IJDAR (2005)
Digital Object Identifier (DOI) 10.1007/s10032-004-0137-0

International Journal on Document Analysis and Recognition IJDAR

A video-based framework for the analysis of presentations/posters

A. Zandifar, R. Duraiswami, L.S. Davis

Perceptual Interfaces and Reality Lab (PIRL), University of Maryland, College Park, MD 20742, USA
e-mail: {alizand,ramani,lsd}@umiacs.umd.edu

Received: 18 December 2003 / Revised: 1 November 2004
Published online: 2 February 2005 – © Springer-Verlag 2005

Abstract. Detection and recognition of textual information in an image or video sequence is important for many applications. The increased resolution and capabilities of digital cameras and faster mobile processing allow for the development of interesting systems. We present an application based on the capture of information presented at a slide-show presentation or at a poster session. We describe the development of a system to process the textual and graphical information in such presentations. The application integrates video and image processing, document layout understanding, optical character recognition (OCR), and pattern recognition. The digital imaging device captures slides/poster images, and the computing module preprocesses and annotates the content. Various problems related to metric rectification, key-frame extraction, text detection, enhancement, and system integration are addressed. The results are promising for applications such as a mobile text reader for the visually impaired. By using powerful text-processing algorithms, we can extend this framework to other applications, e.g., document and conference archiving, camera-based semantics extraction, and ontology creation.

1 Introduction

One of the goals of computer vision is to develop systems that function in the world by understanding the objects in it and performing tasks such as navigating through it. State-of-the-art computer vision systems today can function somewhat below the capabilities of a lower life form in achieving such computer vision goals. A visually impaired person can often accomplish these functions by using other senses and simple aids. On the other hand, there are some high-level functions that a human being uses vision for that a visually impaired person might have difficulty with. Chief among these are identifying and processing text. Text in vision scenes provides an extremely rich source of already processed information that is often highly relevant to the understanding of the information a literate human or a future information-processing appliance might use to understand the world. This information is widespread in human environments such as merchandise labels, printed instructions, room numbers, street signs, newspapers, articles, and others.

On a more practical level, one of the chief methods for scientific and business communication is the use of slide shows and posters. Often, organizations or individuals record these presentations but have no means to index or retrieve these digital images by subject. In both these problems we need to be able to detect and recognize the layout of text in images and make sense of the images.

In this paper we present results from the development of a vision system for the processing of scene text in a relatively restricted context: the processing of images captured in a presentation or a poster session. Our system aims at mapping the layout of a slide or a poster into text and image blocks, performing appropriate rectification and image processing of the text blocks, followed by optical character recognition.

Such a system could be useful to a visually impaired person or for meeting archiving. Text-processing algorithms that extract latent semantics [25] have become very powerful. The availability of the text in the presentations (without having access to the digital source slides) can allow these slides to be indexed and retrieved.

2 Scenario and problems

In this paper the goal is to change information from one medium (lecture presentation/slide/poster) to another (text and graph bounding boxes followed by OCR). Here we consider that images of slides/posters are taken by a digital camera. These images are composed of text and graphic blocks and background. After image blocks are stored, the rectified text blocks are binarized and passed to OCR software. Finally, we store the detected text and images in a searchable format. Moreover, for recognized text blocks, we include the content and font size information. Prior knowledge consists of expected image layout since slides/posters consist of text/graph blocks.

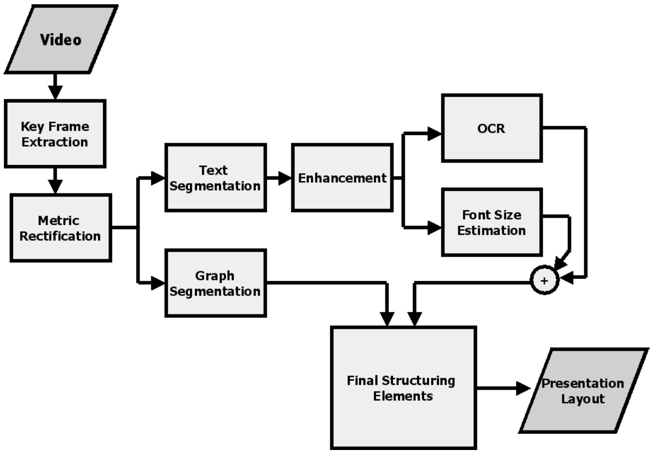


Fig. 1. Schematic of the system for annotation and analysis of lectures/posters

For off-the-shelf OCR software the output of character recognition is reliable only if the text blocks are provided in the frontoparallel view. In practice, the images are deformed when the optical axis of a camera is not perpendicular to the presentation/poster surface. Therefore, the challenge is to extract the frontoparallel view of the deformed image. This is called "metric rectification." In a frontoparallel view, right angles are projected to right angles and parallel lines are projected to parallel lines. For metric rectification, features must be found. Such features can be parallel lines and right angles in the image. Other features can be the text lines. Hence, in Sect. 4.3 we will introduce an automatic and precise line-segment-detection algorithm to detect these features. Then, text and image regions are segmented from rectified images. Before providing text boxes to OCR, we preprocess them to improve OCR output quality. Note that all these problems stated are for one image, not an image sequence. In practice, a digital camera takes a video of text printed on a surface. A video contains a lot of redundant frames with the same information. Thus the problem is how to extract changes in a video, changes in slide/poster content and not illumination change or camera jitter. The schematic of the video-based slide/poster recording framework is shown in Fig. 1.

3 Related work

Camera-based document image analysis is addressed in a recent review article [7]. The following papers touch on problems of video analysis of scene text.

Camera-based acquisition: Mirmehdi et al. [19] address a simple scheme for auto zoom of a camera. This method is useful if the background around an object has low variance compared to the object. Then, in an observation window, variance is used as an indicator of best zoom. In [31], a video-based interface to access textual information for the visually impaired is discussed, and auto-focusing and auto-zooming algorithms are presented. The best focus is achieved when the edges are strongest in the image. The best zoom is set when the readable font size of a text region is more than the OCR readable font size constraint. We consider this method for preprocessing real-time recorded video content by controlling the zoom and focus of a camera.

Key-frame extraction: Since in video of lectures textual information does not vary rapidly, we need to detect the changes in video and remove redundant frames. In [29], a simple difference operation is introduced. This algorithm is very accurate for still camera pose and constant illumination conditions. We used the phase-correlation method from the image-registration literature to detect the changes in slide or poster video content. This algorithm is stable under global illumination changes and slight camera jitter [10,14].

Metric rectification: The common method in the literature is to extract vanishing lines and right angles in an image [1,3,16]. Extraction of vanishing lines is achieved by different methods, such as the projection profile method [3] and the illusory and nonillusory lines in textual layouts [18]. We employ an automatic line segment algorithm for line detection. We cluster the line segments in feature space (edge angle and edge distance as features) using a mean shift algorithm [2]. We implement the algorithm of [16,17], which is suitable for our problem scenario since the image of a poster/slide includes rectangular boxes and lines.

Text segmentation: There are various text segmentation algorithms in the computer vision and document understanding literature that address the following three basic problems: feature extraction, clustering, and validation. For feature extraction, there are different filtering methods: steerable pyramid, Laplacian pyramids [8], Gabor filters, etc. Our system uses Gabor filters for the feature-extraction part. For clustering, we employ a K-means algorithm. More generally, a mean shift filter does not require prior knowledge of cluster numbers [2]. In [4], different features from local moments of pixel intensity are used. We use the text segmentation module in [9,30,13]. In this paper, we consider the clustering method, although for more complex problems [15], learning would be the choice.

Enhancement: A global thresholding scheme is not ideal for camera-captured images due to lighting variation and complex background [8]. The survey [26] compares eleven different adaptive thresholding methods and concludes that Niblack [24] is the best. In this paper, we apply Niblack's method for binarization of text boxes before sending them to OCR.

Contribution: While many of the individual components have been described previously, our contribution is the development of a video-based interface, a unified framework to analyze text and graphs printed in video lectures and storing them in a searchable format. Here, our video-based framework provides a more general approach to poster/presentation analysis compared to the work of [20,28]. Furthermore, we provide the qualitative results of the video-based framework in the poster/presentation scenario, which shows the performance of the framework.

4 Preprocessing

4.1 Key-frame extraction

Since, in a video of lectures, textual information does not vary rapidly, many frames will have the same information. Therefore, we do not want to waste processing resources on the redundant frames. In [29], a simple difference operation is used on three consecutive frames. The difference between two consecutive frames in time is:

$$FD(t) = \frac{1}{mn} \sum_{\forall x,y} |I(x,y;t+1) - I(x,y;t)| \,, \qquad (1)$$

where m and n are the pixel dimensions of a frame. Here, we set a frame as a key frame, if:

$$|FD(t) - FD(t-1)| > \ e \,. \qquad (2)$$

The input to the key-frame-extraction module is a video and the output is a set of sorted frames in time. This algorithm works extremely well if the camera is still and the same illumination condition holds. It often happens that the illumination varies during the lecture presentation and, moreover, there is a slight camera movement while capturing the content. In this case the simple difference algorithm fails. Our solution is to use phase correlation [14] for key-frame extraction. This method, which is well known in the image-registration literature, uses the discrete Fourier transform (DFT) of two consecutive frames to compute the overlap percentage. Consider two consecutive frames denoted as $f_1 = I(x,y;t)$ and $f_2 = I(x,y;t+1)$. Denote the DFT of these frames as $F_1(w_1,w_2)$ and $F_2(w_1,w_2)$. Then the cross power spectrum is:

$$CPM = \frac{F_1(w_1,w_2) \cdot F_2^*(w_1,w_2)}{|F_1(w_1,w_2)||F_2(w_1,w_2)|} \,. \qquad (3)$$

If f_2 is a translated version of f_1, then:

$$f_2(x,y) \approx \alpha f_1(x - \triangle x, y - \triangle y) \,, \qquad (4)$$

where α is a constant illumination factor. So the CPM is:

$$CPM \approx e^{-j2\pi(w_1 \triangle x + w_2 \triangle y)} \,. \qquad (5)$$

Therefore, the inverse of the CPM gives an impulse at $(\triangle x, \triangle y)$ and the impulse height is the amount of normalized similarity overlap between f_1 and f_2 (0 corresponds to no overlap and 1 to the maximum area overlap). This method is fast enough for real-time applications and is invariant to constant illumination changes. To suppress the repeating nature of the frequency spectrum and to give less weight to the boundary pixels, we use a raised cosine function, as a window that smoothly reaches 0 at the boundaries. This spatial filter gives more weight to pixels close to the center of an image than to the boundaries. This spatial filter (Hamming cosine window) is formulated in 1D as:

$$w(i) = 0.54 - 0.46 \cos\left(\frac{2\pi i}{N}\right) ; \quad i = 0 : N-1 \,. \qquad (6)$$

Summarizing the key-frame-extraction method for some threshold *tol* (we experimentally choose 0.2):

1. The first frame is a key frame.
2. While receiving the video sequence do:
   (a) Apply Hamming cosine window to the previous key frame and the new frame.
   (b) Compute the overlap percentage on the filtered images; if it is less than *tol*, then record the new key frame.

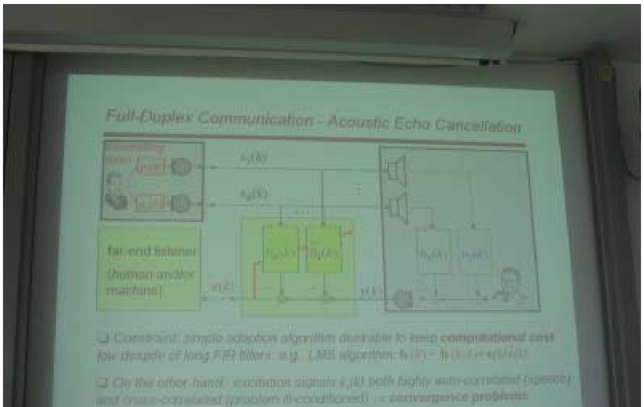
This overlap indicator is extremely efficient and robust for all types of translations and constant illumination changes. At each time step, we keep only two frames in memory and the process is very fast using the FFT (fast Fourier transform).

4.2 Metric rectification

An image of a presentation that is not frontoparallel to the image plane of a camera is deformed due to perspective projection. This distortion is called the *keystone effect.* This means parallel lines and right angles are not projected as parallel lines and right angles in the image plane (Fig. 2). For planar surfaces the deformation can be modeled by a 3 × 3 matrix, a *"homographic transformation,"* that maps the pixels of the unwarped image to the warped image [21]:

$$(u,v) \xmapsto{H} (x,y) \,, \qquad (7)$$

where H is the homographic mapping, (u,v) is the spatial location of a pixel in the image of frontoparallel view, and (x,y) is the corresponding pixel in the image captured by the camera. Knowledge of at least four corners in the image is enough to estimate the eight unknown parameters of the mapping by least-squares-estimation algorithms (up to scale) [12]. Often we do not have the exact correspondences, and also the corners may not be



Full-Duplex Communication - Acoustic Echo Cancellation

far-end listener (human and/or machine)

❑ Constraint: simple adaption algorithm desirable to keep computational cost low despite of long FIR filters, e.g., LMS algorithm: [illegible]

❑ On the other hand: excitation signals [illegible] both highly auto-correlated (speech) and cross-correlated (problem ill-conditioned) => convergence problems

[illegible]: Multichannel Acoustic Echo Cancellation

Fig. 2. Warped image: parallel lines and right angles are not perceived in, respectively, parallel and right angles in the image plane

visible. However, we can use the linear features (lines and right angles) in the image for the rectification process. In presentations, lines and boxes in the image provide such linear clues. In this paper, keystone correction is addressed by estimating vanishing lines and right angles. Before describing the algorithm, we review a few definitions from projective geometry.

Points and lines in homogenous representation: Let $\mathbf{l}$ be a line in a 2D plane denoted by $ax + by + c = 0$. A line is represented by $(a, b, c)^T$, and if a point (x, y) on a plane is represented in homogenous coordinates as $\mathbf{x} = (x, y, 1)^T$, then the line equation in the homogenous coordinates is $\mathbf{l}^T\mathbf{x} = 0$. The intersection of two lines $\mathbf{l}$ and $\mathbf{l}^{'}$ is the point $\mathbf{x}$; $\mathbf{x} = \mathbf{l} \times \mathbf{l}^{'}$. The line joining two points $\mathbf{x}$ and $\mathbf{x}^{'}$ is $\mathbf{l} = \mathbf{x} \times \mathbf{x}^{'}$. Therefore, lines and points are dual in projective geometry.

Intersection of two parallel lines: Consider two parallel lines $\mathbf{l}$ and $\mathbf{l}^{'}$ with coordinates of $(a, b, c)^T$ and $(a, b, c^{'})^T$. The intersection of two parallel lines is $\mathbf{x} = \mathbf{l} \times \mathbf{l}^{'} = (c^{'} - c)(b, -a, 0)$. Ignoring the scale factor $(c^{'} - c)$, the intersection would be $(b, -a, 0)^T$, which does not belong to R^2. In general, the intersection of two parallel lines, an *ideal point*, is of the form $(x_1, x_2, 0)^T$. The line at infinity that passes through an ideal point (from the equation $\mathbf{l}^T\mathbf{x} = 0$) is represented as $\mathbf{l}_\infty = (0, 0, 1)^T$.

Transformation of lines: If a point $\mathbf{x}'$ is mapped by a matrix H to a point $\mathbf{x}$, then we can show $\mathbf{x}' = H\mathbf{x}$, where H is a 3×3 homographic matrix, as:

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix}, \qquad (8)$$

where all entries are scaled by h_9. Therefore, if a point $\mathbf{x}$ belongs to a line $\mathbf{l}$, then:

$$\mathbf{l}^T\mathbf{x} = 0 \Rightarrow \mathbf{l}^T H^{-1} H\mathbf{x} = \mathbf{l}^{'T}\mathbf{x}' = 0\,, \qquad (9)$$

and consequently line $\mathbf{l}$ is mapped to $\mathbf{l}^{'}$ by a matrix H^{-T}:

$$\mathbf{l}^{'} = H^{-T}\mathbf{l}. \qquad (10)$$

Vanishing points and vanishing line: In a perspective image of a plane, an ideal point is mapped by a homographic transformation H to a vanishing point. A vanishing line is an image of the line at infinity in the image plane. Figure 3 demonstrates two vanishing points and the vanishing line of a perspectively skewed image. Here, we denote the two spaces: affine skewed space and perspectively skewed space E and F, respectively. Therefore, as Eq. 10 shows, we can find a transformation that maps the line at infinity in E to the vanishing line in the image (F).

Decomposition of a projective transformation: It is known that H can be decomposed into S, A, and P (similarity, affine, and projection) matrices [16]. Therefore:

$$\begin{aligned} H &= SAP, \qquad (11) \\ &= \begin{pmatrix} sR & t \\ o & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\beta} & \frac{-\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix}, \end{aligned}$$

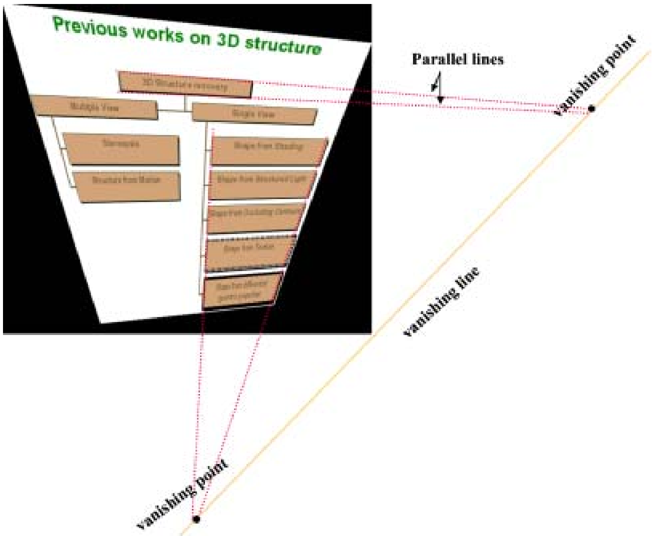


Fig. 3. Vanishing line and vanishing points

where (l_1, l_2, l_3) is a vanishing line vector in the image plane (F), $R_{2\times2}$ is the rotation matrix around the image axis of a camera, and s is the isotropic scaling. The interpretation of α and β is that they specify the image of circular points (see below). Therefore, for the metric rectification, we compute $(\alpha, \beta, l_1, l_2, l_3)^T$.

Circular points: These are points on the line at infinity that are fixed under any similarity transformation. These points are often called *absolute points* $\mathbf{I}$ and $\mathbf{J}$: $(1, \pm i, 0)^T$ denoted in the homogenous coordinates (where $i^2 = -1$). These points are the intersection of any circle with the line at infinity and are mapped to $(\alpha \mp \beta i, 1, 0)^T$ on the affine plane (E) by a matrix A and to $((\alpha \mp i\beta)l_3, l_3, -\alpha l_1 \mp i\beta l_1 - l_2)^T$ in the projective plane (F) by a matrix $(A * P)$. Unfortunately, we cannot compute circular points directly because they are complex numbers. Instead, we calculate them indirectly through their dual conic representation.

Absolute conic: It is known that the absolute conic is dual to the circular points as $C^*_\infty = \mathbf{IJ}^T + \mathbf{JI}^T$, where C^*_∞ is an absolute dual conic.

Rectification algorithm: We can solve for the metric rectification in two ways. In the first method, we extract vanishing lines and then at least two right angles for the metric rectification. We then compute the matrix P from the vanishing line and then A from two right angles. In the second method, we extract five right angles (five pairs of orthogonal lines) and solve for the image of absolute dual conics D in the projective plane. D is denoted as:

$$\begin{aligned} D &= \mathbf{MN}^T + \mathbf{NM}^T, \\ \mathbf{M}, \mathbf{N} &= ((\alpha \mp i\beta)l_3, l_3, -\alpha l_1 \mp i\beta l_1 - l_2)^T\,, \qquad (12) \end{aligned}$$

where M and N are images of circular points in the projective plane. Each pair of orthogonal lines places a linear constraint on D. From D entries, the five known unknown parameters $(\alpha, \beta, l_1, l_2, l_3)^T$ are extracted. Based on the angle between lines in projective geometry, we can show that orthogonal lines are conjugate with respect to

D. Each pair of orthogonal lines adds a linear constraint on D:

$$\mathbf{l}_a^T D \mathbf{l}_b = 0\,, \tag{13}$$

for orthogonal lines $\mathbf{l}_a$ and $\mathbf{l}_b$. In Sect. 4.3 we describe the precise line-detection algorithm we use. For more information on the details of the rectification algorithm, we refer the reader to [16].

In some cases, presentations appear on curved surfaces. These surfaces, applicable surfaces, have special differential geometric properties of vanishing Gaussian curvature at any point and isometry with flat surfaces. We address and develop the 3D structure recovery and unwarping of applicable surfaces using differential geometry in [11].

4.3 Line detection

We compute the edge map of the input image using the Robert operator [8], which is thinned by nonmaxima suppression [6]. Then, we make a feature vector with components of edge angles and edge distances. The distance used is that of an edge line segment to the center of the image and the angle is the angle of an edge line segment with respect to the horizontal axis. In feature space, we find the center of clusters using the mean shift algorithm with a large mean shift radius of the kernel [2]. Then, for each set of pixels with a specific label, we relabel each connected component. Now, the angle map and distance map of the edges are recomputed and pixels are reclustered with the small kernel radius. At the final stage, we determine endpoints of pixels with the same labels. After lines are segmented precisely, the dominant direction of the segmented lines is chosen using the histogram of the segmented line angles. Since lines of different dominant directions are assumed to be orthogonal, so we relabel a pair of orthogonal lines for the metric rectification method either method I or method II.

4.4 Text segmentation and enhancement

Unlike scanner-based systems, in camera-based OCR systems the image is low quality and blurred, so the output of OCR is poor. The quality of the image is a function of the presentation quality, the camera capturing parameters, camera motion, and so on. Here, we assume that the camera is fixed while capturing a video of lectures. Therefore, the challenge is to enhance the image before sending it to OCR. The steps are text segmentation and adaptive binarization.

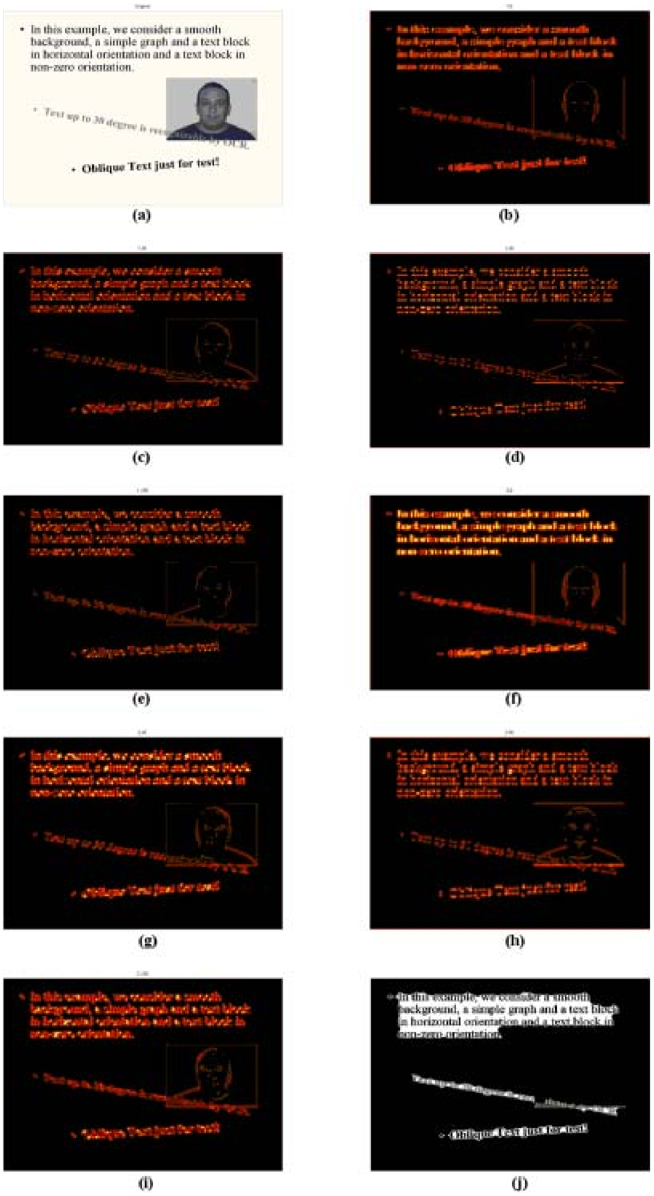

In this example, we consider a smooth background, a simple graph and a text block in horizontal orientation and a text block in non-zero orientation.
Text up to 30 degree is rec
by OCR.
Oblique Text just for test!
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

Fig. 4a–j. Original image and Gabor filter output for each scale and direction. The color bar for this figure is *red* for the minimum and *yellow* for the maximum. **a** Original image. **b** $s = 1, o = 0°$. **c** $s = 1, o = 45°$. **d** $s = 1, o = 90°$. **e**$s = 1, o = 135°$. **f** $s = 2, o = 0°$. **g** $s = 2, o = 45°$. **h** $s = 2, o = 90°$. **i** $s = 2, o = 135°$. **j** Segmented text regions. s and o are the scale and orientations, respectively

Treating text as a distinctive texture, we use Gabor filter banks associated with an edge map for text segmentation. The Gabor filter method gives both the benefits of Fourier methods and local spatial distribution methods. The feature responses of the filters at each pixel are designed to identify text-bearing regions. Although none of the filters can individually identify text and nontext regions, a concatenation of the filters provides text detection. This method is robust and precise for text segmentation in natural scenes, text in different sizes and orientations, and complex background. To improve the segmentation results, we will later introduce postprocessing algorithms on the output of the text-segmentation module. A two-dimensional Gabor function $g(x, y)$ in polar coordinates can be written as:

$$\begin{aligned} g(x, y\ ;\ \sigma_x, \sigma_y, w, \theta) = {} & \frac{1}{2\pi\sigma_x\sigma_y} \\ & \times \exp\left\{-\pi\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\} \\ & \times \exp\left\{jw(x\cos(\theta) + y\sin(\theta))\right\} \end{aligned} \tag{14}$$

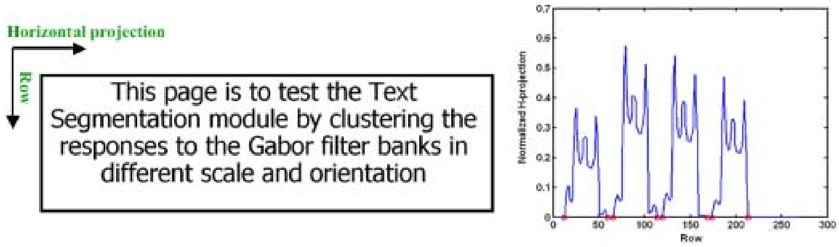


Fig. 5. Font size calculation from horizontal projection profile. *Left*: Binarized text box. *Right*: Horizontal projection profile of the complement of the image on the left; text is in *white*, background in *black*. Font size is 24 pixels

where σ_x and σ_y are the standard deviations of the Gaussian mask in the x and y directions, w_x and w_y are the center frequencies of the filter, $\theta = \tan^{-1}(\frac{w_y}{w_x})$ is the orientation, and $w = \sqrt{w_x^2 + w_y^2}$ is the radial frequency. Gabor functions with different scales and orientations form a complete but nonorthogonal basis set. Expanding an image using this basis provides a localized frequency description. In Fig. 4, the filter outcuts in two scales and four directions are shown. One of the Gabor filter's characteristics is its orientation selectivity. Assume the orientation $\theta = \theta^*$; the Gaussian mask filters the image in the θ^* orientation only and blocks other orientations. For the feature-extraction part, we choose two scales with four orientations $(0°, 45°, 90°, 135°)$ for each scale. We implement the Gabor filter bank from [27]. To increase the precision of the feature extraction part, we choose the magnitude of the responses for each pixel filtered by a nonlinear soft thresholding function of:

$$\Phi(x) = \tanh(\alpha x) = \frac{1 - \exp(-2\alpha x)}{1 + \exp(-2\alpha x)}, \qquad (15)$$

where $\alpha = 0.2$ (experimentally). We associate further a partially redundant feature, a local edge density measure. This feature improves the accuracy and robustness of the method while reducing false detections. Before the clustering step, features are normalized to have zero mean and unit variance [9].

For clustering features in 9D space, we use a K-means clustering algorithm to cluster feature vectors. Empirically, the number of clusters (value K) was set to 3. This value works well with all test images. The cluster whose center is closest to the origin of the feature vector space is labeled as background (there is no significant edge in any orientation and scale if the background is an almost uniform pattern), while the furthest one is labeled as text. If the background is not stationary or highly textured (as often happens in lecture presentations), we could learn the background and subtract it from the key-frame slide. We do not discuss this here.

The output of the clustering is filtered by a median filter to remove small noise due to the nonuniformity of the background. Using a morphological operator (closing with disk), we increase the area of text region candidates. Then we use connected component analysis to label all the text box candidates for future processing. The final stage of the text-detection module is a validation module that confirms text boxes. To increase the text-segmentation module's precision and efficiency, there are a few heuristics that are helpful in removing the outlier detected text boxes. We can remove the box if:

1. The OCR output is null.
2. The text box area is less than some threshold value. (This value is empirically set to 100 because OCR cannot read text with a width of less than 7 pixels and height of less than 13 pixels.)

Adaptive thresholding plays a key role in text image binarization. It is shown in the literature that the global thresholding scheme is not ideal for camera-captured images due to lighting variation and complex background [8]. In the histogram space, the foreground and background density functions are intermixed, so a reliable decision boundary (global threshold) cannot be achieved. With a wrong threshold, we either lose important textual information or add more unwanted edges to the OCR. We implemented the Niblack adaptive thresholding scheme to binarize each text box extracted by the text-segmentation module [24]. In this algorithm, we compute the local threshold value in a local window as:

$$T(x, y) = M(x, y) + k\sqrt{V(x, y)}, \qquad (16)$$

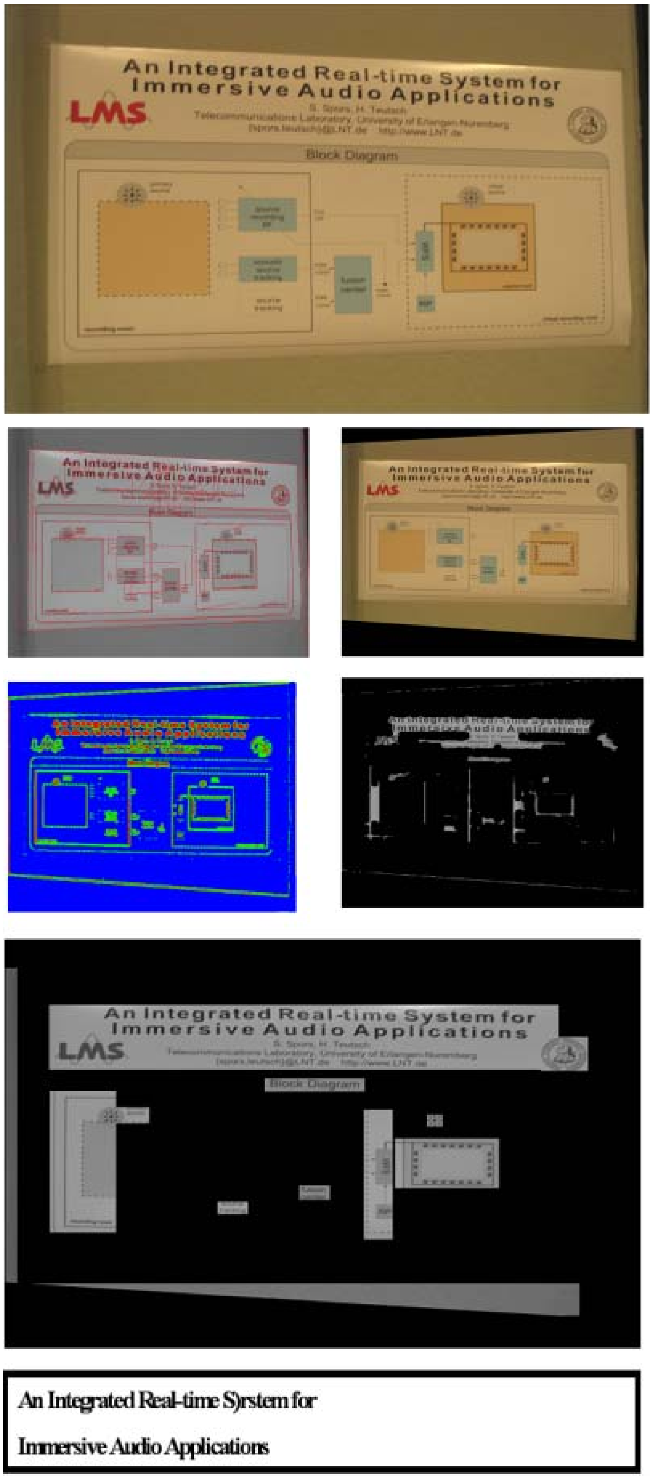
where $M(x, y)$ and $V(x, y)$ are mean and variance at each local window size w centered at pixel (x, y). The Niblack parameter k is an input parameter to the binarization module. For our system, we set k to -0.2.

4.5 Structured output

In a camera-based presentation analysis framework, we seek an annotating scheme to extract important and compressed information about slides/posters. For the text data embedded in a slide/poster we can recover the font size of each text box (like the algorithm in [31]) and its spatial location. Therefore, we can sort them in a structured format like a PowerPoint presentation, e.g., title, text box, and graph captions, for each box whose coordinate, textual content read by OCR, and font size we record. To find the font size, we calculate the horizontal projection profile of a binarized text box. Such a horizontal profile includes pulses (Fig. 5). The average font size is defined as a median over all pulse widths.

5 Implementation issues and results

We developed a video-based framework for analysis of presentations. Our integrated system consists of a Sony DFL-VL500 digital camera, Pentium III 866-MHz computer. This interface captures a video of lectures or a poster/slide and converts the content to the structured


An Integrated Real-time System for
Immersive Audio Applications
LMS
S. Spors, H. Teutsch
Telecommunications Laboratory, University of Erlangen-Nuremberg
{spors,teutsch}@LNT.de http://www.LNT.de
Block Diagram

An Integrated Real-time Syrstem for

Immersive Audio Applications

Fig. 6a–f. This example is to test the overall algorithm (pictures are **a**–**g** scanned from top to bottom and left to right). **a** Original extracted frame. **b** Detected segmented lines by mean shift algorithm. **c** Rectified image. **d** Labeled image; text-graph-background are represented in RGB. **e** Segmented text after morphological operation. **f** Text box regions. **g** OCR output. In this example, the small text box "block diagram" in **f** was not readable by OCR

format. Our software is written in MATLAB and uses IPL and the OpenCV library for image processing [22] and Scansoft2000 for OCR [23].

We tested the key-frame extraction in two ways, simulated and real video sequence. In the simulated version, we initially considered all slides of a presentation. Then, we randomly added in between the frames by a random generator, e.g., after the initial frame 1 we add 14 frames with different random uniform illumination of frame 1 after frame 1. The output of this forward simulation was 164 frames. So we applied the key-frame-extraction method and calculated the exact number of key frames, which initially was 10. We applied the same method to the video of lectures and posters with a value of 0.2 for the overlapping percentage factor, and the results were precise and robust.

The first image in Fig. 6 portrays a still image of a poster (one frame extracted from the key-frame extraction) module. We applied the automatic metric rectification algorithm described in Sect. 4.2. The OCR output by Scansoft2000 of the rectified image is shown in Fig. 6. The smaller figures are intermediate steps of the automatic metric rectification. We applied the algorithm to different images under projection and the algorithm worked extremely well if the slide/poster layout consisted of text/graph boxes.

In Fig. 7, we tested our segmentation module for different examples; original color images are on the left and the output of the text-segmentation module after the morphological operation on the right. We gathered different text sizes and orientations on different backgrounds. The first example is for complex backgrounds and highly textured graphs. The second image is of simple graphs and texts. The third image is of the different text orientations on simple background (Scansoft2000 can process up to 30° rotation). We converted color images to grayscale images. In all the cases we had the ground truth, and the missing rate was negligible. The text box font sizes in such cases were more than OCR readable font size. In the first example of Fig. 7 the rotated text was not readable by the algorithm. The rotated text in gray color space was not clear from the background pattern. These are the main results of our video-based interface:

1. Automatic metric rectification is possible because in lecture presentations/posters we have structured formats like rectangles and lines. This algorithm fails if the necessary information of parallel and orthogonal lines is missing, e.g., a slide with one line of text or a slide with only text and horizontal lines.
2. Automatic orthogonal pair detection fails if there is no clear distinction between the line angles based on the histogram of line angles.
3. The two methods of metric rectification give the same result on manual orthogonal line pairs and parallel lines. But in automatic algorithm, the second one (five-right-angle algorithm) outperforms the first one. In the first algorithm, finding the intersection of circles for many orthogonal lines is unreliable.
4. Key-frame extraction is robust and precise under uniform illumination changes. It detects major changes

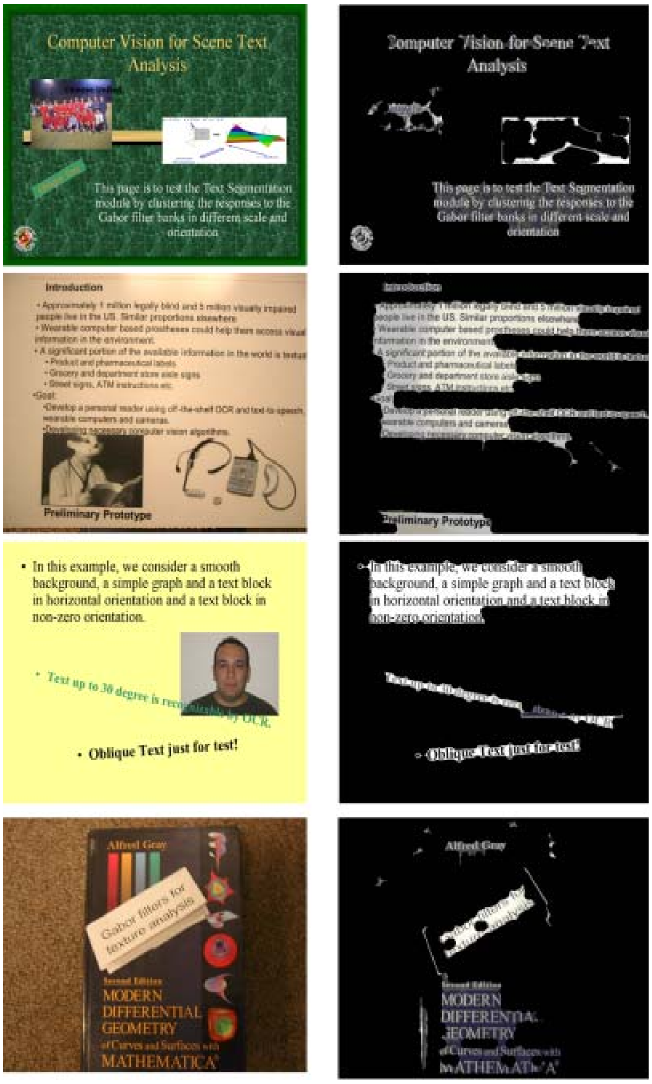

Computer Vision for Scene Text Analysis
This page is to test the Text Segmentation module by clustering the responses to the Gabor filter banks in different scale and orientation
Introduction
Preliminary Prototype
In this example, we consider a smooth background, a simple graph and a text block in horizontal orientation and a text block in non-zero orientation.
Text up to 30 degree is rec[illegible] OCR.
Oblique Text just for test!
Alfred Gray
Gabor filters for texture analysis
Second Edition
MODERN DIFFERENTIAL GEOMETRY
of Curves and Surfaces with
MATHEMATICA

Fig. 7a–d. In this example, we test the power of the text-segmentation algorithm for different presentation/poster and outdoor scene text layout. Top-to-bottom images are **a**–**d**. **a,c** Presentation slides with different text size and orientation, different graphs, and complex or simple background. **c** Image of a poster with simple background. **d** Image of a book on textured background. The results of text segmentation are shown in the second *column*

in the presentations depending on the *tol* value; with the prescribed value in this paper it cannot detect text animation in the slides.

5. The text-segmentation module works well for different text sizes and orientations and a complex background.
6. Our system is capable of reading the textual information of lecture/poster videos, detecting text box coordinates, and estimating pixel font sizes.

In Table 1, we show the overall performance of each module. The test data are a collection of 50 posters taken by a Powershot S200 digital camera (image size is 2 megapixels) and 25 presentation videos taken by a Sony DFW-VL500 digital camera (frame size 480 × 640).

Table 1. Quantitative results

Module	Hit	False	Miss
Line detection	96.1%	6.3%	3.9%
Orthogonal line detection	86.2%	16.7%	13.8%
Text segmentation	98.2%	3.3%	1.8%
Key-frame extraction	98.6%	2.6%	1.4%

The hit rate is the correct detection percentage, the false rate is the false detection percentage, and the miss rate is the missing percentage. These are the numbers for Table 1: for the key-frame-extraction module on the presentation videos, the total number of frames was 571 with 8 errors and 12 false frames. In the text-segmentation module, the detected number of words was 2026 with 36 errors and 46 false detected words. Furthermore, in the text-segmentation module, we used the word-based hit/false/missed rate for the evaluation. On the poster database, there were 693 lines with 27 errors and 44 false detected lines. The number of all orthogonal pairs was 138 with 19 error and 23 false detected orthogonal line pairs. We considered the quantitative results only on submodules, because with their failure the metric rectification result is no longer valid. For correct answers, the average angle error after rectification compared to the ground truth was very small.

6 Conclusions and future work

In this paper, we presented a system to extract textual and graphical information in lecture presentations or posters/slides using video and image processing, optical character recognition (OCR), and pattern recognition. The related computer vision problems were introduced and solved. The results were promising and efficient for the video-based interface. The indexed output is represented in structured format: text, graph, and importance of text in the content. Thus a video of a lecture or slide/poster can be compressed without losing any key information and still be small enough to be retrieved in online environments. The ability to capture and process textual information by a camera-based scanning system has many applications, e.g., mobile text reader for the visually impaired, sign detection and translation, document and conference archiving, semantic extraction, and so on. In the future, we will add more functionality to the integrated system, e.g., super-resolution enhancement [5] of key frames using temporal data, a robust algorithm to text detection, and graph segmentation algorithms.

Acknowledgements. We would like to thank Dr. Daniel Dementhon at the University of Maryland, College Park for providing the precise line-detection toolbox using a mean shift clustering algorithm. The support of NSF Grants 99-87944 and 00-86075 is gratefully acknowledged.

References

1. Clarke JC, Carlsson S, Zisserman A (1996) Detecting and tracking linear features efficiently In: Proceedings of the British Machine Vision Conference (BMVC 1996)
2. Comaniciu D, Meer P (1999) Mean shift analysis and applications. In: IEEE international conference on computer vision, pp 1197–1203
3. Clark P, Mirmehdi M (2001) Estimating the orientation and recovery of text planes in a single image. In: Proceedings of the British Machine Vision Conference, pp 421–430
4. Clark P, Mirmehdi M (2002) Recognising text in real scenes. In: Int J Doc Anal Recog 4(4):243–257
5. Capel D, Zisserman A (2000) Super-resolution enhancement of text image sequence. In: International conference on pattern recognition, 1:600–605
6. Devernay F (1995) A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical report RR 2724, INRIA
7. Doermann D, Liang J, Li H (2003) Progress in camera-based document image analysis. In: 7th international conference on document analysis and recognition, 1:606–617
8. Forsyth DA, Ponce J (2003) Computer vision: a modern approach. Prentice Hall, Englewood Cliffs, NJ
9. Ferreira S, Thillou C, Gosselin B (2003) From picture to speech: an innovative OCR application for embedded environment. In: Proceedings of the 14th ProRISC workshop on circuits, systems and signal processing (ProRISC 2003)
10. Foroosh H (Shekarforoush), Zerubia J, Berthod M (2002) Extension of phase correlation to sub-pixel registration. In: IEEE Trans Image Process 11(3):188–200
11. Gumerov N, Zandifar A, Duraiswami R, Davis LS (2004) Structure of applicable surfaces from single views. European conference on computer vision (ECCV2004), pp 482–496
12. Hartley R, Zissermann A (2000) Multiple view geometry in computer vision. Cambridge University Press, Cambridge, UK
13. Jain AK, Bhattacharjee S (1992) Text segmentation using Gabor filters for automatic document processing. In: Mach Vis Appl 5(3):169–184
14. Kuglin C, Hines D (1975) The phase correlation image alignment method. In: Proceedings of the international conference on cybernetics, 12:163–165
15. Lienhart R, Wernicke A (2002) Localizing and segmenting text in images and videos. In: IEEE Trans Circuits Syst Video Technol 12(4):256–268
16. Liebowitz D, Zisserman A (1998) Metric rectification for perspective images of planes. In: IEEE conference on computer vision and pattern recognition, pp 482–488
17. Liebowitz D (2001) Camera calibration and reconstrcution of geomtery from images. In: PhD dissertation, Oxford University
18. Pilu M (2001) Extraction of illusory linear clues in perspectively skewed documents. In: IEEE conference on computer vision and pattern recognition, pp 363–368
19. Mirmehdi M, Palmer PL, Kittler J (1997) Towards optimal zoom for automatic target recognition. In: Proceedings of the 10th SCIA, 1:447–453
20. Newman W, Dance C, Taylor A, Taylor S, Taylor M, Aldhous T (1999) CamWorks: a video-based tool for efficient capture from paper source documents. In: Procedeeings of ICMCS, pp 647–653
21. Faugeras O (1995) Stratification of 3-D vision: projetcive, affine, and metric representations. In: J Opt Soc Am 12(3):465–484
22. Intel Image Processing Open Computer Vision (OpenCV) Library `http://www.intel.com/mrl/research/opencv`
23. Scansoft2000 (OCR software) `http://www.scansoft.com/devkit/docimage.asp`
24. Taylor MJ, Dance CR (1998) Enhancement of document images from cameras. In: Proceedings of IS&T/SPIE EIDR V, pp 230–241
25. Torkkola K (2002) Discriminative features for document classification. In: Proceedings of the 16th international conference on pattern recognition, 1:472–475
26. Trier OD, Taxt T (1995) Evaluation of binarization methods for document images. In: IEEE Trans Pattern Anal Mach Intell 17(3):312–315
27. Van Hateren JH, Van der Schaaf A (1998) Independent component filters of natural images compared with simple cells in the primary visual cortex. In: Proc R Soc Lond B 265(1394):359–366
28. Wallick MN, Lobo NDV, Shah M (2000) Computer vision framework for analyzing projections from video of lectures. In: Proceedings of the ISCA 9th international conference on intellegent systems
29. Wallick MN, Lobo NDV, Shah M (2001) A system for placing videotaped and digital lectures online. In: IEEE 2001 international symposium on intelligent multimedia, video and speech processing (ISIMP)
30. Wu V, Manmatha R, Riseman EM (1999) extFinder: an automatic system to detect and recognize text in images. In: IEEE Trans Pattern Anal Mach Intell 21(11):1224–1229
31. Zandifar A, Chahine A, Duraiswami R, Davis LS (2002) Video-based interface to textual information for the visually impaired. In: IEEE international conference on multimodal interfaces (ICMI), pp 325–330

Ali Zandifar received his B.S. and M.S. in electrical engineering (Control) from the Sharif University of Technology, Tehran, Iran in 1997. He has started his Ph.D. degree in electrical engineering (communication) at the University of Maryland, College Park in September 1999. He is currently a graduate research assistant at the Perceptual Interface and Reality Lab (PIRL), University of Maryland, College Park. His research interests include shape from X, pattern recognition, statistical image processing, and differential geometry.

Ramani Duraiswami received his B.Tech. from the Indian Institute of Technology, Bombay in 1985 and his Ph.D. from The Johns Hopkins University, Baltimore, MD in 1991. He is an Assistant Professor in the Department of Computer Science, University of Maryland, College Park. He directs the Perceptual Interfaces and Reality Laboratory at the Institute for Multidisciplinary Research in Perceptual Interfaces and Virtual Reality. He has broad research interests in the areas of virtual reality, computer vision, scientific computing, modeling human audition, computational acoustics, applied mathematics, and fluid dynamics.

Larry S. Davis received his B.A. from Colgate University, Hamilton, NY, in 1970 and M.S. and Ph.D. in computer science from the University of Maryland, College Park, in 1974 and 1976, respectively. From 1977 to 1981, he was an assistant professor in the Department of Computer Science at the University of Texas, Austin. He returned to the University of Maryland as an associate professor in 1981. From 1985 to 1994 he was the director of the University of Maryland Institute for Advanced Computer Studies. He is currently a professor at the institute and in the Department of Computer Science, as well as the chair of the Department of Computer Science. He is known for his research in computer vision and high-performance computing. He has published over 75 papers in journals and supervised over 12 Ph.D. students. He is an associate editor of the International Journal of Computer Vision and an area editor for Computer Models for Image Processors: Image Understanding. Dr. Davis has served as program or general chair for most of the field's major conferences and workshops, including the Fifth International Conference on Computer Vision, the field's leading international conference.