A Synergistic Approach to Efficient Interactive Video Retrieval

Andreas Girgensohn, John Adcock, Matthew Cooper, and Lynn Wilcox

FX Palo Alto Laboratory, 3400 Hillview Avenue, Bldg. 4, Palo Alto, CA 94304, USA
{andreasg, adcock, cooper, wilcox}@fxpal.com

Abstract. A video database can contain a large number of videos ranging from several minutes to several hours in length. Typically, it is not sufficient to search just for relevant videos, because the task still remains to find the relevant clip, typically less than one minute of length, within the video. This makes it important to direct the users attention to the most promising material and to indicate what material they already investigated. Based on this premise, we created a video search system with a powerful and flexible user interface that incorporates dynamic visualizations of the underlying multimedia objects. The system employes an automatic story segmentation, combines text and visual search, and displays search results in ranked sets of story keyframe collages. By adapting the keyframe collages based on query relevance and indicating which portions of the video have already been explored, we enable users to quickly find relevant sections. We tested our system as part of the NIST TRECVID interactive search evaluation, and found that our user interface enabled users to find more relevant results within the allotted time than other systems employing more sophisticated analysis techniques but less helpful user interfaces.

1 Introduction

Users such as intelligence analysts often need to find video clips related to a particular topic that is described using both text and images. This type of video search is difficult, because users need visual information such as keyframes or even video playback to judge the relevance of a video clip and text search alone is not sufficient to find the desired clip within a video. While searching text documents is a well-studied process, it is less clear how to best support search in video collections. Typically text documents are treated as units for the purpose of retrieval, so that a search returns a number of relevant documents. The user can then easily skim the documents to find parts of interest. In cases where documents are long, there are techniques to search for just the relevant sections [16].

However, treating entire videos as units of retrieval will often not lead to satisfactory results. After retrieving relevant videos, the task still remains to find the relevant clip, typically less than one minute of length, within the video. Even when such videos are broken into sections, or stories of several minutes in length, it is still time consuming to view all those video sections to find just the relevant clip.

M.F. Costabile and F. Paternò (Eds.): INTERACT 2005, LNCS 3585, pp. 781 – 794, 2005.

Our approach to this problem is to support users in rapidly searching through such video collections. Our target users are analysts who need both visual and textual information or video producers who want to locate video segments for reuse. While the latter will frequently use libraries that support retrieval with extensive meta-data describing properties such as location, included actors, time of day, lighting conditions, our goal is to support the search in video collections where such meta-data is not available. In this work, we assume that time-aligned text, such as transcripts, automatically recognized speech, or closed captions, is available.

Our system design uses a synergistic approach that has the system and the user collaborate on improving the search results. We automate certain parts of the system but to let the users directly perform tasks that humans can do better. For example, the system can retrieve all video containing a particular keyword, but the user can more easily look through keyframes representing the video and find just those of interest. Our system makes novel contributions for the user interface design for video search systems. We use several visualization techniques to direct the users' attention to potentially relevant material and to let them judge quickly what is truly relevant. We also make novel contributions to the video search back-end by providing a story segmentation for automatically recognized speech and by determining terms related to the query in latent semantic text search where the retrieved text passage might not share any terms with the query.

In the next section, we discuss related work. We then describe the setup for a retrieval experiment and our search user interface. Next, we present the components of the back-end search system. Finally, we present the results of the TRECVID evaluation and conclude with a discussion of the implications.

2 Related Work

There is currently a great deal of interest in video search, as evidenced by recently unveiled web-based video search portals by Yahoo [20] and Google [9]. 2004 marked the 4th year of the TRECVID [19] evaluations which draws a wide variety of participants from academia and industry. Some of the more successful ongoing efforts in the interactive search task draw upon expertise in video feature identification and content-based retrieval. The Dublin City University effort [6] includes an image-plus-text search facility and a relevance feedback facility for query refinement. The searcher decides which aspects of video or image similarity to incorporate for each query example. The Imperial College interactive search system [11] likewise gives the searcher control over the weighting of various image features for example-based search, a relevance feedback system for query refinement, and notably incorporates the NN^k visualization system for browsing for shots "close" to a selected shot. The MediaMill, University of Amsterdam system [18] is founded on a very powerful semantic concept detection system and searchers can search by concept as well as keyword and example. Likewise the Informedia system from Carnegie Mellon University [5] incorporates their very mature technology for image and video feature detection and puts the searcher in control of the relative weighting of these aspects. We previously reported preliminary results of our approach [8].

Our effort is distinguished from others primarily by the simplicity of our search and relevance feedback controls in favor of an emphasis on rich interfaces and intuitive paths for exploration from search results. Our scenario is not so much one of query followed by refinement as it is query followed by exploration. Whether explicitly stated or not, a goal in all of these systems is a positive user experience. That is, an informative and highly responsive interface cannot be taken for granted when handling thousands of keyframe images and tens of gigabytes of digital video.

3 Retrieval Experiment

To validate our approach, we participated in the interactive search component of a video retrieval evaluation called TRECVID sponsored by the National Institute of Standards and Technology (NIST) [19]. In the interactive search, participants have access to broadcast news video from four months from the U.S. ABC and CNN networks (128 videos; about 60 hours). The TRECVID evaluation consists of 24 topics such as "find shots of Bill Clinton speaking with at least part of a US flag visible behind him." Users are given 15 minutes for each topic, and must find all video passages relevant to the topic. Some of the TRECVID participants use very elaborate video analysis techniques to support the search [10]. For example, one very successful system allows the user to search for visual features such as animals, buildings, or people [4].

For our retrieval experiments, we used automatically recognized speech from the news videos as time-aligned text. A few errors in the recognized speech do not have a major impact on the retrieval results because stories tend to include important terms repeatedly. We provided both literal and latent semantic text search. The former uses the term frequency (tf; the count for a term in a document) and the inverse document frequency (idf; the count of documents containing a term) as measures of relevance [17]. The latter maps all terms into a reduced-dimensional space such that related terms are placed near each other [1]. Literal search is well suited to searching for proper names whereas latent semantic search is more useful when searching for concepts that can be described with different words, and the exact words appearing in the transcript are unknown to the searcher.

The basic retrieval units are video shots that are uninterrupted sequences with strong visual coherence, generally taken by a single camera [2]. In the news video collection, shots have an average length of six seconds. Those shots are of insufficient length for performing text retrieval on the text associated with them. Because each half hour news video deals with a wide variety of topics, using whole videos as text documents for retrieval is not appropriate, either. Instead, our system pre-processes the text transcript to segment each video into smaller semantically-related units (stories) that are of a length better suited for standard text retrieval techniques. Each story has several associated video shots that can be accessed through the story. Videos, stories and shots form a three level hierarchy. Our application can also support hierarchies with more or fewer levels if that is more appropriate for the video material to be searched.

We also provide support for image similarity search to deal with situations where the visual information is more important than the associated text (e.g., to find

sunsets). In this case, the user selects an image that represents his visual information need and the system searches through the keyframes representing the individual video shots. The system returns those shots whose keyframes have a strong visual similarity to the image supplied by the user. We use color correlograms [12] for our similarity measure. To support our hierarchy of stories and shots, the visual similarity search results are propagated from shots to the stories containing them.

4 User Interface

A typical search in a moderate to large video collection can return a large number of results. This is the result of returning relatively short segments of video that are visually and/or semantically coherent. Our user interface directs the user's attention to the video segments that are potentially relevant. We present results in a form that enables users to quickly decide which of the results best satisfy the user's original information need. Our system displays search results in a highly visual form that makes it easy for users to determine which results are truly relevant.

The basic retrieval units in our system are video shots. Because the frames in a video shot are visually coherent, each shot can be visualized with a single keyframe. A keyframe is an image that visually represents the shot, typically chosen as a representative from the frames in the shot [2]. Time-aligned automatic speech recognition (ASR) output is used to assess the semantic content of each shot. But because shots are too short to be used as units of meaningful content, we use automatically segmented stories as the main retrieval units. Adjacent shots with relatively high text-based similarity are grouped into stories. These stories form the organizing units upon which video shots are presented in our interface. Because each story consists of several shots, it cannot be well represented by a single keyframe. Instead, we represent stories as collages of shot keyframes.

Figure 1 shows the interface for the interactive search. The user enters a query as keywords and/or images (Figure 1B). Keywords are typed and images are dragged into the query section from other parts of the interface. For the TRECVID task, the topic is displayed in Figure 1C. In this case, the user can select keywords and images from the topic description. Once the user has entered a query and pressed the search button, story results appear in Figure 1A, displayed in relevance order. The size of each story icon is also determined by query relevance. A novel feature of our system is that retrieved stories are represented by keyframe collages where keyframes are selected and sized by their relevance to a query so that the same story may be shown differently for different queries. When the user wants to explore a retrieved story, he clicks on the collage. The parent video is opened and the selected story is highlighted in the video timeline (Figure 1E). Below the timeline the keyframes from all the shots in the selected story are displayed (see Figure 1F). The shot or story under the mouse is magnified in the space in Figure 1D. A tool tip provides additional information for the shot or story under the mouse. When the user finds a shot of interest, he drags it to the area shown in Figure 1G to save relevant results. Another novel aspect of our system is that we mark visited stories so that the user can avoid needless revisiting of stories. We present the three types of UI elements that we developed to surface the novel features:

Fig. 1. The interactive search interface. (A) Story keyframe summaries in the search results (B) Search text and image entry (C) TRECVID topic display (D) Media player and keyframe zoom (E) Story timeline (F) Shot keyframes (G) Relevant shot list.

1. Three visualizations provide different information perspectives about query results.
2. Tooltips and magnified keyframes provide users with document information relevant to the query.
3. Overlays provide cues about previously visited stories, current story and shot in video playback, and the degree of query relevance on story and shot.

4.1 Query Result Visualizations: Story Collage, Shot Keyframe, Video Timeline

Query results are returned as a set of stories, sorted by relevance to the query. Each story is represented by a collage of keyframes from the video shots contained in the story. The size of the collage is determined by the relevance to the query so that one can see at a glance which stories are most relevant. We use a collage of four keyframes to give a flavor of the different shots in a story without making the keyframes too small for recognizing details. We use rectangular areas for the keyframes for the sake of fast computation but we could instead use other collages such as a stained glass window visualization [3].

Fig. 2. Story keyframe montage example. The keyframe montage at the right is constructed from the 15 shot keyframes of the story at the left selected and cropped based on their relevance to the query "Boris Yeltsin".

In addition to determining the relevance of stories with respect to the query, we also determine the relevance of each video shot. While the shot relevance does not provide good results on its own, it can be used to determine which shots in a story are the most relevant ones. The most-relevant shots are selected and their keyframes are combined to form a story keyframe-collage. The size allotted to each portion in this 4-image montage is determined by the shot's score relative to the query. Figure 2 shows an example of this where the query was "Boris Yeltsin" and the shots most relevant to the query are allocated more room in the story thumbnail, in this case the 2 shots of the 9 total shots in the story that depict Boris Yeltsin. Rather than scaling down the keyframes, they are cropped to preserve details in reduced-size representations. In the current implementation, the top-center portion of the cropped frame is used but we plan to crop the main region-of-interest with face or motion detection.

Because the automatic story segmentation is not always accurate and related stories frequently are located in the same part of the video, we provide access to the temporal neighborhood of the selected story. First, the timeline of the video containing the story color-codes the relevance of all stories in the video (see Figure 1E and Figure 3). This color-coding provides a very distinct pattern in the case of literal text search because only few stories contain the exact keywords. After a latent semantic text search, all parts of the timeline indicate some relevance because every term has some latent relationship to all other terms. We experimentally determined a nonlinear mapping of the relevance scores from latent semantic text search that highlights the most related stories without completely suppressing other potentially related stories. Immediately below the timeline in Figure 1E collages of neighboring stories around the selected story are displayed. This provides quick

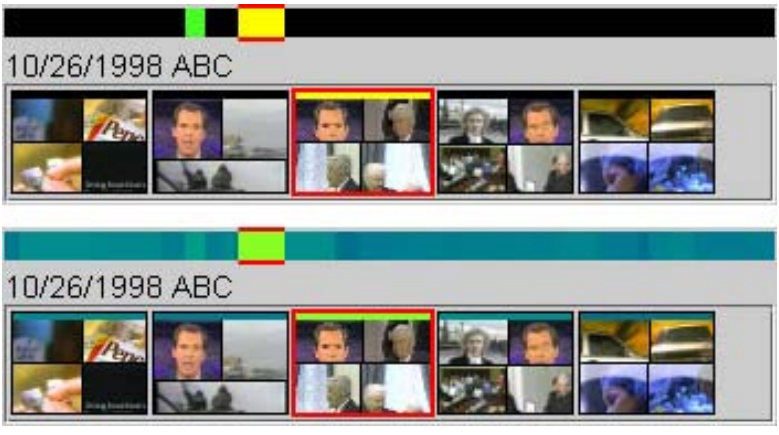

Fig. 3. Timelines for the query "Boris Yeltsin". Brighter colors indicate more relevance. The literal text search timeline above displays two distinct relevant areas whereas the latent-semantic search timeline below indicates some amount of relevance everywhere.

access to keywords in those stories via tool tips. By clicking on the timeline or the neighboring collages, the corresponding story can be selected.

The keyframes for the shots comprising the selected story are shown in a separate pane (see Figure 1F and Figure 4). Double-clicking a keyframe plays the corresponding video shot. The expanded view provides access to the individual shots for play-back, for adding them to the results, and for displaying information about the shots. One or more keyframes of shots can be dragged into or out of the result area to add or remove them from the result list, or into or out of the image search area to add or remove them from the image search. Shots can also be marked explicitly as irrelevant. Such shots are excluded from being automatically added to the results when the user selects the "Add related" button.

4.2 Document Relevance Feedback: Tooltips and Magnified Keyframes

It is useful to provide feedback to the user to indicate why a particular document was deemed relevant to the query and how the document is different from other documents. Tooltips for story collage and video shot keyframes provide that information to the user in the form of keywords that are distinctive for the story and keywords related to the query (see the plain text in Figure 4). Terms that occur frequently in the story or shot and do not appear in many other stories or shots are most distinguishing. While words such as "lately" do not really help in distinguishing the video passage from others, words such as "russia" are helpful. By displaying five keywords, it is likely that at least one or two are truly useful.

The terms in bold are most related to the query and indicate why the document is relevant to the query. We decided against displaying the terms with surrounding text as it is frequently done in Web search engines. The reason is that we do not want the tool-tips to be overly large. Furthermore, the automatic speech recognition makes mistakes that are more noticeable when displaying whole phrases.

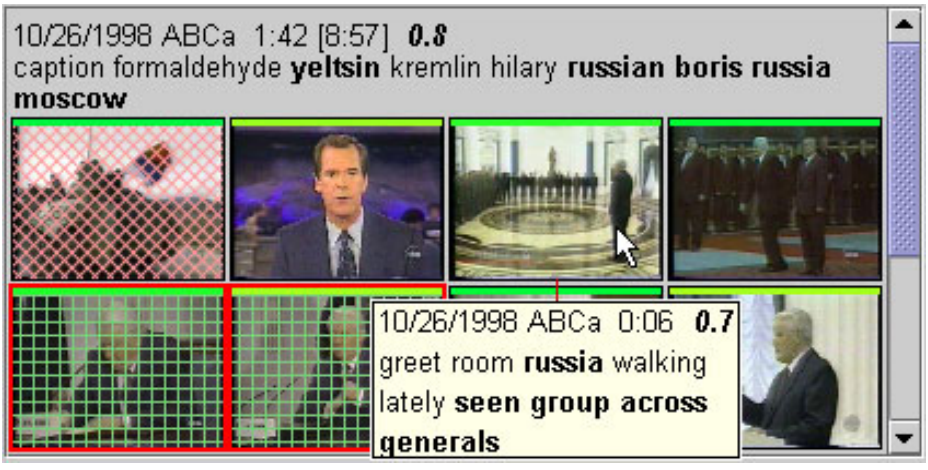

Fig. 4. Tool tip showing distinguishing keywords and bold query keywords

With a literal text search approach, the terms most related to the query are the query terms appearing in the story. When latent semantic text search is used, a relevant document may not contain any of the query terms but terms that are closely related to them. We use the latent semantic space to identify terms in the document that are most similar to the query.

In an earlier version of our application, we displayed keyframes as part of the tool-tips. Users interacting with that version of the application found that the keyframes were either too small to be useful or that the tooltips covered up too much the window. To address this issue, we decided to reuse the video player area as a magnifier for the keyframe under the mouse or the selected keyframe (see Figure 1D). Usually, the video player will be stopped while the user inspects keyframes so that the user can see a magnified version of the keyframe or collage without the need to dedicate some window area for that purpose.

4.3 Overlay Cues: Visited Story, Current Playback Position, Query Relevance

Semi-transparent overlays are used to provide three cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 1A and E). A translucent red overlay on a shot icon indicates that it has been explicitly excluded by the user from the relevant shot set. A translucent green overlay on a shot icon indicates that it has been included in the results set (see Figure 1F). Figure 4 shows the use of patterns instead of translucent overlays for color-blind users and grayscale reproduction of the image. Red diagonal lines indicate exclusion and green horizontal and vertical lines indicate inclusion.

While video is playing, the shot and the story containing the current playback position are indicated by placing a red dot on top of their keyframes. The playback position is also indicated in the timeline by a vertical red line.

Horizontal colored bars are used along the top of stories and shots to indicate the degree of query-relevance, varying from black to bright green. The same color scheme is used in the timeline depicted in Figure 3.

5 Back-End Search System

We pre-process videos to segment them into stories with a text-based latent semantic analysis (LSA) of the text transcripts [1]. For a topic of interest such as the topics provided by TRECVID, users need to issue several queries to find the relevant video shots. We give users the choice among literal keyword text search, LSA-based text search, visual similarity search, or a combination of text and visual similarity search.

At the completion of a topic, the system uses the query history and list of relevant shots to automatically find additional relevant video shots to add to the results.

5.1 Data Pre-processing

As the lowest-level unit, we use video shots that are provided as a reference by TRECVID [15]. Video frames in a shot have strong visual coherence, i.e., the video only changes because of movement in the scene or pans and zooms. We perform an automatic pre-processing step to identify topic or story units from the automatically recognized speech. We use latent semantic analysis (LSA) [1] to improve the performance of the segmentation. LSA turns a large matrix of term-document association data into a "semantic" space wherein terms and documents that are closely associated are placed near one another. Singular-value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. We use a reduced space of with 100 dimensions because that accounts for most of the variance. As a result, terms that did not actually appear in a document may still end up close to the document.

For the story segmentation, we build a latent semantic space (LSS) treating the stopped and stemmed [14] text tokens for each video shot in the testing corpus as a separate document. We then project the text for each shot into this shot-based LSS. This results in a low-dimensional representation for each shot in term of its projection coefficients in the LSS. We then group adjacent video shots into stories following the similarity-based approach of [7]. The similarity between pairs of shots is quantitatively assessed using the cosine similarity between the corresponding vectors of projection coefficients. A similarity matrix is constructed with the (i,j) element equal to the similarity between the i^{th} and j^{th} shots. Areas with high self-similarity appear as dark squares along the diagonal of the matrix. Boundaries between groups of shots with high similarity appear as checkerboards in the similarity matrix (see the left of Figure 5). This is because shots contained in the same story exhibit high (within-story) similarity. Shots from different stories exhibit low (inter-story) similarity. A checkerboard kernel is moved along the main diagonal of the matrix to locate boundaries. Only the part of the matrix that overlaps the moving kernel needs to be computed. The checkerboard kernel acts as a matched filter; the shot-indexed kernel correlation score exhibits local maxima at the boundaries between stories. The points of highest kernel correlation are chosen as story boundaries subject to heuristic

constraints on the minimum and maximum length of a story. After determining story boundaries, we create a new LSS treating each story as a document.

5.2 Search Engine

Queries are specified as a combination of text and images. The searcher can opt to perform a text-only or image-only search by leaving the image or text query area empty. For the text portion of the query, the searcher can choose either a literal keyword text search or a LSA-based text search. Literal text search performs better for proper names (e.g., of persons) whereas latent semantic search can find related concepts.

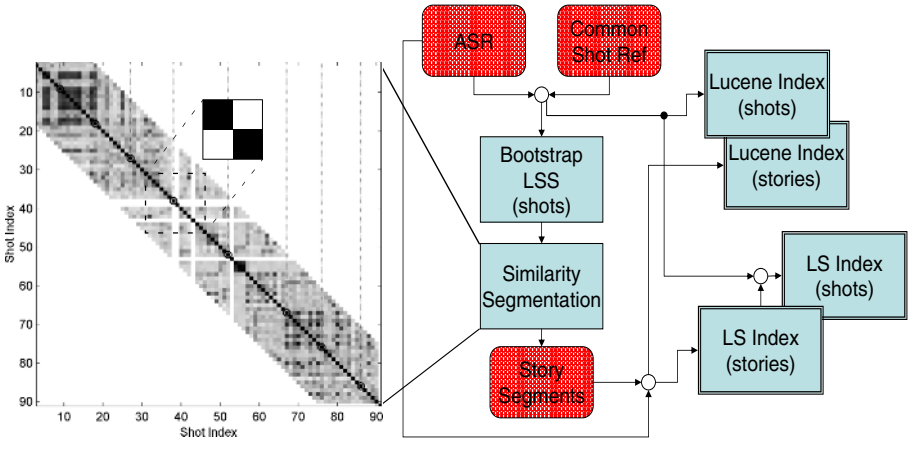

Fig. 5. Data preprocessing flow. A story-level segmentation is derived from the reference shot boundaries and the self-similarity matrix of the text transcripts. Dark areas in the similarity matrix on the left indicate high similarity and a checkerboard kernel finds boundaries between those areas. Both LS and Lucene indices are created at both the shot and story levels.

When determining text-query relevance for shots, the shots inherit part of the retrieval score of their parent stories to properly handle terms that co-occur in the same story but in different shots. We use only automatically recognized speech to provide text for story and shot segments. The literal text search is based on a Lucene [13] back end and ranks each story based on the tf-idf values of the specified keywords [17]. In this mode the story relevance, used for results sorting and thumbnail scaling and color coding as described in Section 4, is determined by the Lucene retrieval score. When the LSA-based search is used [1], the query terms are projected into a latent semantic space (LSS) like the one used in the story segmentation created from the detected stories. The query terms are scored in the reduced dimension space against the text for each story and each shot using a cosine similarity function. In this mode, the cosine similarity value becomes the query relevance score.

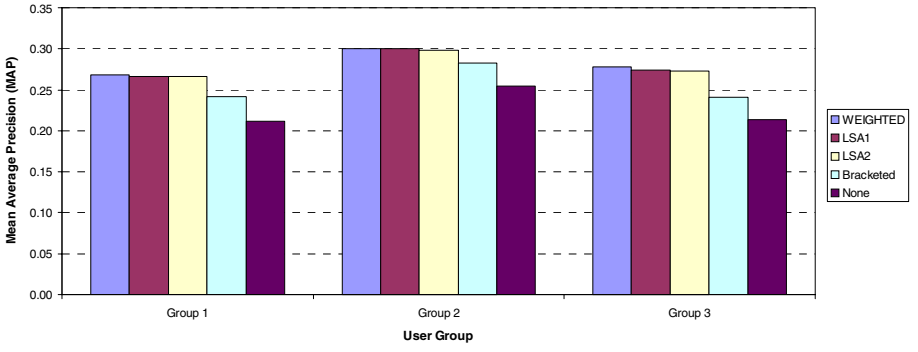

Fig. 6. Overall mean average precision (MAP) performance by user group and post-processing system type employed. The "None" column is the MAP performance of the user selected shots without any automatic augmentation.

For a literal text search, it is common to highlight the matching keywords from the query in the search results to provide an indication of the context in which the document and the query matched. To achieve a similar effect for a latent semantic text search, we project the query term vector (which is likely to have very few entries since most user-entered queries use few terms) into the reduced dimension space and expand it back into the full term-vector space with the inverse of the projection matrix. This produces a dense query vector. Next, to identify keywords, first eliminate terms from the dense query vector that do not occur in the search document, then choose some number of the remaining terms with the highest corresponding values in the dense query vector as query-related "keywords". These terms can now be used to highlight the context of the similarity between the query and the returned document.

An image similarity matching capability is provided using color correlograms [12]. Correlograms provide signatures for color groupings in images and tend to produce better image search results than color histograms or similar measures. During a visual search, the correlograms of the search images are compared to the correlograms of the keyframes of the video shots. To generate an image-similarity relevance score at the story level, the maximum score from the component shots is propagated to the story.

5.3 Post Query Processing

The goal of the TRECVID interactive search evaluation is to find all relevant shots. To aid the user in this, the system attempts to find additional relevant shots after the searcher finishes searching for video shots relevant to a topic. We use two strategies to select additional shots. First, we address the fact that shots are sometimes segmented at the wrong place by adding all shots bracketing the shots selected by the user *(Bracketed)*. Second, we issue additional queries to find shots similar to the ones the user selected. We use three variants for the second strategy. The first variant *(WEIGHTED)* uses the weighted average of the scores of all queries issued by the user to compute a new score for every video shot in the collection. Each individual query's scores are weighted by the recall of that query as judged against the user-identified list of relevant shots. The second variant *(LSA1)* combines the text from all

user-selected shots to form a single LSA query and we add the best results from that query. The third variant *(LSA2)* uses the text from every user-selected shot to form a separate LSA query and combines these separate query results as in the WEIGHTED method.

The WEIGHTED method is also used when the user presses the "Add related" button (see Figure 1F) to add 10 shots to the result area. By performing this action during interactive operation the user may check the automatically added results and remove irrelevant ones.

6 Tests and Results

The TRECVID evaluation consists of 24 topics (one of which had no relevant shots in the test set and was discounted). 15 minutes are allowed for answering each topic. Since answering all topics would take 6 hours, we assign subsets of topics to individual searchers. We employed 6 searchers (5 male; 1 female) to each answer 12 topics. All searchers have experience with video processing but most of them had not used the user interface before their 30-minute training session with a different news video collection. None of the searchers had seen the test collection or the topics before the search session. We grouped the topics into quarters and assigned them to the searchers in a standard latin square arrangement such that every searcher had a different combination of quarters. We then grouped searchers who had answered complementary sets of topics to create 3 groups of 2 searchers.

We evaluated search results by computing the average precision for each topic. This is the average of the precision values obtained after each relevant shot is retrieved. Relevant shots that are not retrieved are assumed to have a precision of 0. The mean over all topics (mean average precision; MAP) is used to compare results.

Figure 6 shows the mean average precision results for the three groups of searchers. In addition to the results for the user-selected shots, the figure also shows the results of bracketing shots and the three post-processing strategies described in the previous section. The post-processing strategies have similar performance (*WEIGHTED* is best and *LSA2* worst) and increase the MAP by 0.054 on average. While there are significant differences in performance between the groups of searchers, those differences are fairly small compared to the overall range of submitted results.

Figure 7 shows the MAP performance of our system with different post-processing strategies compared to all TRECVID submissions. Our best submission placed 3rd overall and only 4 submissions from 3 groups performed better than our worst performing submission [19]. Those 3 groups (University of Amsterdam/MediaMill, CMU, and IBM) have very mature image retrieval efforts and employ very sophisticated semantic image processing and feature detection. For example, the top-scoring MediaMill system uses a semantic lexicon with 32 concepts such as aircraft, bicycle, or Bill Clinton. This allows them to do well in TRECVID 2004 topics such as *"find shots of one or more bicycles rolling along."*

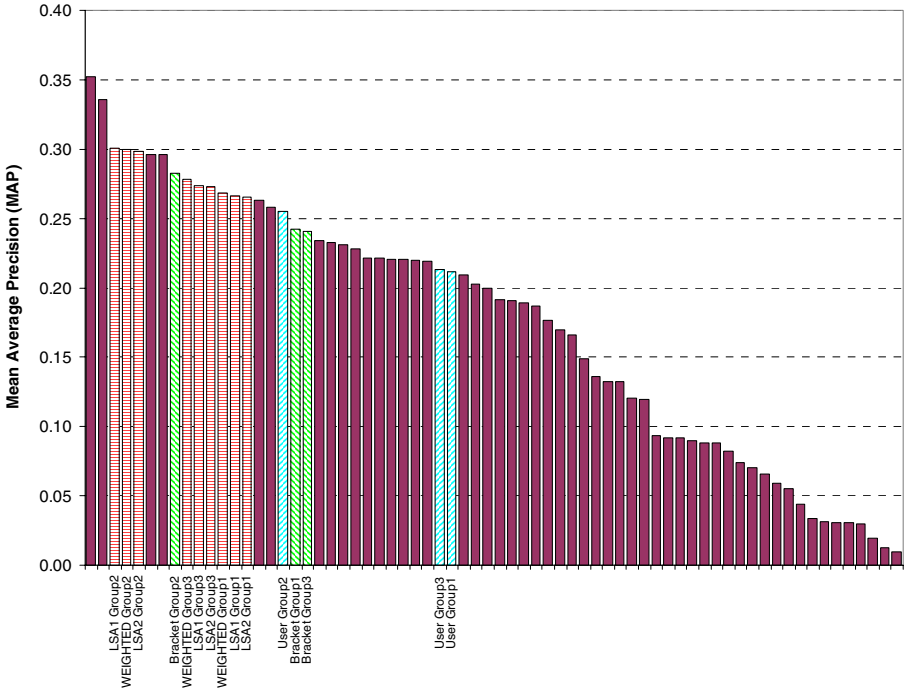

Fig. 7. Interactive search MAP scores for the entire set of TRECVID submissions. Scores or our 3 user groups with and without automatic post-processing are shown as striped bars. Other TRECVID participants' submissions as solid bars.

7 Conclusions

We presented an approach to supporting users in searching video collections. Our novel contributions fall into two areas. First, we use visualization techniques to draw the users' attention to promising results and support them in selecting relevant results. Second, we process the unstructured text associated with the videos and segment it into stories. We also determine keywords to present to the user. This is a difficult problem with latent semantic search.

Rather than using elaborate media analysis techniques, we provided an efficient user interface that enables users to quickly browse retrieved video shots and to decide which of those are truly relevant. Several visualization techniques were used to cue users to likely candidates for relevant video passages. We grouped video shots automatically into stories and represented those stories as keyframe collages where the more relevant keyframes were allotted more space. Redundant coding of size, position, color, and brightness were used to indicate document relevance to the users. We also marked already-visited stories across multiple searches to enable users to determine at a glance which results still had to be explored. This was especially important because the appearance of stories changed for different queries.

These features enabled the TRECVID participants in the interactive search evaluation to find many of the relevant video shots within the allotted time. Our evaluation results were very competitive with systems employing more sophisticated analysis techniques. We are currently looking beyond the TRECVID evaluation to determine how our system can be best adapted to real-world usage scenarios and plan to incorporate our current design into a larger video reuse system.

References

1. M.W. Berry, S.T. Dumais, G.W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, *SIAM Review*, 37(4), p. 573-595, 1995.
2. J.S. Boreczky and L.A. Rowe. Comparison of video shot boundary detection techniques. *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996.
3. P. Chiu, A. Girgensohn, and Q. Liu. Stained-Glass Visualization for Highly Condensed Video Summaries. *Proc. IEEE Intl. Conf. on Multimedia and Expo*, 2004.
4. M. Christel and N. Moraveji. Finding the Right Shots: Assessing Usability and Performance of a Digital Video Library Interface. *Proc. ACM Multimedia*, pp. 732-739, 2004.
5. M. Christel, J. Yang, R. Yan, A. Hauptmann. Carnegie Mellon University Search. TREC Video Retrieval Evaluation Online Proceedings, 2004.
6. E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G.J.F. Jones, H. Le Borgue, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N.E. O'Connor, N. O'Hare, S. Rothwell, A.F. Smeaton, P. Wilkins. TRECVID 2004 Experiments in Dublin City University, TREC Video Retrieval Evaluation Online Proceedings, 2004.
7. M. Cooper and J. Foote. Scene Boundary Detection Via Video Self-Similarity Analysis. *Proc. IEEE Intl. Conf. on Image Processing*, pp. 378-381, 2001.
8. A. Girgensohn, J. Adcock, M. Cooper, and L. Wilcox. Interactive Search in Large Video Collections. *CHI 2005 Extended Abstracts,* ACM Press, pp. 1395-1398, 2005.
9. Google Video Search. http://video.google.com
10. A.G. Hauptmann and M.G. Christel. Successful Approaches in the TREC Video Retrieval Evaluations. *Proc. ACM Multimedia*, pp. 668-675, 2004.
11. D. Heesch, P. Howarth, J. Megalhaes, A. May, M. Pickering, A. Yavlinsky, S. Ruger. Video Retrieval Using Search and Browsing. TREC Video Retrieval Evaluation Online Proceedings, 2004.
12. J. Huang, S.R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. *Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.*, pp. 762-768, 1997.
13. Jakarta Lucene. http://jakarta.apache.org/lucene/
14. M.F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3), pp. 130-137, 1980.
15. G.M. Quénot, D. Moraru, and L. Besacier. CLIPS at TRECvid: Shot Boundary Detection and Feature Detection. TREC Video Retrieval Evaluation Online Proceedings, 2003.
16. G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. *ACM SIGIR conference on R&D in Information Retrieval*, pp. 49-58, 1993.
17. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), pp. 513-523, 1988.
18. C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, F.J. Seinstra. The MediaMill TRECVID 2004 Semantic Video Search Engine. TREC Video Retrieval Evaluation Online Proceedings, 2004
19. TRECVID. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html
20. Yahoo! Video Search. http://video.search.yahoo.com