Interactive Video Search Using Multilevel Indexing

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

FX Palo Alto Laboratory Inc., Palo Alto, CA 94304, USA

Abstract. Large video collections present a unique set of challenges to the search system designer. Text transcripts do not always provide an accurate index to the visual content, and the performance of visually based semantic extraction techniques is often inadequate for search tasks. The searcher must be relied upon to provide detailed judgment of the relevance of specific video segments. We describe a video search system that facilitates this user task by efficiently presenting search results in semantically meaningful units to simplify exploration of query results and query reformulation. We employ a story segmentation system and supporting user interface elements to effectively present query results at the story level. The system was tested in the 2004 TRECVID interactive search evaluations with very positive results.

1 Introduction

The infrastructure and technology for maintaining large digital video collections has reached a point where use and distribution of these assets over wide area networks is fairly commonplace. Witness the popularity of video sharing through BitTorrent [1]. However, search technology within such collections remains relatively primitive despite the increasing demand for improved access to these assets. Video management systems rest on the integration of two evolving technologies. First, video analysis and segmentation systems build content-based indices into the video data. Secondly, information retrieval systems and user interfaces are applied to allow searchers to identify content that satisfies some information need. The video information retrieval problem is the focus of a growing research community as well as the TRECVID evaluations [2].

Large video collections present unique challenges to the search system designer. Numerous existing video indexing and retrieval systems rely on text annotations of one form or another. For example, recently deployed Web-based video search systems build indices using explicit annotations such as program abstracts [3] and text from closed-captions [4], or implicit annotations such as filenames and nearby text in referencing documents [5]. More advanced systems take a multi-modal approach, integrating features such as text derived from optical character recognition (OCR), image similarity, and semantic feature extraction [6–9]. While text-based information retrieval technology is fairly mature, its effectiveness for video retrieval is limited. When the search need is satisfied

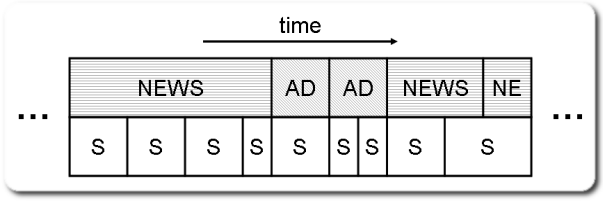

Fig. 1. The top row of the figure illustrates the boundaries between news stories and advertisements. The bottom row depicts the visual shot boundaries. Each story contains one or more shots. Story boundaries and shot boundaries need not align.

by a large unit of video, such as a program, text search can be very effective. But if the requirement is to identify the depiction of a specific event or item at the shot level, the precision of text based indexing generally falls short. Even when accurate transcripts are available, text-based indices only partially bridge the gap between a searcher's information requirement and the video media. This gap is most challenging when the information need is a complex visual category that is unlikely to be explicitly referenced in a text transcript, but exists even for the most explicit "*find person X*" queries since the presence (or absence) of referring text is not an accurate indicator of exactly where the person may appear (or not appear) [10]. This sort of semantic disconnect is exacerbated when the vocabulary used to refer to a search object varies in ways which may be unknown to the searcher, although this problem can be mitigated with techniques such as latent semantic analysis [11].

Given the limitations of automatic content-based indexing techniques, the burden is on the searcher to evaluate query results for accurate, fine-grained relevancy to the information need. The design goal of an interactive search system is to facilitate this task. More specifically, the interactive system must leverage content-based indexing to provide both a putative set of relevant content and also an interface by which the searcher can efficiently filter out the irrelevant results. Our search system is based on two key design choices. First, we use automatic analysis to organize the search results according to a topic-based story segmentation rather than a visual shot segmentation. Second, we have designed a user interface with dynamic visualizations of search results to enable the searcher to both quickly review a list of relevant story segments and peruse the shots within those relevant story segments.

In the next section, we describe media analysis components of our search system. The following two sections describe the system's interactive design elements and automated search capabilities. Section 4 describes the results of our experiments performed in conjunction with the TRECVID 2004 evaluations [2]. While the TRECVID broadcast news corpus is a strongly structured genre (the talking heads are called anchors for good reason), our segmentation and search applications use no genre-specific information.

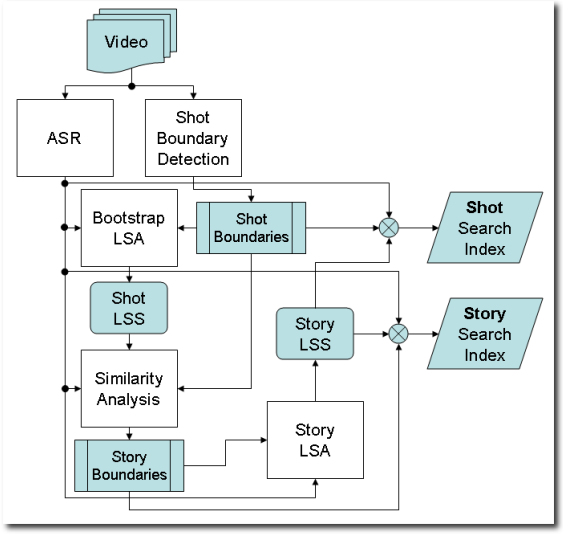

Fig. 2. Transcripts and shot boundaries are used to build a latent semantic space (LSS) which then provides the basis for a self-similarity-based segmentation of the transcripts to create story boundaries. The story boundaries in turn are used to segment the text transcript and build the LSS used for search.

2 Multilevel Indexing

Search results are presented in our interface at an intermediate level using a semantically-derived segmentation of the source material. In a wide variety of video genres, including news and documentaries, numerous shots may be contained within a single, semantically coherent, story. Figure 1 illustrates graphically the hierarchy of shot and story segments. The underlying "true" story boundaries need not align with shot boundaries. For instance, a news anchor can change topics without a visual transition. For simplicity we assume that they do in fact align and each shot is assigned uniquely to a single story. The story-level segmentation puts related shots into groups easily appreciated by the searcher. Additionally, since the story segmentation forms the basis of the latent semantic space (LSS) used for querying the database, there is a greater synergy between the search engine and the search results than through keyword matching search or image/shot-based search results. By providing story-based results, the searcher can more easily and productively browse and evaluate the query results returned by the system.

Figure 2 illustrates the process used to build the story-level segmentation and index. Preprocessing steps produce a time aligned text transcript and a shot-level segmentation. Shots are then merged into stories based on text similarity in the derived LSS.

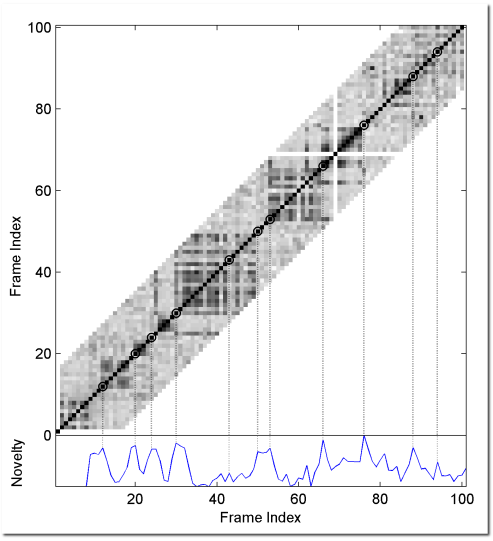

Fig. 3. A portion of the text-based self-similarity matrix for a video program. The (i,j) element in the matrix corresponds to the cosine similarity between the i^{th} and j^{th} shot-based text segments. Only elements near the diagonal are used in the analysis. Boundaries are chosen at points of maximum novelty shown below.

2.1 Preprocessing

The preprocessing of the videos consists of two components: shot boundary determination and text-extraction. Although the TRECVID test data for the evaluations we present includes a reference shot segmentation[12], we typically generate a shot segmentation with our own shot boundary determination (SBD) system based on a combination of similarity analysis and supervised nearest neighbor classification [13]. Our SBD system was evaluated at TRECVID 2003 and 2004 with very favorable results. Automatic speech recognition (ASR) transcripts are provided by LIMSI [14] as part of the TRECVID corpus. In the absence of these transcripts, closed captioning, manual transcription, or another source of ASR (or possibly OCR) would be required to generate text annotations.

2.2 Story Segmentation

We combined the text transcripts and visually-based shot segmentation to build a story segmentation using latent semantic analysis (LSA) [15, 16] and a similarity-based segmentation technique [17] as follows. We stopped and stemmed the text of the speech recognition transcripts [18], resulting in a dictionary of approximately 12000 terms. The shot-level segmentation was then used to compute document statistics and collect term vectors from which to build an LSS. The TRECVID 2004 corpus consists of 128 approximately half hour news programs, with an average of 260 shots per program in the reference shot segmentation.

Shots were joined until a minimal number (20) of text tokens was contained in each segment. This resulted in an average of 72 shot-based non-overlapping text segments per program, 9188 text segments total, in the bootstrap step. A truncated singular value decomposition (SVD) of order 100 was performed on this 12000-term by 9188-document matrix to generate the LSS used for story segmentation. A term vector was generated for every shot in the original shot segmentation and projected into this shot-based LSS. Thus each shot is represented by a vector of projection coefficients in the LSS.

For each term vector the cosine similarity with adjacent vectors was computed to generate a (partial) similarity matrix for each program. An example appears in Figure 3. The (i,j) element of the matrix is the cosine similarity between the projection coefficient vectors from the i^{th} and j^{th} shots in the video. From this matrix a novelty score was computed via kernel correlation [17], and local maxima in the novelty exceeding a preset threshold were chosen as story-boundaries. The novelty score computed from the example similarity matrix is shown in the bottom of Figure 3. If necessary, boundaries were added at lesser maxima in the novelty score until each resulting story contained less than a maximum number of shots (16). This process resulted in an average of 25 story segments per half hour program and 83 text tokens per story.

The story segments were then used to segment the transcripts and build a new LSS for use in search. The text from each story segment was treated as a unit for the computation of the story-level LSS. Corresponding document frequency statistics were recomputed and story term vectors generated resulting in a 12000-term by 3237-document term matrix from which to compute the new story-based LSS, also of order 100. During search operations query text was compared to story text or shot text by measuring the cosine similarity between term vectors in the story-based LSS. The number of documents and text tokens involved in this semantic analysis process is fairly small by the standards of the literature on latent semantic analysis [11], and the resulting semantic groupings are predictably noisy. Despite this apparent drawback, the semantic index smooths across vocabulary use and co-occurrence in ways that are intricately tied to the content of the corpus from which it is derived. With appropriate feedback (see Figure 5) the searcher may gain insight into fruitful topics to explore. Additionally, the smoothing can be expected to mitigate the impact of errors in ASR transcripts. We also generated a keyword matching index at both shot and story levels to be used during search at the discretion of the user. In practice this option was rarely used, and only after exploring the vocabulary space with the LSS-based search.

3 Search Support in the User Interface

Figure 4 shows the interactive search interface. In the user interface, stories are the main organizational units for presenting query results to the user. The user enters a query as keywords and/or images in region A. Once the user has entered a query and pressed the search button, story results are displayed in order of

Fig. 4. The search interface. A text and image query is specified in area A. Query results are summarized as story keyframe collages in area B. Selecting a story expands the shots program timeline in area C. Relevant shots are collected in area D. the TRECVID topic is displayed in area E.

decreasing relevance in region B. The collages in the search results area are also sized proportional to their relevance. When the user wants to explore a retrieved story, he clicks a collage. The corresponding video program is opened in region C and the selected story is highlighted in the video timeline. Below the timeline the keyframes from the shots in the selected story are expanded. The program timeline is color coded by story relevance and segmented by story. Below the timeline a click-able sequence of story collages is displayed. This is also pictured in Figure 6. This enables the user to explore the neighborhood of a search result in a structured fashion. When the user finds a shot of interest, he drags it to the result area in region D.

3.1 Representing Shots and Stories

In the user interface, stories are the main organizational units for presenting query results to the user. While the frames comprising a video shot are visually coherent, and can generally be well represented with a single keyframe, stories consist of multiple shots and are unlikely to be satisfactorily represented by a single keyframe. Therefore, we represent stories as collages of relevant shot keyframes as in Figure 5. This provides an easy way for the user to visually judge whether a story is worth exploring. To build a story collage we select the shots with the highest relevance scores and use their keyframes. We determine a retrieval score for each shot using the same story-based LSS. We crop the keyframes instead of scaling them down in an attempt to keep the content rec-

Fig. 5. Tooltips showing relevant keywords for the top two results pictured in Figure 4 for the query terms: "hockey","ice". Words shown in bold are those closest to the query terms. Words in plain font are those with the highest tf*idf values. Note that the obvious query term, "hockey", does not appear.

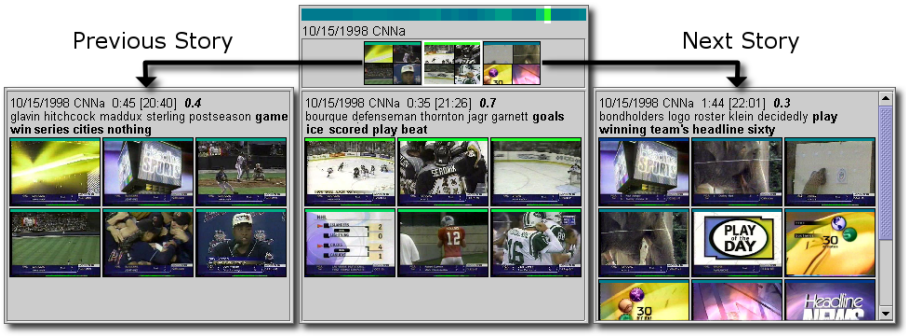

Fig. 6. Expanded shots for the top story in Figure 4 and the stories immediately preceding and following. By visual inspection of the keyframes and keywords it is easy to see that the preceding and following stories are not related to the search topic, hockey.

ognizable at reduced scales. Each keyframe is assigned an area in the collage proportional to its relevance score.

3.2 Keyword Relevance Cues

As depicted in Figures 5 and 6, tooltips for story collage and video shot keyframes provide information about both the most distinctive words in the document and the words most relevant to the current query. These keywords provide guidance to the user for re-formulating search terms. The words with the highest term frequency * inverse document frequency (tf*idf) [19] values are used as the most distinctive keywords for a shot or story. In Figures 5 and 6 these are rendered in normal strength font. The terms shown in bold are the story terms most closely related to the query and indicate why the document is deemed relevant by the search system. When using a keyword matching text search, the terms most related to the query are the subset of the query terms that appear in

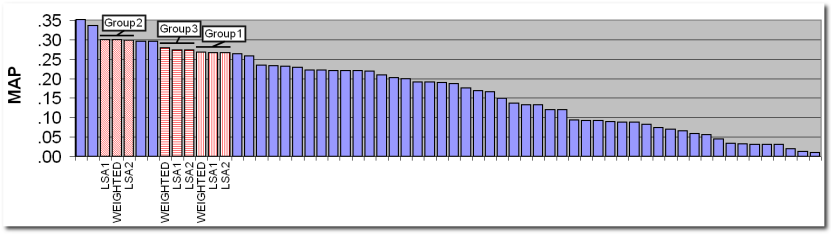

Fig. 7. Mean average precision (MAP) for all TRECVID 2004 interactive search submissions. We submitted 3 runs each, one for each post-processing method, from 3 different user groups for 9 total runs. Our 3 groups placed 3rd-5th, 8th-10th, and 11th-13th in overall MAP.

the story. When LSA-based text search is used, a relevant document may not contain any of the query terms. We use the LSS to identify terms in the document that are closest to the query by first projecting the query term vector into the LSS, then re-expanding the reduced-dimension vector back to a full length term vector. The document terms are then ranked by their corresponding values in this smoothed query term vector. For the example illustrated in Figure 5 the query text was "hockey ice". In this example the most discriminative terms include proper names such as "bourque", "brind'amour", and "portugal", while the boldface terms closest to the query are more general and reflect the global nature of the LSS: "goal", "score", "beat", "win".

4 Evaluations

We tested our search system as part of the TRECVID 2004 interactive search evaluation[20]. The TRECVID interactive search task is specified to allow the user 15 minutes on each of 24 topics to identify up to 1000 relevant shots per topic from among the 33,367 shots in the test corpus. It is unlikely that a user will be able to identify every relevant shot in the allowed time, or alternatively, identify 1000 shots with any degree of care. Meanwhile, the primary global performance metric, mean average precision (MAP), does not reward returning a short high-precision list over returning the same list supplemented with random choices. Thus, the search system is well advised to return candidate shots above and beyond what the user identifies explicitly.

We apply 3 different methods to fill out the remaining slots in the TRECVID shot result list. A first step in all cases is a heuristically motivated one where the shot immediately preceding and immediately following each shot in the user-identified relevant list is added. The rationale behind this step is that relevant shots often occur in clumps [21], and that a searcher may simply miss some. The nature of the reference shot segmentation [12] contributed to this phenomena during evaluation. Because shots were limited to a minimum length of 2 seconds, many shots contained subshots whose content was not reflected in the main shot keyframe. Other SBD systems are subject to similar problems when the shots

contain significant visual change; static keyframing can not anticipate future user needs. In these cases the searcher needs to play the video clip to accurately evaluate the shot contents. The bracketing step accounts for roughly half of the improvement in MAP that we garner from the post-processing. After bracketing, three variations of automated query are used to fill the remaining slots:

Weighted Query History (WEIGHTED) Each query from the interactive session is re-issued and its precision measured against the list of relevant shots. Each shot is given the precision-weighted sum of the individual query relevance values.

Single LSA Query (LSA1) In this mode the text from the shots that have been judged by the searcher to be relevant is combined to form a single LSA-based text query. This query is applied to the unjudged shots and the highest scoring ones retained for the result list.

Multiple LSA Query (LSA2) The text for each shot in the relevant list is used to form an individual LSA-based text query. The relevancy values for each query are then combined based on the precision against the relevant list as in the WEIGHTED method.

The mean average precision (MAP), for all participants is shown in Figure 7. We employed 6 searchers to collectively complete each search topic 3 times. As described above, 3 variations of automated search post-processing step were applied to each topic result, yielding the 9 submissions graphed in Figure 7. The difference in performance between automation types was quite low, as was difference in performance between users. The MAP performance of our system was very competitive with systems using very rich video features as part of their search criteria [6–9]. As shown in Figure 7, 2 other submissions outperformed our best submission, and 4 outperformed our worst submission. While the performance of a search system of this nature is subject to a great many confounding factors, we attribute the success of our effort to the combination of the story-based search and powerful interface elements.

5 Summary

We have described the design and implementation of an interactive video search system aimed at leveraging the power of automated analysis techniques to facilitate the application of human judgment of query results in an interactive system. A story-based segmentation of the source material, coupled with interface methods that succinctly summarize the results and their relevancy to the query, form the foundation of the system. In evaluation we found our system to be very competitive with other systems employing more advanced content-based video analysis.

As we move forward we plan to incorporate more content-based indexing methods into our search system, but will continue our emphasis on applying these methods in ways that leverage the unique discriminative abilities of the searcher. In the near term, highly accurate recovery of complex semantic information from

video may be impractical. Nevertheless, the opportunistic use of ever-improving content analysis within inspired user interfaces holds a great deal of potential for creating powerful access to a wide variety of video media.

References

1. Fonda, D.: Downloading hollywood. Time Magazine **165** (2005)
2. Kraaij, W., Smeaton, A.F., Over, P., Arlandis, J.: TRECVID 2004 – an introduction (2004) http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/tv4intro.pdf.
3. Internet Archive: Moving images archive (1996) http://www.archive.org/movies.
4. Google: Google Video Search (2005) http://video.google.com.
5. Yahoo: Yahoo! Video Search (2005) http://video.search.yahoo.com.
6. Snoek, C., Worring, M., Geusebroek, J., Koelma, D., Seinstra, F.: The MediaMill TRECVID 2004 semantic video search engine. In: TREC Video Retrieval Evaluation Online Proceedings. (2004)
7. Heesch, D., Howarth, P., Megalhaes, J., May, A., Pickering, M., Yavlinsky, A., Ruger, S.: Video retrieval using search and browsing. In: TREC Video Retrieval Evaluation Online Proceedings. (2004)
8. Christel, M., Yang, J., Yan, R., Hauptmann, A.: Carnegie mellon university search. In: TREC Video Retrieval Evaluation Online Proceedings. (2004)
9. Cooke, E., Ferguson, P., Gaughan, G., Gurrin, C., Jones, G., Borgue, H.L., Lee, H., Marlow, S., McDonald, K., McHugh, M., Murphy, N., O'Connor, N., O'Hare, N., Rothwell, S., Smeaton, A., Wilkins, P.: TRECVID 2004 experiments in dublin city university. In: TREC Video Retrieval Evaluation Online Proceedings. (2004)
10. Yang, J., yu Chen, M., Hauptmann, A.: Finding person X: Correlating names with visual appearances. In et al, E., ed.: International Conference on Image and Video Retrieval, Springer (2004) 270–278
11. Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, vector spaces, and information retrieval. SIAM Rev. **41** (1999) 335–362
12. Ruiloba, R., Joly, P., Marchand-Maillet, S., Quénot, G.: Towards a standard protocol for the evaluation of video-to-shots segmentation algorithms. In: European Workshop on Content Based Multimedia Indexing, Toulouse, France. (1999) 41–48
13. Cooper, M.: Video segmentation combining similarity analysis and classification. In: MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, ACM Press (2004) 252–255
14. Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI broadcast news transcription system. Speech Commun. **37** (2002) 89–108
15. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. SIAM Rev. **37** (1995) 573–595
16. Choi, F.Y.Y., Weimer-Hastings, P., Moore, J.: Latent semantic analysis for text segmentation. In: 6th Conference on Empirical Methods in Natural Language Processing. (2001) 109–117
17. Cooper, M., Foote, J.: Scene boundary detection via video self-similarity analysis. In: IEEE Intl. Conf. on Image Processing. (2001) 378–381
18. Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–130
19. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press (1999)
20. TRECVID: TREC video retrieval evaluation. Workshop (2001, 2002, 2003, 2004) http://www-nlpir.nist.gov/projects/trecvid/.
21. Pirolli, P., Card, S.: Information Foraging. Psychological Review (1999)