VIDEO SEARCHING AND BROWSING USING VIEWFINDER

By

Dan E. Albertson	**Dr. Javed Mostafa**	**John Fieber**
Ph. D. Student	**Associate Professor**	**Ph. D. Candidate**
Information Science	**Information Science**	**Information Science**

School of Library and Information Science
Indiana University

Abstract

Several researchers consisting of students and faculty from the School of Library and Information Science at Indiana University developed a video retrieval system named ViewFinder for the purpose of providing access to video content for a project named the Cultural digital Library Indexing Our Heritage (CLIOH) at Indiana University Purdue University at Indianapolis (IUPUI). For our role in the Text Retrieval Conference (TREC) and its video track, we took the existing system, made notable modifications, and applied it to the video data provided by the conference. After conducting 1 interactive search run, we generated our search results and submitted them to TREC where human judges determined the relevancy of each returned shot and assigned an averaged precision ranking for each topic. From these results we were capable of drawing conclusions of the current system, and how to make ViewFinder more productive in future versions.

Introduction

With the accumulation of digitized video, groups and individuals are becoming more and more interested in the preservation and organization of such content. Along with this preservation and organization, there is a need for systems that can provide easy and efficient access to archived video. This problem is the focus of ViewFinder, a video retrieval information system.

The main goal of Viewfinder is to have it applied to a project being conducted at Indiana University Purdue University at Indianapolis (IUPUI) named the Cultural digital Library Indexing Our Heritage (CLIOH). This project deals with the preservation of multi-media content regarding the ancient world (Mayan ruins, etc.). One such form (of content) is video, and this is the current focus of ViewFinder. ViewFinder attempts to provide users with individual keyframes (of *shots* located within *video files*) according to the user's information need.

For the purpose of participating in the Text Retrieval Conference (TREC) and its video track, we took the existing system, made notable modifications, and applied it to the video data provided by the conference. We then followed conference procedures and performed 1 interactive ("human in the loop") search run consisting of 25 individual topics (also provided by the conference). We then generated results and submitted them to TREC, where human assessors compared our results with the number of manually identified relevant shots, and assigned an average precision for each topic. You may further explore the average precision formula used by TREC in Vorhees, E. M., and Harman, D. K. (2001). In addition to conducting an interactive search run, our system was developed with use and knowledge of the actual search test collection, known as type-A.

Related Literature Review

In recent years there have been various advances in regards to this research problem. This is possibly due to the large increase of multimedia content (especially video) being digitized and made accessible via the World Wide Web and other multimedia information systems.

Along with this increase in video content, there is an increase in people who choose to search for such content. Spink, Goodrum, and Hurson (2001) concluded from a study on Excite query logs between the years 1997 and 1999 that queries for video content increased over 100%. In fact, video queries counted for 0.7% of overall queries in 1997 and counted for 1.6% in 1999 (Spink et al., 2001). Spink et al. (2001) go on to further conclude "video searching became more frequent during this period with the expansion of video material on the Web."

These findings suggest that it is very important for IR researchers to explore how to better provide easier and more efficient access to video content. Cruz and James (1999) provide insight into this problem by focusing on aspects such as user query generation coupled with the user-interface design. They go on to detail their system named Delaunay (Cruz and James, 1999). They express the importance of users having the capability for "pre- and post-query refinement" (Cruz and James, 1999). Furthermore, Cruz and James (1999) also stress the importance of accommodating the search interface to both novice and expert users of multimedia retrieval systems. One example (of their system) is that novice users have the option of a Search Assistant, which may assist in "pre-query refinement" (Cruz and James, 1999).

Spink et al. (2001) also pay close attention to query generation of the user. They claim that, "Web users generally search for multimedia information as they search for textual information" (Spink et al., 2001). Also, Spink et al. (2001), find that multimedia queries contain more search terms (mean of 2.4) than that of general (non-multimedia) Web queries (mean of 1.91).

Spink et al. (2001) further discovered that the term "video" is the most commonly used query term when users search for video content. This brings them to suggest other search features, such as a file extension (.mpeg, .avi, .mov, .wav, etc) search feature, which could be very helpful in user query formation (Spink et al., 2001).

While these search features may prove to be helpful in future (video/image) IR systems, current video/image IR systems still primarily utilize text-based searches. Other research has attempted to increase video/image IR system satisfaction by moving away from total reliance of textual searching and incorporating content-based searching. Zhou and Huang (2002) describe a retrieval system where contextual information (color, shape, texture, etc. described as "low-level features") is combined with user's keywords (or "high-level semantic concepts"). They go on to present that searching on contextual information alone is usually not sufficient in generating relevant results, however, would serve the purpose in thesaurus updating (or adaptation of keywords to images and vice versa) (Zhou and Huang, 2002).

These studies, along with numerous others, reflect the many different directions video retrieval research is currently taking. For example, image and video analysis has been ongoing for many years, and the advancement of this research can be used to explore technologies of (image/video) retrieval. Also, the growing number of users searching for video content has spurred a movement toward applying user-centered design concepts in the development and evaluation of video/image IR systems.

Problem

As mentioned in the earlier sections of this paper, our problem focuses on providing easy and efficient access to video content where large archived (video) data exists. The video data provided by TREC proved to be sufficient in exploring these research problems with ViewFinder. Moreover, the total size of the video collection consisted of 68.45 hours (of MPEG-1) including 40.12 hours for the search test collection, 23.26 hours for the feature development collection, and 5.02 hours for the feature test collection (Smeaton and Over, 2002).

To conduct system tests and the TREC tasks we applied the existing ViewFinder system to the data provided by the conference (through the Internet Archive). Some of the modifications we made to the system consist of a reformulating (system) queries, switch from a MySQL database to Oracle, user-interface adjustments, incorporated a textual keyword search feature, and adapted the search attributes (for proper interaction with the Internet Archive video metadata).

Due to time constraints, the Oracle database resulted in a very basic structure. The indexed metadata for individual *video files* include the titles, descriptions, and descriptors (provided by the Internet Archive), which are identified by an automatically generated video id. For each individual *shot* (keyframe) there is a thumbnail, corresponding URL (to the shot keyframe/thumbnail), and an automatically generated id for both video and shot source. There were a total of 3 tables created for the database. One table contains the shot data (keyframe/thumbnail URL, shot id, and video id), and the other tables contain data corresponding to the video files (e.g. one table contains video id, title, and description fields; and the other table contains video id and descriptor fields).

Although, this proved to be sufficient data to develop a prototype for participating in the video track, we initially assumed that it wouldn't serve the purpose of a practical video retrieval system. Moreover, although TREC evaluates search returns of individual shots (located within a video file), our database only included metadata corresponding to each individual video file. We would encounter the problem of not being capable to distinguish between shots located within the same video file (other than by visually evaluating the shots after a search has been performed).

This prevented us from measuring relevancy rankings between individual shots located within the same video file (e.g. all shots of a matching video are considered "relevant" by the system). For example, once the system identifies any relevant (or matching) video(s), all corresponding keyframes are returned to the user in sequential order. Moreover, if the system matches the user's query to 2 video files, which (both) contain 50 shots, the user would be presented with a shot order such as: shot 1 from video 1, shot 1 from video 2, shot 2 from video 1, shot 2 from video 2 … up until the final shot. (Here, it is up to the user whether or not to browse the returned keyframes.) In the latter sections of this paper we discuss future improvements of ViewFinder, that we feel will eliminate these problems for upcoming TREC conferences.

Methodology

For our interactive ("human in the loop") search run, we allowed the user to evaluate relevancy of the returned results. Moreover, it was up to the user whether or not to reformulate the query and continue searching, or stop and settle on results. In addition, we also made no attempt to restrict or assist the user in query formulation/reformation, nor did we place any time restriction on the user (for individual search topics).

While using the ViewFinder system, the user is given several options of searching techniques (or search features). One of the features consists of a keyword search (See Appendix A for Interface Snapshot). This search allows the user to type in keywords and compare them to the description (field) for each individual video file. If there are any matches between the keyword(s) and any video description, the keyframes corresponding to matching video(s) are returned to the user. Moreover, the keyword search performs a "phrasal" search, or in the case that more than one keyword (a phrase) is entered, the exact phrase must match within the video description in order for results to be returned.

The user is also presented with several (video) attributes in which they are allowed to browse. These attributes are presented to the user in a series of drop down menus (Also See Appendix A for Interface Snapshot). One example is that the user can select "Title" in the "Search By" drop down menu, and retrieve all the video titles in the collection. The user can then select a particular title (by clicking on it and highlighting it) and click the "Search" button, which will run a query for that particular title, and return the associated keyframes.

A similar operation can be conducted with the "Descriptors" option in the drop down search menu. However, unlike the title search (which will only return results for one individual video title) it is possible for the descriptors search to return shots from several different video files (if the same descriptors overlap for multiple videos).

Another search option of ViewFinder is the "Promote" search. This is found in the drop down menus located directly below each of the individual keyframe panels (excluding the middle keyframe). This "Promote" feature will take the descriptors associated with that particular keyframe, and compare it with the descriptors for all other video files, and return any matches. Once the "Promote" search feature has been utilized, the "promoted" keyframe is then displayed in the middle (#5) image panel.

Since ViewFinder can only display up to 9 individual keyframes at one time (8 for search results, and 1 for displaying "promoted" keyframe), the user is still capable of browsing all video shots/keyframes returned (in the case of there being more than 8 matching keyframe). Utilizing the "More Clips" and "Back" buttons located on the interface allows for such browsing. The "More Clips" button becomes initialized after more than 8 keyframes are returned by a query, and the "Back Button" is initialized after the "More Clips" button has been clicked (and the user is on a page other than the first).

These search sessions ended when the user felt they exhausted all relevant video shots. After the user decided to end each of the search topics, they would select the "Finish" button, which would print out up to 100 (top) search results to the Java console, where they were gathered and formatted.

Results

Human assessors from NIST manually judged the relevancy of each returned shot. After concluding on the number of relevant shots returned as compared to the total number of relevant shots identified in the data set, an averaged precision was assigned to each search topic performed. You can read more on the averaged precision formulation in Vorhees, E. M., and Harman, D. K. (2001).

We conducted 1 interactive search run where we attempted to answer all 25 search topics. The mean averaged precision of ViewFinder for all 25 search topics was 0.05472. We had a range of 0.251 with a minimum score of 0.000 (on topics 75 and 85) and a maximum of 0.251 (on topic 76). Ranking among other participating systems included a range from 1st (0.170 topic 94) to a tie for worst (0.000 topics 75 and 85). Moreover, our average ranking for the 25 topics was 17.36 out of an average of 36.88 participating runs. However, there may be some discrepancy in comparing our results with the results of other systems for the reason that search runs (for other systems) varied from interactive to manual, and system development varied from type-A to type-B. (To explore the differences between interactive and manual search runs, and type-A systems and type-B system development please refer to Smeaton, A. F, and Over, P. (2002)).

Conclusions

After reviewing the results, we were initially correct in assuming that the lack of metadata for each individual shot greatly inhibited ViewFinder's searching performance. In future video tracks we plan on populating a database with metadata for each individual shot, which will provide a more robust search for specific information needs. Instead of limiting the search attributes to title, description, and descriptors alone, we would like to add attributes for keywords, subject(s), notable people, important landmarks, and landscapes/cityscapes (just to name a few). Also, we hope to incorporate content-based image retrieval in future versions of ViewFinder. This will allow users to build a more diverse search strategy and allow searches for shots/keyframes with similar shapes, patterns, and colors.

References

Choi, Y. & Rasmussen, E. M. (2002). Users' relevance criteria in image retrieval in American history. *Information Processing & Management, 38*(5), 695-726.

Cruz. I. F., & James, K. M. (1999). User interface for distributed multimedia database querying with mediator supported refinement. *Proceedings of the International Database Engineering and Applications Symposium, Montreal, Canada,* 433 – 441.

Hassan, I, & Zhang, J. (2001). Image search engine feature analysis. *Online Information Review 25* (2), 103 - 114.

Smeaton, A. F., & Over, P. (2002). The TREC 2002 Video Track Report. *Presented at the Eleventh Annual Text Retrieval Conference, Gaithersburg, MD,* November 19th – 22nd, 2002.

Smeaton. A. F., Over, P., & Taban, R. (2001). The TREC-2001 Video Track Report. *NIST Special Publications 500- 250: The Tenth Annual Text Retrieval Conference, Gaithersburg, MD,* 52 – 60.

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

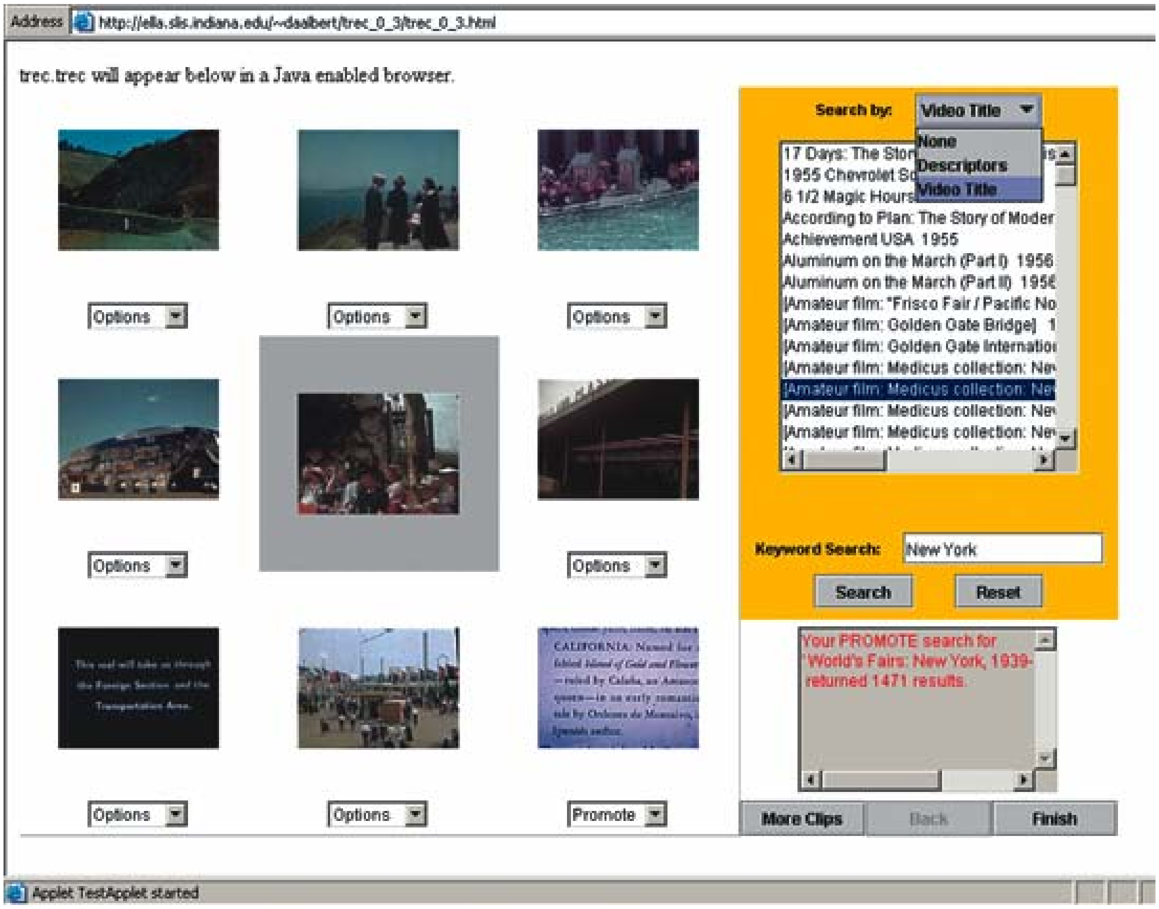
Vorhees, E. M., & Harman, D. K. (Eds.). Common Evaluation Measures. (2001). *NIST Special Publication 500-250: The Tenth Text Retrieval Conference,Gaithersburg, MD*, A14 – A23.

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's user-interface. Several search features are being displayed. The "Search By" menu (right hand side) is querying for the titles of the video files. Video titles are then listed in the text box below the "Search By" menu, where one title is highlighted. The "Keyword Search" text field, where the phrase "New York" is entered, is located below the video title listing. Various functions including "Search", "Reset", "More Clips", "Back," and "Finish" buttons are also located below the keyword text field.

The individual thumbnails/keyframes are displayed to the left of the search features. The "Promote" feature has been utilized and the corresponding keyframe is now displayed in the middle image panel.



Address http://ella.slis.indiana.edu/~daalbert/trec_0_3/trec_0_3.html

trec.trec will appear below in a Java enabled browser.

Search by: Video Title

17 Days: The Stor
1955 Chevrolet Sc
6 1/2 Magic Hours
According to Plan: The Story of Moder
Achievement USA 1955
Aluminum on the March (Part I) 1956
Aluminum on the March (Part II) 1956
[Amateur film: "Frisco Fair / Pacific No
[Amateur film: Golden Gate Bridge] 1
[Amateur film: Golden Gate Internatio
[Amateur film: Medicus collection: Nev
[Amateur film: Medicus collection: Nev
[Amateur film: Medicus collection: Nev
[Amateur film: Medicus collection: Nev

Keyword Search: New York

Your PROMOTE search for 'World's Fairs: New York, 1939- returned 1471 results.

Applet TestApplet started