the edges when an image is zoomed out and the button appears tiny. The size specified is currently independent of text. (Further improvement to button sizing, possibly with font metrics lookup, is possible.)

The default zone-relative placement of buttons should obscure as little "important" bitmap text as possible. It might be inside or immediately outside the border. The worst place inside is probably the upper-left corner. The right edge appeals over the left, since text is usually left-justified, but often not right-justified. Arbitrarily, the buttons are placed in the upper right corner; lower right or the centroid might work, too.

The use of numerals for both reading group and order within group was problematic. After experimenting with a "[2] 3" notation, capital letters were substituted, leading to a "A3" notation for group A, order number 3. "A" is the first user-selected reading group (internal number 2). A missing letter denoted the default group, e.g. “3”. The dialog box was altered to reflect this style (Figure 9). On the buttons themselves, the default group text is in black, the user-set groups in red. (It might be interesting to color code each user-set group differently, to see if that is helpful or confusing.)

5.3.3 Expedited Interaction

Within the “reading order” mode, it would be better to just touch each zone in turn to set the order, and let the system assign the order numbers. Rather than add this to the button functionality, it is touching the non-button part of the zone that does this. Then button presses are merely for corrections, a slower but infrequent process.

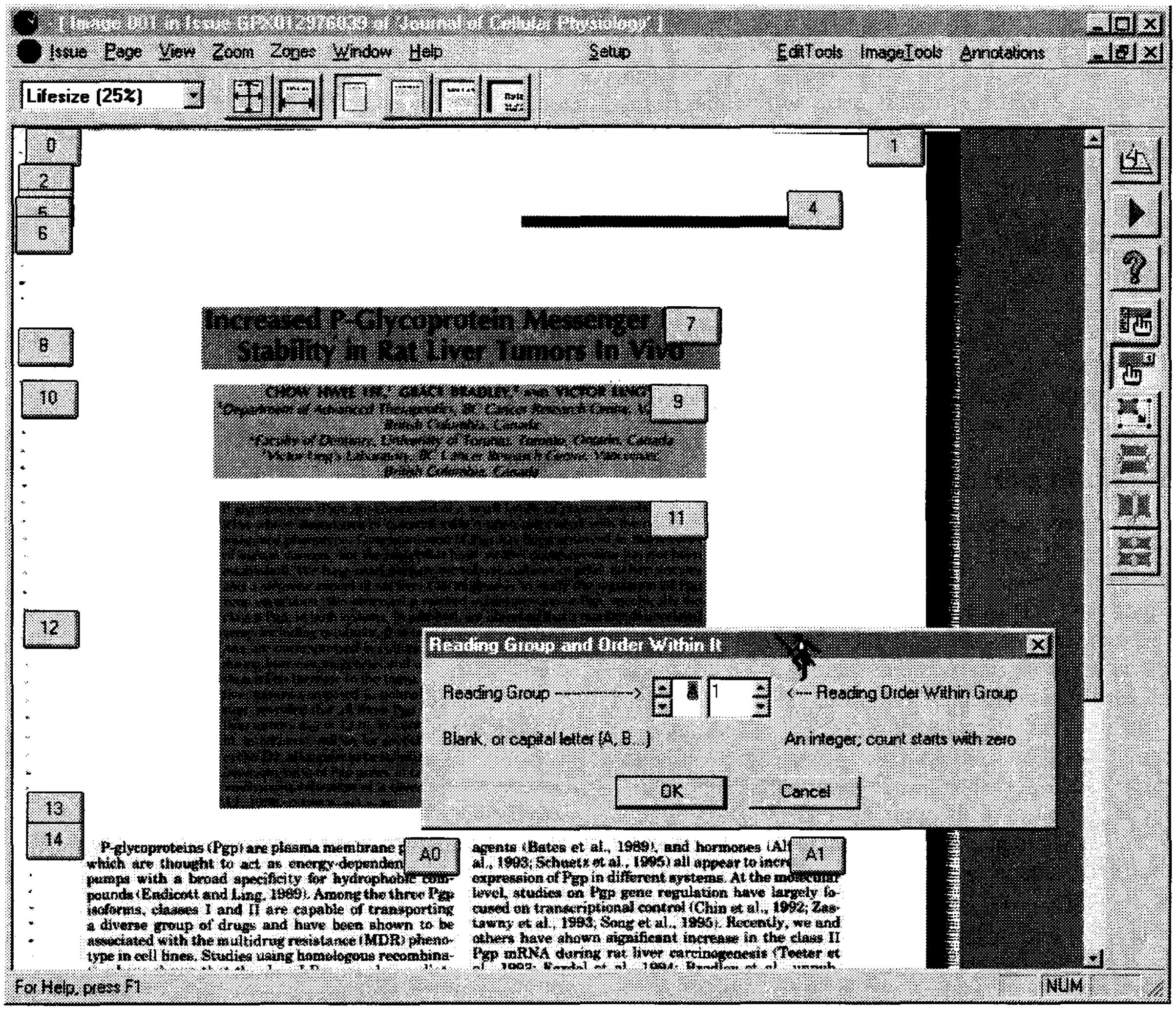
This is a little complicated, given our two-part reading-group/reading-order bifurcation. It was decided to program the interaction this way: the first time one enters reading-order mode for a given image, zone touches cause sequential numbering within group “A”. One leaves that reading group by going to one of the other modes besides reading-order. Re-entering reading-order mode moves to group “B”. This approach, while a little awkward, avoids introducing more controls, and seems reasonable when in the projected usage, few images will have more than one user-defined reading group.

If a zone is tiny, the fixed-size button placed strictly in the upper right corner totally obscures it, denying access to any non-button zone area to click on for order setting. A related problem: a small zone at the bottom or left edge has its button clipped at the image bounds. For these cases, button placement is shifted sideways, and/or aligned with a different zone corner. Nevertheless, each button is positioned independently, so buttons of adjacent zones can overlap. This may not be a problem in practice, since tiny or near-border zones seldom require user concern about reading order.

6 Further GUI Developments in Progress

6.1 Correction of Zone Types

In adjust-labels mode, clicking on a particular zone


[Image 001 in Issue GPX012976039 of 'Journal of Cellular Physiology']
Issue Page View Zoom Zones Window Help
Setup
EditTools ImageTools Annotations
Lifesize (25%)
0
1
2
5
6
4
7
8
9
10
11
12
13
14
A0
A1
Increased P-Glycoprotein Messenger Stability in Rat Liver Tumors In Vivo
For Help, press F1
NUM

Reading Group and Order Within It

Reading Group ------------> [ ] [1] <--- Reading Order Within Group

Blank, or capital letter (A, B...)

An integer; count starts with zero

OK Cancel

Figure 9. Displaying and Setting Reading Order. In this example, the two paragraphs near page bottom were assigned their own reading group “A” using the “expedited interaction” method of clicking the body of each in turn. If the user then had second thoughts about the right-most paragraph, clicking on its button brings up the dialog box shown, for correction. Selecting “blank” in the reading group field makes the order field read-only and automatically fills in the default ordering value (evidently 15 or 16 in this case).

brings up, immediately to its left, a popup menu. This predominately 2-level menu offers the entire set of possible zone types. Again, we face the issue of a rich versus sparse representation. With a large set of available zones, finding the ones in this menu of most importance because more problematic. In the long run, making this menu's contents dynamic, thus settable at runtime, instead of compile time, would allow both flexibility and concision. A tree control has been prototyped that reproduces the rich menu hierarchy and would allow a user to mark each zone type therein as skipped (hidden), optional, or required.

6.2 Zone Splits and Other Manipulations

From a GUI-design perspective, the area of greatest difficulty may be in splitting zones horizontally or vertically, or otherwise extracting portions of a zone. Splitting cannot happen at any arbitrary pixel, but only where the underlying OCR data permits. Furthermore, for vertical splits, it appears that some awareness of the underlying OCR data needs to be brought to the user's attention in a dynamic way. Zone "direct manipulation" [7] for splitting may be better done via mediating devices, such as sliders, that can be easily paired with OCR data display. Mediated horizontal and vertical splitters are being prototyped.

In Zone Checker, it is possible to create a new zone from thin air, and to alter the shape (or delete) existing ones. However, the degree of integration between PRO-specified zones and resized/new zones is rather weak. Specifically, changing a zone's shape has no effect on the extent of its OCR data, nor do new zones, used as overlays, inherit OCR data from their parent zone. This is an area for future consideration. Of more immediately need is the ability to merge adjacent zones. This will be non-trivial when they are side-by-side and individual OCR text lines must be aligned and appended.

7 Conclusions

Zone annotations are convenient "handles" for manipulating the underlying OCR data, and several ways of doing so have been presented. In this matter, as in many software design issues, much of the effort goes into finding the right balance between contending goals. We have retraced one exploration, that sought the balance between production and research needs, between a focused versus extensive feature set, and between engaging design concepts and implementation pragmatics. The tool thus created has multiple prongs and enjoys multiple purposes in furthering the next version of MARS.

References

[1] Ford, Glenn, *MARS Technical Specification 0.0B*, CEB internal document, 12/2/1996

[2] Lead Technologies, *LeadTools API Manual for Pro Express*, v. 7.0, Charlotte, NC, 1997.

[3] Kaufman, Leah, and Brad Weed, "Too Much of a Good Thing? Identifying and Resolving Bloat in the User Interface", *SIGCHI_Bulletin*, **32** (4), ACM, Oct.,1998, pp. 46-47.

[4] Le, Daniel, Jongwoo Kim, Glenn Pearson, George Thoma, "Automated Labeling of Zones from Scanned Documents", *SDIUT'99*, April, 1999.

[5] Rogue Wave Software, *DB Tools.h++ User's Guide and Tutorial*, Corvallis, OR, 1998.

[6] DiLascia, Paul, "New Interface Look: Cool Menu Buttons", *Microsoft Systems Journal*, Jan., 1998.

[7] Shneiderman, Ben, *Designing the User Interface*, Addison-Wesley, 1987.

[8] Hauser, S., personal communications, CEB, 1998.

[9] NLM, *List of Serials Indexed for Online Users* , ISSN 0736-7139, 1997.

[10] Kanungo, T., G. Marton, O. Bulbul, *Paired Model Eval. of OCR Algor.*, LAMP-TR-030, Inst. Adv. Comp. Stud., U.Maryland, College Park, Dec. 1998

Appendix. Built-in Auto-Labeling

Experience with these mostly-journal-independent "first generation" rules (Table 2) indicate that they were quite good at detecting white-out zones, albeit with some mislabeling of in-border page numbers as such. They were moderately good at the zones of most MEDLINE importance, depending on the degree to which the journal style is a common one. For instance, in a brief test with 10 autozoned page images from a single "easy format" journal issue, all title zones were correctly auto-labeled, as were 9 of 10 author zones. However, 3 of 10 affiliations were tagged as authors. The main abstract zones were correctly tagged in 10 of 11 cases, but 3 abstracts had a short last sentence, separately zoned, that was missed. The two mentioned problems might be ameliorated by more extensive interzone comparisons, and by further enlargement and refinement of the word lists beyond those in Table 2. The separately-developed "second generation" system [4], which, for example, has a word list for Affiliations based on historic data that is an order of magnitude larger than Zone Checker's, shows the effectiveness of these strategies. Other aspects that were nascent here and full-bodied in [4] include a numeric confidence value associated with label discovery and a multi-phase convergence upon the set of labels for a page. But perhaps most trenchantly, the advantages of aligning algorithms more closely with specific journal styles is manifest.

Table 2 (a-d). Built-in Zone Feature Recognition and Labeling Rules.

When an image's OCR and zone-location data is read into Zone Checker, the default label given to all zones is "Unknown Content". An quick software screening sees if this consists only of MARS I's manually-zoned title and abstract. Otherwise, auto-labeling analysis begins. Each zone is handled independently until the very end. For each, a group of quickly-calculated features are found in unnormalized and normalized forms. (The rightmost column of Table 1 in [4] enumerates those normalized features selected for export to a neural net system; others calculated include zone order, number of words, and number of initials, e.g., " A."). Next, a series of if-then-else tests proceeds (as given by row order in the subtables below) until a label is assigned. The most important labels for MEDLINE are shown **in bold**. A final limited revision step uses zone interdependencies: if no abstract was found above, one of the unknown or general-text zones in a particular part of the page may be relabeled as abstract.

a. Zones with Few or No Valid Characters. A long, thin zone without characters may be a rule line or similar separator. Otherwise, it's probably a "white-out blem", typically a black gutter or page edge artifact. "White-out Text" is most often text fragments on the facing page across the gutter. A page number is another possibility.

Constraints on OCR Text in Zone	Constraints on Zone Location	Assigned Zone Label Type
No characters	Overlaps central 2/3rds of image; > 1" long and < 1" wide (either orientation)	Rule Line/Separator
	otherwise	White-out Blem (non-text)
Entirely low-confidence characters (< 7 out of 9)	Fully within 1" of any image edge	White-out Blem (non-text)
Single numeral	Ditto	Page Number
Other single character	Ditto	White-out Blem (non-text)
> 1 characters	Fully within 1" of the left or right edge	White-out Text
	Fully within 1" of the top edge or 2" of bottom	Page Number

b. General Cue Word Matches. The matching here and in (c.) is intentionally case insensitive. A "delimited" cue word or phrase is on a line by itself, or followed by a space, semicolon, period, colon, or dash. A few dozen common biomedical headings are recognized, e.g., "chemicals", "enzyme assays", "experimental techniques", "growth conditions", "materials", "media", "methods", "mice", "production of", "reagents", "strains".

OCR Text in Zone	Constraints on Text	Assigned Zone Label Type...	
		...if one line of text	...if multiple lines
"received" or "first received"	Begins zone; Can skip any one prefix character, like "("	Date Received Or Accepted	
"revised" or "accepted"	Contained in zone		
"abstract"	Begins zone; Delimited	Abstract Heading	**Abstract**
"keywords" or "key words"	Begins zone; Delimited	Keyword List	
"introduction"	Begins zone; Delimited	Introduction Heading	General Text
Certain biomedical headings	Begins zone; Delimited	General Heading	General Text

c. Assignments for Central Zones. These final assignments are for zones that width-wise overlap the middle 2/3 of the image. If a label is still not assigned at the end of this process, it is left as unknown.

OCR Text in Zone	Constraints on Text	Additional Constraints on Zone Location	Assigned Zone Label Type
"case report", "case reports", "comentary", "editorial", "opinion", or "notes"	Begins zone; Delimited	In top 25% (2.75") of image	Section Name (if single line of text), or Title
--	Average point size of top-confidence characters > 14	In top 40% (4.4") of image	**Title**

c. continued next page

OCR Text in Zone	Constraints on Text	Additional Constraints	Assigned Zone

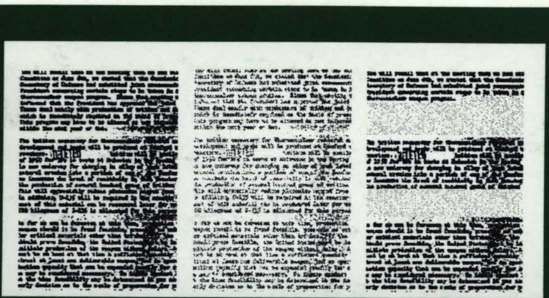
			on Zone Location	**Label Type**
Year, month (or its abbrev.), "journal", "j.", "volume", "vol.", "number", or "no."	Contained in first line; Delimited		In top 1.5 inches*	Header*
See Figure Xd	See Figure Xd		Top edge below 1"; bottom above 50% (5.5")	**Affiliation**
"to whom correspondence" or "author to whom correspondence" or "corresponding author"	Begins zone; up to 3 prefix characters can be skipped; Delimited	> 3 lines	Top edge below 6"	Correspondence To
				Affiliation
Initials (e.g., " X.") or commas (or "and" counted as if a comma)	On average > 1 initials per 20 characters or > 1 commas per 20. Not more than 1 occurrence of "and"		Fully in top 40%	**Authors**
Highest confidence characters	Average font size > 8 and < 13			General Text

* further subcategorized by size and position as Left, Right, Center, or Full Header

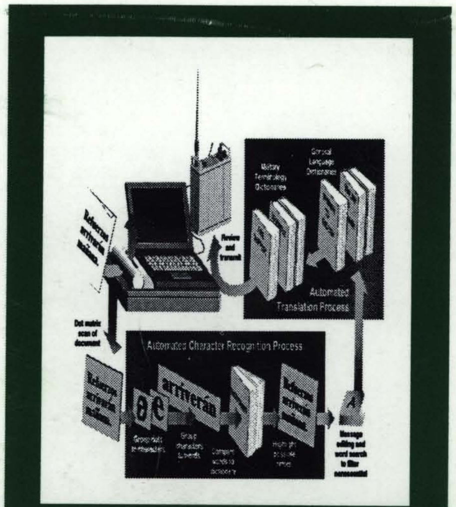
d. Cue Words for Author Affiliation. Affiliation zones are recognized by a coarse location screening followed by cue word matching, with the heuristic that there must be at least two cue words found on average per OCR line. A match constraint is that the first word of the OCR text must be capitalized (except suffix matches). String routines were built that match in the face of OCR errors due to common character misrecognitions [8]. Using reference books, a specialized cue word list was created, made up of the following groups (with varying degrees of coverage of non-English words), each with at least the number of items shown in the right column. NLM-indexed biomedical journal titles were used as a source for biomedical fields of study [9].

Category	**Representative Subcategories, and Examples in Quotes**	**Items**
General nouns for geopolitical entities	"Commonwealth", "State", "Republic", "Ville", "Town"	40
Specific geopolitical descriptors	Country names, with variant spellings	300
	States or provinces of US (including 2-letter abbreviations), Canada, Mexico, and China.	300
	Major cities (chiefly world capitals)	
	Common town suffixes (mainly US). …"ton", "mouth", "stad"	50
General nouns for geographic features. These often appear as part of a multi-word town or organizational name	Bodies of water and shorelines, e.g., "Rive", "Porto", "Springs"	180
	Terrain ("Mt.", "Valle", "Serra")	100
	Human constructions ("Depot", "Church", "Mill")	40
General geographic adjectives	Compass directions ("Western", "Ost", "Southeastern")	40
	Relative position or size ("Upper", "Outer", "Mid", "Greater", "Petit")	40
Specific national, regional, or body-of-water adjectives and nouns	"British", "European", "Caspian" "Scandinavia", "Atlantic"	60
Honorific titles. Here used as part of a hospital or institutional name.	"King", "Colonel", "Saint"	60
Common religious nouns, including proper names	"Order", "Mercy", "Maria", possessives like "Paul's"	60
Other Adjectives applicable to countries, towns, or hospitals, such as denominational terms	"Novo", "Royal", "Adventist", "United"	50
Organization descriptors	"Dept.","College", "Hospital", "Center", "Universidade"	40
Biomedical fields of study and diseases	Nouns, e.g., "Surgery", "Pediatrics", "Genetics"	70
	Suffixes like "…ology" in several languages	5
	Adjectives, e.g., "Health", "Prenatal", "Cellular", "Microbiol"	80

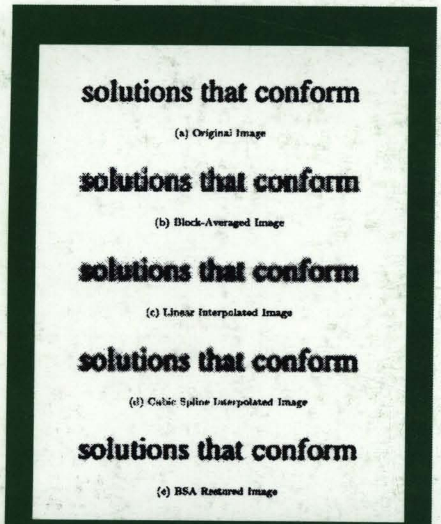
Author Index



Duplicate Detection


Military Terminology Dictionaries
General Language Dictionaries
Review and transmit
Automated Translation Process
Dot matrix scan of document
Automated Character Recognition Process
arriverán
Group dots in characters
Group characters to words
Compare words to dictionary
Highlight possible errors
Message editing and word search to filter nonessential

Application and Systems



(a) Original Image

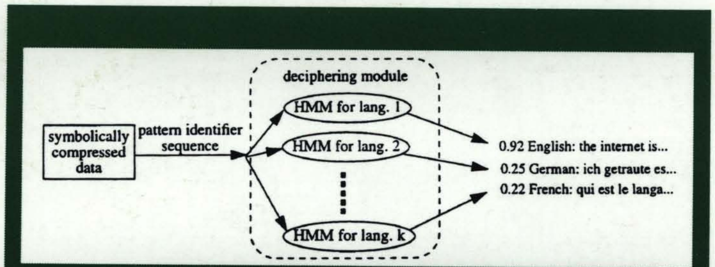
(b) Block-Averaged Image

(c) Linear Interpolated Image

(d) Cubic Spline Interpolated Image

(e) BSA Restored Image

Enhancement an Assessment


deciphering module
symbolically compressed data
pattern identifier sequence
HMM for lang. 1
HMM for lang. 2
HMM for lang. k
0.92 English: the internet is...
0.25 German: ich getraute es...
0.22 French: qui est le langa...

Information Extraction