Video Searching and Browsing Using ViewFinder: Participation and Assessment in TRECVID-2003

Dan Albertson

Laboratory for Applied Informatics Research, Indiana University, Bloomington
1320 E. 10th St, LI011, Bloomington, IN 47405. Email: daalbert@indiana.edu

Javed Mostafa

Laboratory for Applied Informatics Research, Indiana University, Bloomington
1320 E. 10th St, LI011-D, Bloomington, IN 47405. Email: jm@indiana.edu

John Fieber

Laboratory for Applied Informatics Research, Indiana University, Bloomington
1320 E. 10th St. LI 011, Bloomington, IN 47405. Email: jfieber@indiana.edu

This research project explores the topic of video information retrieval in conjunction with the task definitions and data provided by the Text REtrieval Conference's (TREC) 2003 Video Workshop (TRECVID-2003). Included in this paper, we discuss our processes and various phases in participating with TRECVID-2003. Specific sections discussed include database development, data indexing and retrieval approaches, development of user-interface and client side features, interactive search experiments, results, and conclusions.

Introduction

Everyday more and more video is being digitized and made available through various information systems and/or the World Wide Web. As a result, demands for video resources have increased significantly and such querying is becoming more prevalent in everyday information seeking [4]. Spink et al. (2001) observed this by examining Excite query logs over the span of 3 years and found that searches for video content actually doubled. For these reasons, along with other similar findings, there suggests a growing importance in the exploration of problems and questions surrounding video retrieval; thus, there is a need for members of the research community to collaborate and learn from one another through professional and academic forums (such as TREC).

In order to participate in the 2003 Text REtrieval Conference's (TREC) video workshop (TRECVID-2003), several researchers from the Laboratory for Applied Informatics Research (LAIR) at Indiana University, Bloomington developed a video retrieval system named ViewFinder. This is the second consecutive year that results from ViewFinder have been entered into TRECVID; however, numerous modifications had to be made from the 2002 system in order to conform to some of the new participation requirements. Factors that significantly contributed to this year's system and experimental adjustments include an entirely new video and image dataset, automatic speech recognition (ASR) and closed-caption (CC) outputs (provided by the workshop), and stricter task definitions.

The problem at hand attempts to explore query modeling and user-interfaces (of video retrieval systems) by enabling users to search and browse through the assigned TRECVID data. There were several major components that we fulfilled for the purpose of exploring this problem and finalizing the TRECVID-2003 experiments.

First, we concentrated on indexing the ASR data, and applied an appropriate weight for each keyword. This data would be utilized in a keyword search feature, in which users can formulate queries (consisting of select terms) of their choosing.

Next, we gave users the option to browse video data without having to perform a formal keyword search. As a result, the system offers several major headings in which the user can browse the associated keyframes. Moreover, ViewFinder's interface displays the keyframes designated for each video shot, thus allowing the user to browse by visual clues as opposed to text-based (although further textual information is available upon request).

In regards to the user experiments, we participated and submitted results for 1 run which fulfilled the interactive task as defined by the workshop.

Further details of the above information (including system development and search experiments) along with discussion of the experimental results and conclusion will be covered in the following sections.

Methods

Building upon previous research and experiences, we employed certain methods which we believe to be suitable for participating in TRECVID-2003. This section will cover specific aspects of our methodology including system development (database and client side) along with experimental design.

Data and Keyword Indexing

Considering an entirely new data set was issued for this year's TRECVID, the tasks of creating a database schema and keyword indexing had to be performed. The visual data provided by TRECVID-2003 included video and image data. More specifically, TRECVID issued around 133 hours of video data, which derived from CNN Headline News, ABC World News Tonight, and CSPAN (only around 13 hours worth of CSPAN). This resulted in approximately 125 thousand keyframes (images) to represent all individual shots. All CNN and ABC video was originally broadcast during the span of January 1998 to June 1998, while CSPAN video ranges from 1998 to 2001.

Accompanying the visual data, TRECVID also provided an assortment of textual information. One such example of this data corresponds to the collection of video files as a whole (i.e. information regarding individual files). This data was issued in XML format in which we extracted (using Java's XML API) and indexed (using JDBC) to form the "Video Table" (*see Table 1 for database schema and corresponding attributes of Video Table*).

Table 1: Database Schema of ViewFinder

Table Name	Attributes
Video Table	video_id, video_url, video_use, video_source, video_date, num_of_shots
Shot Table	video_id, video_filename, video_start_time, video_duration, shot_id, shot_start_time, shot_duration, image_url, time_of_shot
Keyword Table	video_id, shot_id, keyword, weight, freq_per_shot, freq_per_video, freq_per_dataset
Unique Terms Table	video_id, keyword, num_of_shots, idf

Next, we made use of textual data which comprised the common shot boundary directory (also issued by TRECVID). This data contained a separate XML file for each video and includes textual information corresponding to each shot. Considering the format of this data (XML), we parsed and indexed it in a similar fashion to the video collection data. The resulting data from this process can also be found in Table 1, and categorized under "Shot Table".

The last set of textual data that was indexed includes the automatic speech recognition (ASR) output. This data had a different format than what was previously mentioned (i.e. not XML), so different techniques had to be used to extract the keywords. This procedure included simple string comparison and modification techniques as offered through various Java APIs.

Embedded within the ASR data (along with the terms) were timestamps that indicated when the keywords were spoken and the duration of each. Moreover, other tags indicated a timestamp for a certain block of keywords (i.e. for a "statement. We would use this timestamp for keyword indexing and shot association purposes. More specific information regarding this process is discussed below.

The ASR output was utilized using two different approaches, and indexed accordingly. By extracting all the lines from the ASR output and comparing the timestamps (of the ASR files) with timestamps within the shot boundary directories, we were capable indexing all the keywords and associating each with a corresponding shot and video ID. This process resulted in the formation of our "Keyword Table" (*see Table 1 and Keyword Table*). *Note that certain timing (compliance) calculations had to be performed in order to make the two timing formats comparable.*

As just mentioned, in the "Keyword Table" all terms were extracted, indexed, and assigned a video and shot ID. In the case that the same keyword appeared in the same shot (of the same video) the redundant use of the keyword was disregarded, but a (keyword) frequency per shot integer was incremented and indexed accordingly. Moreover, redundant keywords in a video file were still indexed; however, they are distinguished by different shot IDs and weights (which is discussed below).

Next, the ASR data was used to form a table of unique terms (*see Table 1 and "Unique Terms" table*). Here, each unique term was indexed per video. In this instance, if the same term appears multiple times in the same video, instead of re-indexing it, the number of shots the keyword appeared in was tracked and indexed along with the keyword.

After populating the "Keyword" and "Unique Term" tables we were capable of applying certain weights to each keyword. First, an *idf* weight was given to each term located in the "Unique Terms" table. The calculation used to formulate the *idf* weight is seen directly below in *Equation 1.*

$$idf = \log_2(N/n)$$

N = total number of shots in a video file

n = total number of shots in which the term appears

Equation 1: idf Used in ViewFinder

Once the *idf* value for each unique term was stored, an overall *tf·idf* weight was then calculated and assigned to each keyword (appearing in the "Keyword Table"). This weight consists of the product of the *idf* calculation mentioned above and the term frequency per shot (previously stored in the "Keyword Table").

User-interface and Client Side Features

The graphical features and user-interface of ViewFinder were constructed and operate using Java's Swing API. The interface itself is made up of two primary panels, which include a results display panel and a searching features (querying) panel (*See Appendix A for snapshot of ViewFinder interface*).

The results panel takes up approximately the left half of the interface, and has several functions associated with it. First, it is used to display keyframes of individual shots returned after the user has queried the system; thus, allowing the user to visually browse the search results. The results panel can display up to 8 keyframes (results) at a time, with results being ranked from most relevant (upper-left corner) to least relevant (bottom-right). (The displayed keyframes were generated from the images issued by TRECVID and were reduced to approximately ¼ their original size (i.e. thumbnails) for display purposes).

The results panel also offers the user several other features including the option to view further textual information regarding a specific keyframe (shot), and the option to expand upon a previous search. These options are presented to the user in a series of drop down menus located directly below the 8 displayed keyframes (where each menu corresponds to the keyframe located directly above it).

The options included within the menus are "Details" and "Promote". By selecting "Details" the system will be prompted to retrieve textual data such as video source, video date, video ID, shot ID, and a larger sized image of the keyframe (i.e. the shot details) and display the information in a separate window.

On the other hand, "Promote" will retrieve the keywords associated with that particular shot (which exceeds a certain *tf·idf* weighting threshold), compare them to all the other shots in the database, and then return shots which have matching keywords. Moreover, the system will perform a Boolean 'OR' search therefore shots which contain any of the promoted keywords will be returned. In addition, shots which have 2 or more matching keywords have the corresponding *tf·idf* values combined resulting in an overall boost in relevancy weighting. All returned shots are then sorted and returned according to relevance. Once a "Promote" search has been performed, the keyframe which has been promoted is transferred to the middle image position (within the results panel) for visual reference for the user.

The search panel (appearing on the right-hand side of the ViewFinder interface) offers several ways in which users can formulate queries and browse the video data. For searching, a text box where terms can be entered and compared with the keywords indexed (in the "Keyword Table") is available. Similar to the "Promote" search feature mentioned above, if there are 2 or more search terms in which to compare, the system will perform an 'OR' search; thus, returning all shots that contain any of the entered keywords. In addition, the same procedure applies when multiple keywords match for an individual shot (i.e. term weights are added together as mentioned above). *Note that considering all ASR keywords contain only capital letters, the keyword search feature is not case sensitive as all queried terms are modified for comparison purposes*.

Aside from the keyword searching function, the system also allows for certain types of video browsing. The browsing options are presented in a drop down menu appearing at the top of searching panel (top right of the ViewFinder interface). By clicking on the menu, the users can choose from video date, video source, and date + source in which to browse. After selecting one of the options, a series of choices are then retrieved and returned to the user and presented in the list box located directly below the drop down menu (*See Appendix A for snapshot of ViewFinder interface*). The user can then select one choice and hit search, which will retrieve the results and display the corresponding keyframes in the results panel.

Other features of the search panel include the "More" button which becomes available in the case that more than 8 shots are returned after a search; thus, allowing the user to browse all returned shot if necessary, and the "Back" button where the user can re-examine previously viewed search results. Also, a feedback field, which will display the last performed query and the number of results returned is located in this panel.

Search Experiment Design

Our experimental designed consisted of performing 1 interactive search run. This complied with the mandatory run detailed in the participation requirements, which was to only include experimental results regarding the ASR output. For classification purposes, ViewFinder was categorized as a 'C' system, as it was trained according to the methodology mentioned above, and didn't meet the criteria of a category 'A' or 'B' system as described in the requirements.

All 24 search topics (which was designated for the interactive task) was completed in a sequential order. We employed 1 search subject which completed all the topics over 2 testing sessions. Furthermore, this was treated as a simulated search experiment as the primary system designer completed each topic. There was a maximum of 15 minutes in which to complete each searching topic. The overall average time for each topic resulted in 10.4 minutes per topic.

Considering that ViewFinder doesn't have any content-based searching capabilities, no such runs (as detailed by TRECVID) using visual data could be performed.

Results

This section discusses the results from the submitted run as described above and compares those with other systems participating in the search task. The measurements of mean averaged precision (MAP), interpolated recall precision, and precision at *n* shots were performed by assessors at the National Institute for Standards and Technology (NIST), and can be further explored in the proceedings of TREC-10 [5]. Moreover, further results analysis was performed and includes system ranking (of ViewFinder) across each search topic.

Out of the 24 search topics designated for the interactive task, there was a total of 2067 relevant shots identified by TRECVID, in which ViewFinder (after completing all 24 search topics) ended up retrieving 282 (13.6%) of them. This came out to an average of 11.75 relevant shots per topic where a range of 58 (max) to 0 (min) was observed.

Our results can also be reflected by the mean averaged precision measured at 0.030 and by the mean precision at the total of relevant shots at 0.051.

The mean precision for each search topic had a range of 0.169 (0.169 to 0.000). This (MAP) is compared to 0.135, which was the mean average precision across all submitted runs.

Other results issued by TRECVID include the interpolated recall precision and the level of precision at *n* shots. The results of these two measurements are summed up in the following table (*Table 2: Summary of Interactive Search Results*).

Table 2: Summary of Interactive Search Results

Interpolated Recall Precision		Precision at *n* Shots	
0.0	0.5835	5	0.2250
0.1	0.0816	10	0.1333
0.2	0.0473	15	.01028
0.3	0.0047	20	0.0896
0.4	0.0006	100	0.0446
0.8	0.0006	500	0.0163
1.0	0.0006	1000	0.0118

Finally, ViewFinder's performance can be measured in terms of ranking across the 24 search topics. For each, there was an average of 74 submitted runs where ViewFinder's average ranking was 47.68. Furthermore, the best finish was 18th while the worst ranking was a T-68th.

Conclusions and Future Improvements

For this year's TRECVID experiments, ViewFinder only made use of textual data and by analyzing the results we can draw several conclusions regarding this approach. From first glance, our *tf·idf* weighting seems to be somewhat pertinent considering the number of relevant shots returned (*See Results section above*). However, we realize that various adjustments need to be made for best application of this formula.

Although we were somewhat pleased with the percentage of relevant shots returned by ViewFinder, our mean average precision obviously suffered. As a result, we are beginning to explore how to better limit the search results (i.e. attempt to only include relevant shots).

One such possibility includes incorporating a stop word list, which wasn't used for this year's ViewFinder. This could reduce the number of returned shots by disregarding the use of widely used terms (the, and, on, etc.). Specific characteristics of such a stop word list (one for the purpose of video retrieval) have yet to be discussed.

Next, we would like to incorporate additional Boolean options, instead of limiting the search feature to only include 'OR'. Here, users would be capable of further refining search results by utilizing other operators such as 'AND' and 'NOT'.

Finally, we would like to make the search and browse functions of ViewFinder cross compatible. Moreover, currently with ViewFinder each searching feature operates independently from one another (i.e. the keyword search will take precedent over the browse features if search terms have been entered into the keyword field). Also, there is no way to search within a set of results (i.e. once a browse function has been performed), which would be useful in limiting the number of irrelevant shots.

As for contextual based searching, our initial goal for TRECVID-2003 was to also submit a run based solely on these features (i.e. image analysis). However, due to time constraints, we were unable to complete the image processing and database population tasks.

We still plan on exploring video retrieval in this fashion, and have done some preliminary experiments using Java's Advanced Imaging (JAI) API. With JAI, we were capable of extracting color information from certain keyframes (which was taken from TRECVID-2002 videos) and incorporating a search by "Histogram" function into a prototype of ViewFinder. The preliminary results were somewhat satisfying, but we feel that additional content-based search features need to be included (along with color histogram) to make for a practical search function. Such other content-based features may include searching by edge, shape, and object detection. By analyzing this year's search topics, we feel that a content-based search feature is necessary to participate in future TRECVIDs, which our goal is to have such a prototype completed and functioning by 2004.

References

[1] Jose, J. M., Furner, J., & Harper, D. J. (1998). Spatial querying for image retrieval: a user-oriented evaluation. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia,* 232 – 240.

[2] Rodden, K., Basalaj, W., Sinclair, D., & Wood, K. (2001). Does organization by similarity assist image browsing? *Proceedings of the SIGCHI Conference on Human factors in Computing Systems, Seattle, WA,* 190 – 197.

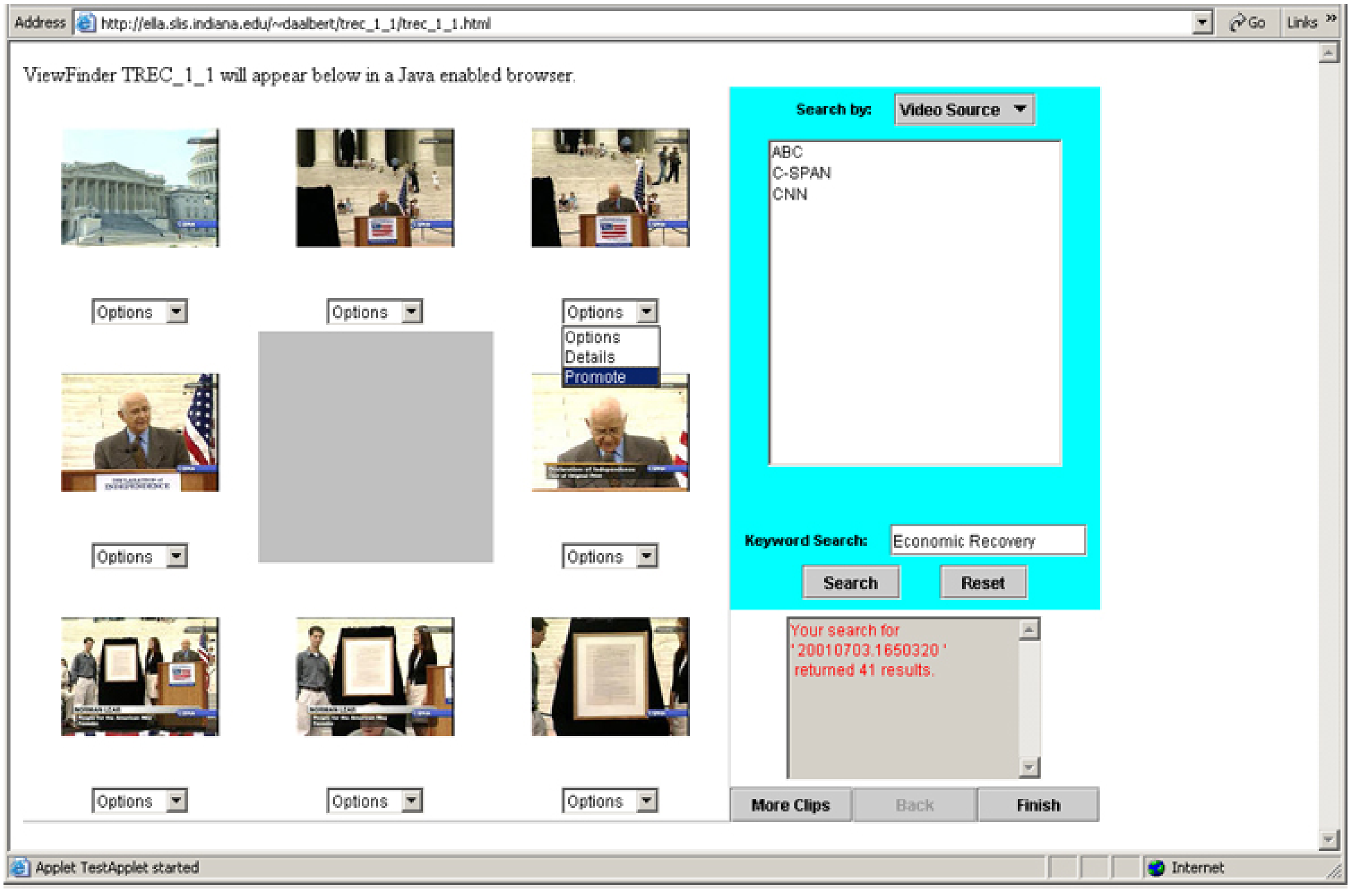
[3] Santini, S., & Ramesh, J. (2000). Integrated browsing and querying for image databases. *IEEE Multimedia, 7*(3), 26 – 39.

[4] Spink, A., Goodrum, A., & Hurson, A. R. (2001). Multimedia we queries: Implications for design. *Proceedings of the International Conference of Information Technology: Coding and Computing, Las Vegas*, NV, 589 – 593.

[5] Vorhees, E. M., & Harman, D. K. (Eds.). Common Evaluation Measures. (2001). *NIST Special Publication 500-250: The Tenth Text Retrieval Conference,Gaithersburg, MD*, A14 – A23.

[6] Zhou, X. S., & Huang, T. S. (2002). Unifying keywords and visual contents in image retrieval. *IEEE Multimedia, 9*(2), 23 - 33.

Appendix A: Snapshot of ViewFinder user-interface.



Address http://ella.slis.indiana.edu/~daalbert/trec_1_1/trec_1_1.html

ViewFinder TREC_1_1 will appear below in a Java enabled browser.

Search by: Video Source

Keyword Search: Economic Recovery

Your search for '20010703.1650320' returned 41 results.

Applet TestApplet started