Video database visualization

Sami Nieminen, David Doermann, Daniel DeMenthon

Language and Media Processing Laboratory, University of Maryland, 1998

ABSTRACT

In this document we present a system for visualizing video databases and some applications of such a system. The visualizing system uses dynamic queries to select the displayed data.

Visualizing, combined with dynamic queries, is an easy way for humans to interactively find data of interest. Our graphical user interface approach is tested with data on digital videos. The tests show the usefulness of the visualization approach in selection and assessment of the properties of the data.

The system is fast and efficient selection of the data of interest. The experiments show that a visualized interface is an intuitive method for browsing large databases.

We tested the system with a database of video streams. Using the visualized distribution we selected best candidates for features which well discriminate videos according to their type.

1. INTRODUCTION

Collections of digital video have become commonplace. Dealing with such large amounts of information requires tools for investigation and assessment of data. Especially, large amounts of information presented in numerical form are difficult for humans to assess.

Visualization of the data is an intuitive way for humans to understand the information. Rather than providing exact data values of an object, visualization provides trends about the object's properties. It is especially useful for comparing the values of groups of objects.

A very common task with databases is to find value types which distinguish a set of data units in to a group, separated from other groups. Visualization is an easy and simple way to accomplish this. We use a visualizing system to find discriminating features from many two-dimensional distributions.

Video database visualization has been presented by N. Vasconcelos and A. Lippman [1]. Using a simple two-coordinate feature space they find a separating threshold for romance/comedy and action type movies. Scene length and activity are the distinguishing features, and the visualization is used to confirm the valid thresholds.

Dynamic query is an interactive method for selecting data of interest. Dynamic query interfaces were developed as a mechanism for visualizing multidimensional data [2]. Queries are graphical (as opposed to textual in languages such as SQL) and provide continuous feedback to the user as the query is formulated. Dynamic queries have been presented by Tanin, Beigel, and Shneiderman [2]. They describe a method for implementing fast dynamic queries with large databases.

Our system has an interactive query interface. The system shows the results in a graphical two-dimensional coordinate display. The system performs quick visualization of the given data according to user selections. Applications of the system, among them a quick visual search for features to distinguish video types, are investigated.

We tested the system with a database of video stream information. The information contains various parameters of the stream, most notably color and audio data.

2. VIDEO DATABASE REPRESENTATION

We have developed a graphical user interface (GUI), called VideoSelector, for a database of digital video streams.

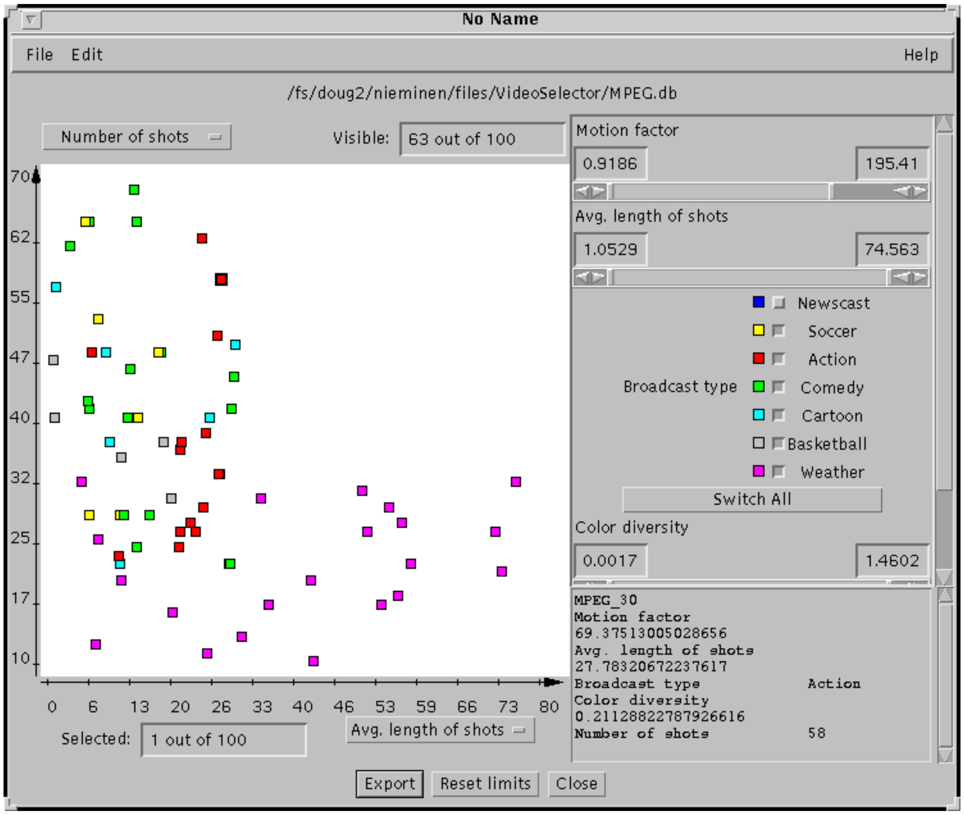


Figure 1. A visual representation of a video database.

Figure 1 shows a typical display of the VideoSelector.

The VideoSelector database contains a user-defined set of parameters for each data unit. For example, the videos in our database have values for average length of shots, the highest amplitude, and similar parameters (see chapter 4). Each data unit is represented by a glyph on the display. The glyph's position is set according to the feature values of the two coordinate axes. Clicking on a glyph highlights it and shows the parameter names and values of the unit. Clicking on the glyph again de-selects the unit.

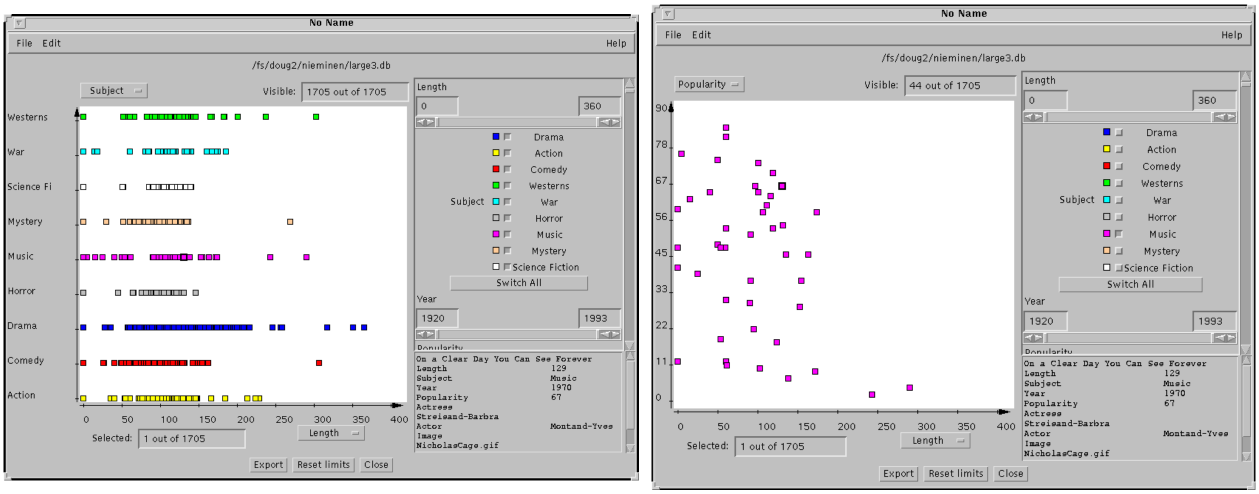
The shown data units are determined by the positions of the sliders and checkboxes in the widgets panel to the right of the coordinate display. These widgets set the ranges for the values of the data units which are shown. If a value of a unit is outside the ranges, the unit is not shown.

This property most prominently helps distinguishing the characteristics of a certain type of a video. After selecting the ranges for data of interest, the display is updated to show the units within those ranges. The checkbox widgets can be used to limit the display to certain types of data units. For example, among a group of different television broadcast videos, the user may select 'Newscast' videos.

The projection of the data into two-dimensional feature coordinates is an intuitive way for humans to visualize multidimensional data. The selectable coordinates let the user investigate the data of all pairs of data features.

The VideoSelector GUI is appropriate for browsing large databases. Dynamic queries allow step-by-step focusing to data of interest. It is an effective way to discard large amounts of unwanted data. For example, figure 2 (a) shows a

1700+ -unit movie database [8]. Figure 2 (b) shows how to the display is reduced to 44 units, showing music movies rated according to popularity. The change has been made in two steps: by de-selecting non-music movie units from display, and by changing the vertical axis to popularity.



(a) (b)

Figure 2. Display of a large movie database as a (a) full display, (b) display restricted to music movies.

The interface is implemented in the Java programming language. Java allows the transfer of the program to different computer architectures with no modifications.

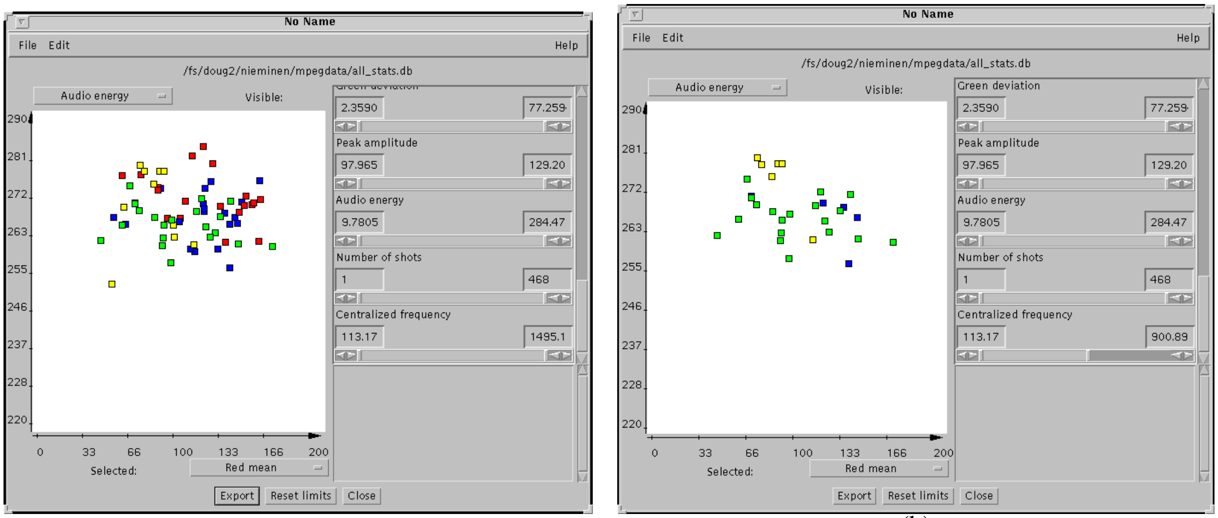
3. DYNAMIC QUERIES

The graphical user interface allows the execution of dynamic queries, i.e. queries with non-textual input, visual results and real-time processing. The query commands are implemented with selector widgets: range slider, a check box selector and a colored checkbox selector. Both selectors set a ranges, which determine the data units to display. The queries are interactive, the results are updated each time when a query is sent using the selector widgets.

An advantage of the selector widgets is that each widget controls a dimension in the feature space. The display is two-dimensional. By changing the limits for the features which are not used as the axis coordinates, the data units are seen irregularly appearing/disappearing on the display. This is helpful in inspecting the distribution of data units limited by certain characteristics in a space defined by other characteristics.

To keep the queries real-time even when the database is very large, the units of the database are sorted into many lists, one list for each feature. When a feature range is changed (either by a checkbox or by a numerical range slider), the units affected by the change are retrieved from the corresponding feature list. The lists are pre-sorted and only the affected units are updated, which makes the query result displaying fast.

For each data unit there is an index, which tells outside how many ranges it is. This index is updated each time a range change occurs. If the index is zero, the data unit is shown, otherwise it is not shown.



(a) (b)

Figure 3. Distribution of videos on red mean/audio energy axes. In (b) the same distribution, with videos with centralized frequency values over 900.89 Hz excluded.

Figure 3 shows the VideoSelector display when one of the non-displayed coordinate limits is narrowed. The data units do not disappear along coordinates axes, but it can be clearly seen that the number of Sports (red) videos is reduced by the change.

4. VIDEO FEATURES IN THE DATABASE

We selected certain features to be computed for the database videos, with the intention of creating distinguishing features for the video types present in the database.

Fifteen features for each video were entered into the database.

Table 1: Video features

Feature	Description
RGB mean	Mean value of RGB histogram
Red mean	Mean value of red histogram
Green mean	Mean value of green histogram
Blue mean	Mean value of blue histogram
Red deviation	Standard deviation of red histogram
Blue deviation	Standard deviation of blue histogram
RGB deviation	Standard deviation of RGB histogram
Green deviation	Standard deviation of green histogram
Peak amplitude	Highest amplitude of the video soundtrack
Peak frequency	Highest frequency of the video soundtrack

Table 1: Video features

Feature	Description
Audio energy	The normalized energy of the audio signal of the soundtrack
Centralized frequency	Frequency of the highest soundtrack amplitude
Number of shots	Number of shots separated by scene changes in video
Avg. length of shots	Average length of shots
Object motion	Sum of motion vector lengths which belong to areas classified as object
Type	String describing the type of the video: cartoon, music, talk or sports.
Name	Name of the video

The features were computed on a Sun Ultra 1 workstation under Solaris 2.5 operating system. Several video and audio processing tools were used.

The color information was computed by creating a large image of the keyframes of the video, and then using the 'kstats' program from the Khoros image processing package [3] to calculate the image's histogram, and the mean and standard deviation values of the histogram.

The audio features were calculated using a modified version of the Osprey [4] audio analysis program. The audio track was extracted from the MPEG data, and given to Osprey. The program calculated the peak amplitude, peak frequency, audio energy and centralized frequency of the audio signal.

The detection of number of shots, average length of shots and object motion features were developed at the Language and Media Processing Laboratory.The number of shots was computed using the MERIT scene-detection algorithm developed by Kobla [5].

The object motion information was computed using an algorithm based on DeMenthon's camera motion detection method [6]. The object motion value is computed from vectors in areas of the video which are determined to be objects. The object detection method computes all the motion vectors in a video frame. The motion vector may be caused by camera motion. Motion vectors which are not caused by camera motion are labeled to be object motion vectors. A normalized sum of the object motion vector magnitude is the object motion amount in a video.

Type is manually given label to the video according to the video's contents. Name is the name of the video file.

The input data to the database is formatted as ASCII text strings. The input has one or more data type header lines (beginning with pound '#' character'). The name of and type of the features (integer, real, string as a checkbox selection) is defined in the header, and the data in the following lines gives the comma or semicolon -separated values for each video according to the header format. There can be many headers in a datafile. Figure 4 shows an example of a data file. The data format if flexible and can contain most information present in today database applications.

```
# VideoName, Name;RGB mean, Real; RGB deviation, Real; Red mean,
#real;  Red deviation, real; Green mean, real; Green deviation, real;
#Blue mean, real; Blue deviation, real;
baseball1.mpg, 82.6031, 43.6072, 88.2172, 44.4613, 86.691, 43.3357, 72.901, 41.3231
baseball2.mpg, 84.2862, 42.8964, 88.1082, 43.3519, 89.7291, 44.0718, 75.0212, 39.6044
basketball1.mpg, 155.959, 59.6939, 162.832, 65.9264, 154.147, 61.0459, 150.897, 50.4088
birthday.mpg, 110.074, 54.0798, 114.467, 52.2891, 106.263, 50.1057, 109.492, 59.1185
# Videoname, name; Type, ColorBoxString
baseball1.mpg, Sports
baseball2.mpg, Sports
basketball1.mpg, Sports
birthday.mpg, Music
```

Figure 4. The VideoSelector data format.

5. EXPERIMENTS WITH THE SYSTEM

5.1 Video type distributions

A typical usage of the VideoSelector is to see the distribution of different types of videos. We have manually labeled a database of 70 videos into four classes: Cartoon, Music, Sports and Talk. Using visualization information we attempted to find features which discriminate video types well.

Each video type was painted in a different color on the display. Each class had 20 videos, except for music, which had ten. The 'Talk' class contains shots from debates and talk shows. The videos had fifteen different features, which were used as coordinates. Most of the videos were 60 seconds, some were several minutes long.

The features were selected to determine unique characteristics of certain types of videos. To verify these assumptions, we investigated the distributions in several feature pair coordinates.

It is natural to assume that the amount of motion in 'Sports' videos is higher than in others. Figure 5 shows the distribution in RGB mean/object motion coordinates. Object motion is the amount of object (non-background) movement in the video. Sports videos have the highest values of object motion in the distribution, which conforms well with the idea of high-motion sports scenes.The group with the second highest motion values is also distinct; most cartoon videos have higher motion than music videos or talk videos. The group with the lowest amount of motion is, as expected, talk videos.

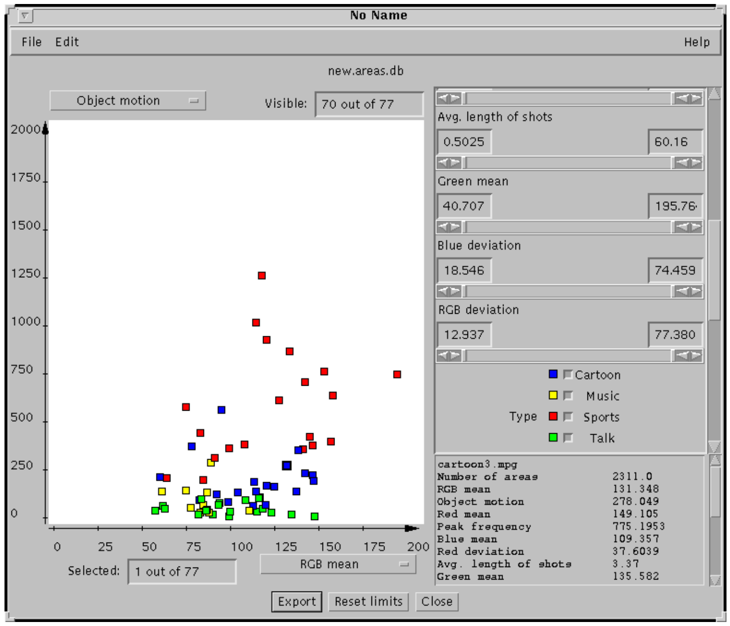


Figure 5. Distribution of videos in RGB mean/object motion coordinates.

Figure 6 shows a distribution where cartoon and talk videos are distinctly separated. The separation axis feature is ‘centralized frequency’. The ‘Cartoon’ videos have audio frequencies between 700-1500 Hz, while ‘Talk’ has frequencies between 500-950 Hz. The cartoons have mostly child-like, high frequency voices, while the talks are spoken by adults with voices of lower frequency.

The centralized frequencies of music videos were spread over the distribution. Music in general does not have any typical frequency area to discern it from other sounds.

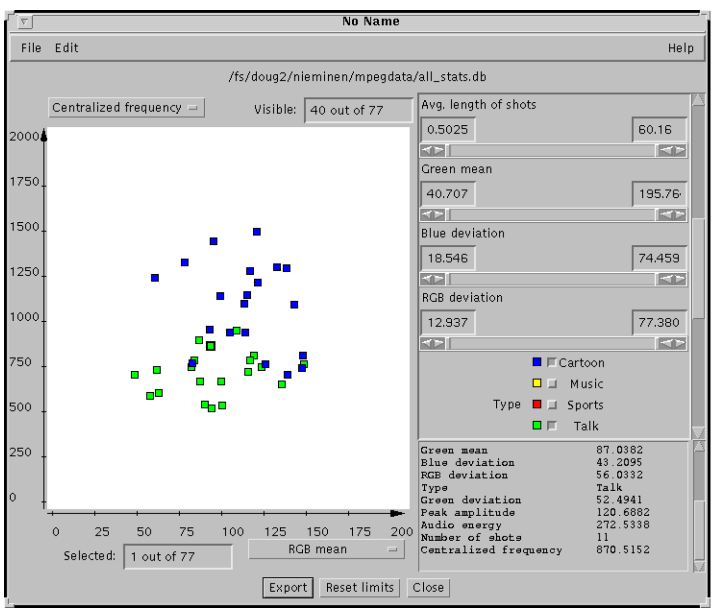


Figure 6. Distribution of music and talk videos in RGB mean/centralized frequency coordinates.

Separating individual videos within the same group also produces intuitive results. Figure 7 shows the distribution of ‘Talk’ -group videos along ‘Average length of a shot’ -axis. Although all video glyphs on the display have the same color, clicking on individual video glyphs shows the exact data of the video. The talk videos on the high end of the axis are named ‘discussion’; they are interview footage, debates and speeches.

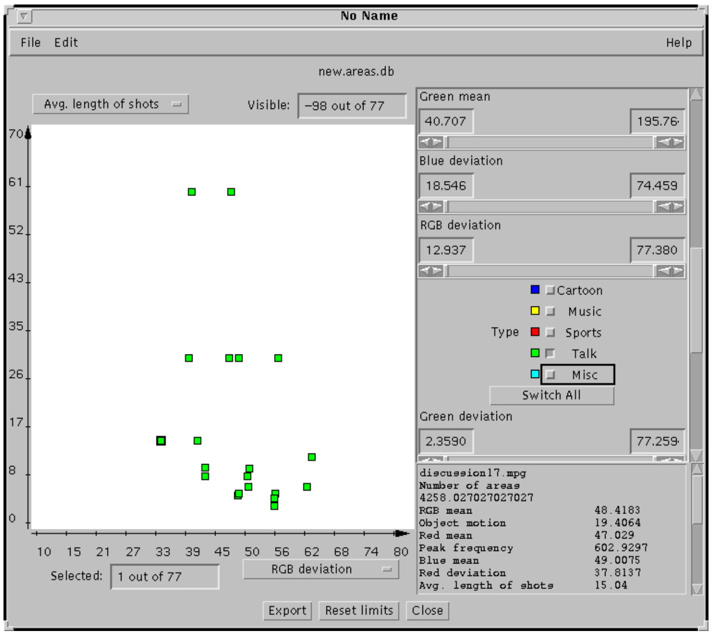


Figure 7. Talk videos along average length of shots axis.

We assumed that color histogram of animated videos differ from real-life footage. Although the red mean and RGB mean features did discriminate video types form each other, there was no clear distinction of cartoon videos from other video types. Figure 8 shows that although cartoon videos are more tightly grouped than other types, the values are distributed evenly over the whole ‘RGB Mean’ axis. Other color mean and standard deviation distributions follow the same pattern.

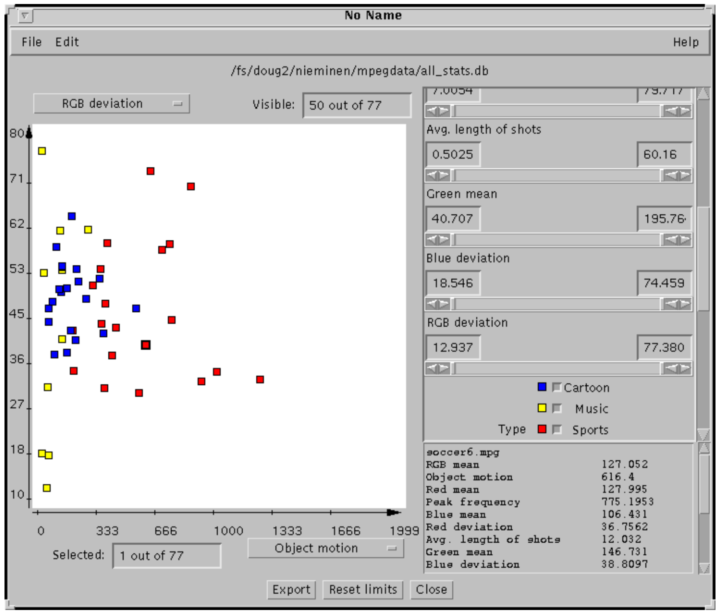


Figure 8. Animated and real-life videos in object motion/RGB deviation -coordinates.

6. CONCLUSIONS

The visualization approach is a natural interface for most databases. In addition to interactive database browsing capabilities, the system is also good to quickly find distinguishing features for different video types.

The method of visualizing the whole database first and then zooming into the data of interest is intuitive and fast compared to non-graphical query methods. By examining distributions of data in several coordinate spaces it is easy to find trends in the data, for example a relationship between motion in a video and the video type.

Although a it uses with only simple numerical and textual data types as input, the interface can be used to display most types of databases and little or no modification to the input.

7. ACKNOWLEDGMENTS

This research was partially funded by grants from GETA School of Helsinki University of Technology and from Tauno Tönningin Säätiö.

8. REFERENCES

1. N. Vasconcelos and A. Lippman. Towards semantically meaningful feature spaces for the characterization of video content. IEEE Int. conf. of image process., Los Alamitos, CA, USA, 1997, vol. 1, pp. 25-28.

2. Egemen Tanin, Richard Beigel, and Ben Shneiderman. Incremental Data Structures and Algorithms for Dynamic Query Interfaces. Fifth ACM International Conference on Information and Knowledge Management. CIKM '96, Rockville, MD, Nov. 16, 1996, 12-15.

3. Khoros Image Processing Toolkit 2.1. Khoral Research Inc., www.khoral.com.

4. David K. Mellinger: Osprey 1.3. ftp://ftp.mathworks.com/pub/contrib/v5/signal

5. V. Kobla, D.S. Doermann, and K-I. Lin, Archiving, indexing, and retrieval of video in compressed domain, In Proceedings of SPIE conference on Multimedia Storage and Archiving Systems, Volume 2916, pp. 78-89, November, 1996.

6. Daniel DeMenthon: Detecting significiant dynamic events by curve simplification in MPEG video sequences. To be published in SPIE Electronic Imaging ‘99, January 27 1999, San Jose, CA, USA.

7. Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen, and Kari Torkkola:LVQ_PAK: The Learning Vector Quantization Package. http://www.cis.hut.fi/nnrc/nnrc-programs.html.

8. Spotfire Pro 3 Third Generation Interactive Data Visualization. Sporfire Inc., www.spotfire.com.