Video Retrieval with the Informedia Digital Video Library System

Alexander Hauptmann[1], Rong Jin[1], Norman Papernick[1], Dorbin Ng[1], Yanjun Qi[1], Ricky Houghton[2], and Sue Thornton[2]

[1]School of Computer Science,
Carnegie Mellon University
Pittsburgh, PA

[2]Sonic Foundry - MediaSite Systems
Pittsburgh, PA

Background: The Informedia Digital Video Library System.

The Informedia Digital Video Library [1] was the only NSF DLI project focusing specifically on information extraction from video and audio content. Over a terabyte of online data was collected, with automatically generated metadata and indices for retrieving videos from this library. The architecture for the project was based on the premise that real-time constraints on library and associated metadata creation could be relaxed in order to realize increased automation and deeper parsing and indexing for identifying the library contents and breaking it into segments. Library creation was an offline activity, with library exploration by users occurring online and making use of the generated metadata and segmentation.

The goal of the Informedia interface was to enable quick access to relevant information in a digital video library, leveraging from derived metadata and the partitioning of the video into small segments. Figure 1 shows the IDVLS interface following a query. In this figure, a set of results is displayed at the bottom. The display includes a window containing a headline, and a pictorial menu of video segments each represented with a thumbnail image at approximately ¼ resolution of the video in the horizontal and vertical dimensions. The headline window automatically pops up whenever the mouse is positioned over a result item; the headline window for the first result is shown.

IDVLS also supports other ways of navigating and browsing the digital video library. These interface features were essential to deal with the ambiguity of the derived data generated by speech recognition, image processing, and natural language processing. Consider the filmstrip and video playback IDVLS window shown in Figure 2. For this actual video in the IDVLS library, the segmentation process failed, resulting in a thirty-minute segment. This long segment was one of the returned results for the query "Mir collision." The filmstrip in Figure 2 shows that the segment is more than just a story on the Russian space station, but rather begins with a commercial, then the weather, and then coverage of Hong Kong before addressing Mir. By overlaying the filmstrip and video playback windows with match location information, the user can quickly see that matches don't occur until later in the segment, after these other stories that were irrelevant to the query. The match bars are optionally color-coded to specific query words; in Figure 2 "Mir" matches are in red and "collision" matches in purple. When the user moved the mouse over the match bars in the filmstrip, a text window displayed the actual matching word from the transcript or Video OCR metadata for that particular match; "Mir" is shown in one such text window in Figure 2.

By investigating the distribution of match locations on the filmstrip, the user can determine the relevance of the returned result and the location of interest within the segment. The user can click on a match bar to jump directly to that point in the video segment. Hence, clicking the mouse as shown in Figure 2 would start playing the video at this mention of "Mir" with the overhead shot of people at desks. Similarly, IDVLS provided "seek to next match" and "seek to previous match" buttons in the video player allowing the user to quickly jump from one match to the next. In the example of Figure 2, these interface features allowed the user to bypass problems in segmentation and jump directly to the "Mir" story without having to first watch the opening video on other topics.

Figure 1. Text Query and Result Set in the Informedia System.

From the 11 hours of video, we extracted about 8000 shots, where a shotbreak was defined as an edited camera cut, fade or dissolve using standard color histogram measures. Instead of documents, the Video TREC track had defined shots as the unit of retrieval. We aggregated the MPEG I-frames for each shot to be alternative images for each shot. Whenever something matched to an image within a shot, the complete shot was returned as relevant. In total, there were about 80,000 images to be searched.

IDVLS Processing Components:

a. IMAGE PROCESSING

SHOT BREAKS: Color histogram analysis is applied to the MPEG-encoded video. This enables the software to identify editing effects such as cuts that mark shot changes. A single representative frame from each shot is chosen for use in poster frames or in the filmstrip view.

VIDEO OCR: The majority of traditional image processing techniques like optical character recognition (OCR) assume they work with a single image, but image processing for video works with image sequences where each image in the sequence often changes only slightly from the previous image. An overview of the Informedia Project's Video OCR (VOCR) process illustrates these points; Sato et. al discuss VOCR elsewhere in detail [14].

The goal of VOCR was to generate an accurate text representation for text superimposed on video frames. The VOCR process is as follows:

- Identify video frames containing probable text regions, in part through horizontal differential filters with binary thresholding.
- Filter the probable text region across the multiple video frames where that region is identified as containing approximately the same data. This time-based filter improves the quality of the image used as input for OCR processing.
- Use commercial OCR software to process the final filtered image of alphanumeric symbols into text. Optionally improve the text further through the use of dictionaries and thesauri.

The text detection phase can be used in key frame selection heuristics, just as face detection is used. The resulting text from VOCR processing has been used as additional metadata to document the contents of a video segment; this text can be searched just like transcript text.

OCR technology has been commercially available for many years. However, reading the text present in the video stream requires a number of processing steps in addition to the actual character recognition. First the text must be detected. The it must be extracted from the image, and finally converted into a binary black and white representation, since the commercially available OCR engines do not recognize colored text on a variably colored background. Since the extraction and binarization steps are quite noisy and do not produce perfect results, we decided to run the OCR engine on every 3rd frame where text was detected. Thus we obtained over 100 OCR results for a single occurrence of text on the screen that might last for just over 10 seconds. Frequently many of the results would be slightly different from each other, with a very high error rate. On this video collection, the word accuracy for detected text was estimated to be 27%.

FACE DETECTION AND MATCHING: The Informedia system implements the detection of faces in images as described in [2] and face matching through 'eigenfaces'. While we experimented with face recognition using a commercial system [22] as well as an implementation of Eigenfaces [15], the accuracy of face recognition in this type of video collection was so poor, that it proved useless. Therefore, we only used a face detector that reported the presence of faces in each key frame.

IMAGE MATCHING: Color histograms have been widely adopted by many image retrieval systems [5, 6,11] and, they served as the initial image query technique available to IDVLS users. While color histograms were applicable to the broad range of images accessible in the IDVLS library, their use in image indexing and retrieval revealed a number of problems. The histograms did not include any spatial information and hence were prone to false positives. Finally, they were unsuited for retrieving images in finer granularities, e.g., particular colors or regions. Referring to Figure 3, a user looking for a shot of grasslands could instead have retrieved these assorted images of predominantly blue and green colors.

b. Audio Processing

SPEECH RECOGNITION:

The audio processing component of our video retrieval system splits the audio track from the MPEG-1 encoded video file, and decodes the audio and downsamples it to 16kHz, 16bit samples. These samples are then passed to a speech recognizer. The speech recognition system we used for these experiments is a state-of-the-art large vocabulary, speaker independent speech recognizer [18]. For the purposes of this evaluation, a 64000-word language model derived from a large corpus of broadcast news transcripts was used. Previous experiments had shown the word error rate on this type of mixed documentary-style data with frequent overlap of music and speech to be just over 30%.

Figure 2. Marking the filmstrip of a query with word location in the transcript and OCR.

SPEAKER IDENTIFICATION

Our speaker identification technology is based on standard Gaussian mixture models as defined by [9]. We use the segmented method with multiple training samples derived from chunks of audio that are 30 seconds in duration. We use weighted rank scoring as defined by Markov and Nakagawa [10].

c. **Text Analysis**

Titles: Segments are scanned for words that have a high inverse document frequency, and that are strongly distinguishing segments. All text is indexed and searchable. All retrieval of textual material was done using the OKAPI formula [13]. The exact formula for the Okapi method is shown in Equation (1)

$$Sim(Q,D) = \sum_{qw \in Q} \left\{ \frac{tf(qw,D)\log(\frac{N - df(qw) + 0.5}{df(qw) + 0.5})}{0.5 + 1.5\frac{|D|}{avg_dl} + tf(qw,D)} \right\} \qquad (1)$$

where *tf*(*qw*,*D*) is the term frequency of word *qw* in document *D*, *df*(*qw*) is the document frequency for the word *qw* and *avg_dl* is the average document length for all the documents in the collection.

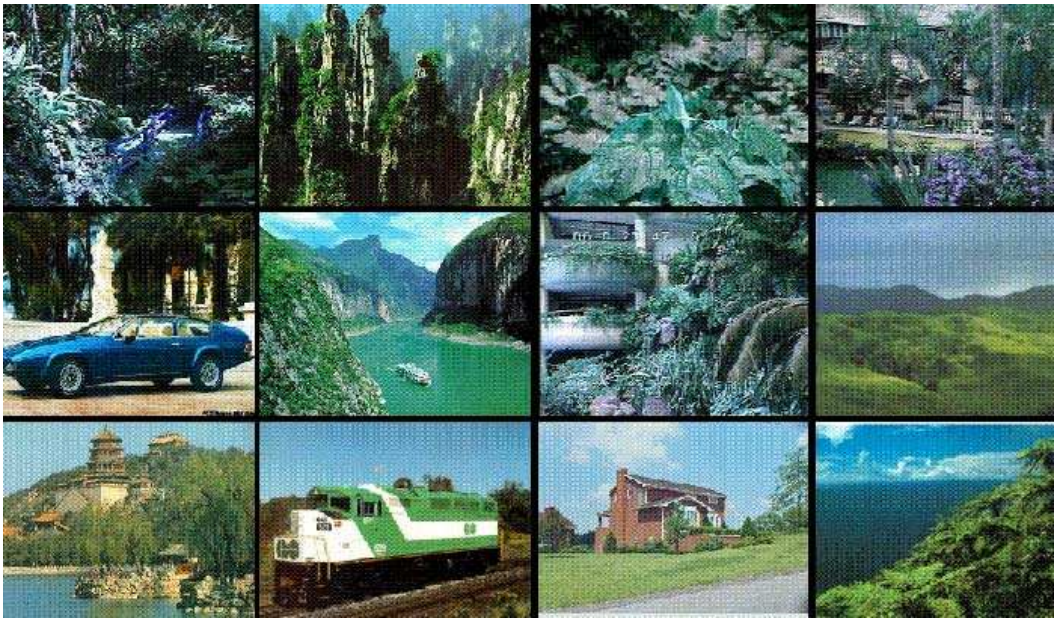

Figure 3. Result of a color histogram search

Approach to the TREC Video Track

Our approach to the Video Track in Trec was to use the Informedia system with only minor changes and see how well it would work. We treated general information queries the same as known item queries. Specific modifications are discussed in the sections for the interactive and automatic system. For simplicity, we always assumed that the unit of retrieval was a single shot.

The Interactive Retrieval System

Since Informedia only uses static images for image matching, we decided make up for this shortcoming by utilizing multiple image search engines:

- Histo144v50 Image Search

Histo144v50 is based on a simple color histogram of the target image. First, the image is converted to the Munsell color space. We are using the Munsell Color space as described in [8]. The hue is isolated. Miyahara and Yoshida describe using Godlove's formula to represent the perceptual distance between some colors in the HVC space. We are using Euclidian distance to approximate Godlove's formula. The image is broken into 9 equally sized regions. A 16-bin histogram is taken of each region. The histograms are appended to each other to form a 144 dimensional vector. The vector is then reduced in dimensionality to 50 by multiplying with a previously computed singular value decomposition. Each vector is then placed in a tree data structure that allows K-nearest-neighbors searches.

- MCPv50 Image Search

MCPv50 computes the color and texture of the target image. The image is broken into 9 equally sized regions. A 15-bin histogram is taken for the Red, Green, and Blue. Then, six texture histograms of 15 bins each are taken. All of these vectors are append to make a 1215 dim vector. This vector is reduced to 50 dim by multiplying with a previously computed singular value decomposition. Each vector is than placed in a tree data structure that allows K-nearest-neighbors searches.

- Cuebik Image Search

Cuebik is based on one of the behaviors of the IBM QBIC image search engine. A palette of 255 colors is chosen for a database by marking the strongest colors found in a large sample of images. The target image is reduced to 256 equally sized regions. Each region is mapped to one of the palette colors, and recorded. A search is done by choosing a set of regions and finding all images that have the same color in the same region.

In addition to the above image search engines, we also used a downloadable version of the original IBM QBIC system as well as a search engine provided by James Wang from the University of Pennsylvania.

The search process foe each interactive query was as follows:

1. Determine key words in the text description of the query and use Video OCR text search to find them.
2. Use the supplied query images to initiate a search for relevant segments.
3. If a segment key frame or title looks related to the answer, open up its filmstrip and view details.
4. If the segment filmstrip looks related to the topic, but does not provide an answer, look one segment forward and back. If the topic in the adjacent segment is the same, scan the filmstrip of an additional segment forward or back.
5. If a frame answers the query, use that frame for relevance feedback with each of the image search engines to find more like it.
6. If a frame seems to be related, but does not answer the question, use that frame with each of the search engines to find more like it.
7. Repeat all steps as needed.

Automatic Retrieval

In the following we will elaborate only on the known item query set, because comprehensive relevance judgments were available for this set allowing automatic estimation of precision and recall for variations of our video retrieval system. The 34 known item queries are distinguished from the remaining 'general search' queries in that the information need tends to be more focused and all instances of query-relevant items in the corpus are known. This allows an experimental comparison of systems without the need for further human evaluations.

Since the evaluation could be done automatically, the top 100 search results were scored for all systems. The general unit of retrieval was a 'shot', in other words a time range between two shot changes, for example editing cuts and fades. Systems had to determine shot changes automatically. An item was considered relevant if at least 33% of the length of the returned item overlapped with the target item in the list of shots relevant to this query and less than 33% of the time range for the returned item was outside the target range. This requirement ensured a reasonable overlap of the returned shot with the target shot. An example of a typical query is shown in Figure 4. This query is to be used for automatic systems, but not for interactive evaluations. It is a known item query, indicating that all results are known inside the video collection. According to the text description, the query is looking for video scenes of water skiing, and gives an example of the type of video that is desired. Two images from the example video are extracted and also shown in Figure 4.

A