LeeDeo: Web-Crawled Academic Video Search Engine

Dongwon Lee*   Hung-sik Kim   Eun Kyung Kim   Su Yan   Johnny Chen   Jeongkyu Lee[+]

Penn State University, USA   [+]University of Bridgeport, USA

{dongwon,hungsik,ezk112,syan,jzc160}@psu.edu,   [+]jelee@bridgeport.edu

Abstract

We present our vision and preliminary design toward web-crawled academic video search engine, named as LeeDeo, *that can search, crawl, archive, index, and browse "academic" videos from the Web. Our proposal differs from existing general-purpose search engines such as Google or MSN whose focus is on the search of textual HTML documents or metadata of multimedia objects. Similarly, our proposal also differs from existing academic bibliographic search engines such as CiteSeer, arXiv, or Google Scholar whose focus is on the search and analysis of PDF or PS documents of academic papers. As desiderata of such an academic video search engine, we discuss various issues as follows: (1)* Crawling*: how to crawl, identify, and download academic videos from the Web? (2)* Classification*: how to determine the so-called academic videos from the rest? (3)* Extraction*: how to extract metadata and transcripts from the classified videos? (4)* Indexing*: how to build indexes for search engines? and (5)* Interface*: how to provide interface for efficient browse and search of academic videos?*

1. Introduction

Large-scale bibliographic search engines such as Cite-Seer[1], Google Scholar[2], or Microsoft Libra[3] have proven themselves to be immensely useful for diverse users on the Web looking for research articles or metadata therein. Such search engines exploit the hyperlinks on the Web in crawling, gathering, and updating contents. Therefore, they often contain a large number of bibliographic data objects with considerable noises and errors (due to imperfect software artifacts). So far, main data objects supported by such search engines have been confined to mostly research articles in PDF or PS format.

As technologies rapidly advance, however, we envision that the way scholars disseminate their findings in this hypertext era be substantially augmented. For instance, in addition to publishing research articles in publication outlets, scholars may summarize their findings in a short video clip and circulate it for wider dissemination in YouTube[4] or Yahoo Video[5]. For instance, [19] discusses such an idea early on. Similarly, increasingly more number of publishing venues record author's presentation as a video clip and share them on the Web. Colloquium talks are frequently recorded and posted to departments' web pages. Furthermore, many schools start to post instructors' lectures as video clips to the Web for wider audience. Potentially, such multimedia-rich mediums are very useful for many people. However, no hypertext based search engines have existed that can crawl, index, and search such "academic" multimedia-rich data objects as scholar's presentation video in a conference, expert's tutorial presentation, or instructor's lecture video.

Our approach, termed as LeeDeo (LEarning and Educational viDEO Search Engine), differs from existing general-purpose search engines such as Google or Ask.com[6] whose focus is on the search of textual HTML documents or metadata of multimedia objects. Similarly, LeeDeo also differs from commercial video search engines such as Live Video Search[7] or Google Video[8] in that: (1) LeeDeo specifically focuses on gathering and indexing only "academic" or "educational" video clips from the Web, and (2) while existing video search engines' indexing is limited to associated keywords such as titles of videos or snippets of pages from which videos are found, LeeDeo aims to provide more comprehensive indexing by exploiting the extracted transcripts from videos as well as state-of-the-art video abstraction and content-based indexing techniques

---

[1] http://citeseer.ist.psu.edu/
[2] http://scholar.google.com/
[3] http://libra.msra.cn/

[4] http://www.youtube.com/
[5] http://video.yahoo.com/
[6] http://www.ask.com/
[7] http://video.live.com/
[8] http://video.google.com/

(e.g., [2, 4, 13, 18]). Finally, LeeDeo also differs from the well-known educational video repositories such as the Open Video Project [11] in that LeeDeo builds video collections using a search engine framework with less human intervention and a much larger scale. Due to the proliferation of available educational video clips on the Web in recent years, we believe that the search engine based on comprehensive collection is much needed (at the cost of some noises and errors in its data quality).

In this paper, therefore, we present our vision and preliminary design of LeeDeo toward such a web-crawled academic video search engine. Our narrow focus on the "academic videos only" yields several interesting research (both technical and non-technical) issues, many of which we do not have definite answers or solutions yet. Some of them include: (1) What is the academic video? How to detect academic videos from others on the Web? How to crawl and update them fast? How is it different from other *vertical* or *niche*[9] search engines? (2) Can one exploit the peculiar features of academic videos? For instance, assuming all academic videos must have an audio channel embedded, can one take advantage of it for better data extraction, classification, and indexing? (3) Multimedia-rich data objects on the Web often imply different copyright issues from HTML web pages or PDF research papers (e.g., video clips with the YouTube logo). What are the implications and opportunities? (4) How can K-12 educators use such an academic video search engine in classrooms? How will researchers and graduate students use it for learning and disseminating new information?

The current architecture of the LeeDeo consists of: (1) *Crawler*: Utilizing the hyperlinks on the Web, a crawler travels web pages to gather academic videos from educational as well as commercial sites. First-level filtering of irrelevant videos is done, too; (2) *Extractor*: Various metadata (e.g., author, title, video length) and associated information (e.g., URL to videos, site information, keywords) of videos are extracted and cleaned. Transcripts of videos are also extracted using speech recognition facility; (3) *Classifier*: Using extracted information, various classification tasks are conducted. Non-academic videos (which passed the first-level filtering by the crawler) are classified and filtered out, And, then appropriate tags/keywords are assigned to videos (e.g., Physics colloquium video, Conference presentation video); and (4) *Indexer*: Based on extracted metadata and transcripts, IR style indexes are built. For instance, *tf.idf* like weighting can be applied to tokens in transcripts. Due to the abundant information found in the extracted transcripts (compared to a limited set of video metadata), in-

---

[9]The vertical or niche search engine is a specialized search engine designed to search and index specific topics (e.g., Travel, Medicine) or formats (e.g., PDF articles), in contrast to the general search engine that aims to search as many topics and formats as possible.

dexes with a better quality can be built.

2. Academic Video Search Engine

What are "academic" or "educational" videos? Inherently, the definition of academic videos is fuzzy so that we do not attempt to give one. Rather, informally, we first list the kind of videos that we are interested in collecting and archiving in this project. Incomplete list includes: (1) Lecture, tutorial, and demonstration videos, (2) Conference and workshop presentation videos, (3) Colloquium and invited talk videos, and (4) Educational User-Created Contents (UCC) videos. For instance, some examples are shown in Table 1. Overall, we are interested in collecting videos that are useful for educational purpose and designed to disseminate new findings or understanding to wide audience. In terms of idiosyncratic features, academic videos are often relatively long (from 20-30 minutes to 1-2 hours), compared to other short video clips (e.g., 4-5 minutes music videos), and carry the main message in its audio channel (e.g., a biologist explains her main findings "verbally" while using slides and chalk/board). Of course, sometimes, there are educational videos that rely on animation or such. However, we believe that such cases are rare. Therefore, videos that do not contain any audio channel, for instance, can be quickly ruled out as "non academic videos" by our crawler. Furthermore, keywords and vocabularies spoken by the presenter in academic videos are often domain-specific, narrowly-focused, and distinctive (e.g., Multimedia colloquium talks would use terms like "CBIR" or "index" more often).

2.1. Crawling for Videos

In general, the number of plausible web sites where academic videos can be found are far more limited than that of those sites having HTML pages or PDF articles. For instance, it is more likely that educational video repositories such as those in Table 1 or scholars' home pages in *.edu* domain (in US) have academic videos linked. Therefore, by periodically monitoring those plausible web sites (as seed pages), one can have a relatively-complete academic video search engine. Another possible solution is to rely on existing general or video search engines such as Google, MSN, Yahoo Video, Google Video, etc. Using a set of keywords (e.g., "video", "talk", and "colloquium") as input queries, for instance, one can query existing search engines and use the intersection or union of returned pages as seed pages. We plan to investigate both approaches. Nevertheless, in the long run, crawling the Web for academic videos is a challenging task. In particular, the following challenges must be addressed.

Example	URL
CS conference presentation	`http://videolectures.net/`
CS conference presentation @ KDD	`http://www.kdd.org/webcasts.php`
CS conference presentation @ VLDB	`http://organ.kaist.ac.kr/videos/videos.html`
Technical presentation	`http://code.google.com/edu/videolectures.html`
Technical presentation	`http://www.youtube.com/profile?user=googletechtalks`
Technical presentation	`http://research.google.com/video.html`
Econimics presentation	`http://eclips.cornell.edu/homepage.do`
Biological presentation	`http://www.jove.com/`
Biological presentation	`http://www.dnatube.com/`
General scientific presentation	`http://www.researchchannel.org/prog/`
General scientific presentation	`http://scivee.tv/`
General scientific presentation	`http://sciencehack.com/`
Lectures @ Berkeley	`http://webcast.berkeley.edu/index.php`
Lectures @ MIT	`http://web.sls.csail.mit.edu/lectures/`
Lectures @ Princeton	`http://www.princeton.edu/WebMedia/lectures/`
Lectures @ YouTube	`http://youtube.com/view_play_list?p=803563859BF7ED8C`
Colloquium	`http://web.mit.edu/people/cabi/Links/physics_seminar_videos.htm`

Table 1. Examples of seed URLs for crawling for academic videos.

Site	Author	Title	Tags	Description	Submission Time	Video Length	Recording Date	Comment	URL
youtube.com	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
video.google.com	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
videolectures.net	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
researchchannel.org	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
archive.org	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
scivee.tv	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
dnatube.com	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
jove.com	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes

Table 2. Example metadata of videos and their availability on the Web.

First, often, videos are embedded in web pages using peculiar features of some video players (e.g., flash player). Therefore, being able to automatically detect and download a myriad types of embedded videos from the Web is challenging since simply there are so many varieties for a crawler to deal with and there is little governance of standards. Second, in general, the average size of embedded videos on the Web is much larger than that of HTML pages or PDF articles. For instance, in our preliminary testing, about 60% of videos are less than 50MB and the rest of 40% are as large as 240MB. Therefore, while downloading and storing such videos, a crawler has to consider the network bandwidth and storage issues carefully in its crawling policy [12]. To our best knowledge, no existing crawling policy literatures address the issue of the efficient download of large data objects like video clips. Third, to avoid unnecessary waste of storage space and copyright violation, it is important to be able to detect redundant copies of videos or copyright-violated videos while crawling using techniques such as [5, 6]. However, this often requires sophisticated and expensive contents-based video comparison or copied video detection techniques. How to incorporate such services into a crawler using simple disambiguation techniques (e.g., [7, 15]) needs to be carefully studied.

In the current prototype (to be elaborated in Section 3),

we use tools such as WMRecorder[10], vdownloader[11], or Python scripts to download videos from the Web, and support both .avi and .flv types.

2.2. Extracting Metadata of Videos

Various metadata of videos are available – some are about the characteristics of individual video clips while others are about the site from which video clips are downloaded. Some semantic metadata even can be derived from contexts or other metadata [16]. Rich metadata helps users in browsing and searching videos. However, since our videos come from diverse data sources without uniform standards to follow, available set of metadata varies. For instance, Table 2 shows a small subset of metadata that can be extracted from eight sources. Note that different sites support different metadata. While some sites such as YouTube provide a rich set of metadata of videos in XML format, other sites display metadata of videos in web pages, making the extraction challenging.

---

[10]`http://www.wmrecorder.com/`
[11]`http://www.vdownloader.es/`

2.3. Extracting Transcripts from Videos

One of unique features of academic videos is that they all must have "audio" channels – i.e., speaker must speak in the videos. Therefore, videos without a sound track will not be collected by crawlers in the first place. Furthermore, we can exploit this feature by extracting transcripts from audio channel using speech recognition software. Similar idea was explored in [3] with a limited digital library context. Below, we briefly describe the current design in the prototype. First, audio channels are extracted from videos using the *ffmpeg* library, which is free and open-source C implementation supporting most codecs[12]. The ffmpeg supplies very fast and powerful encoding and decoding schemes via simple command-line interface. Then, we have .mp3 files extracted for each video file. Next, transcripts are automatically extracted from .mp3 files by using speech recognition software such as *IBM ViaVoice* and *Dragon NaturallySpeaking*. In the current prototype, NaturallySpeaking was used to dictate the voice of academic videos.

Several technical issues still need to be addressed. First, continuous speech recognition under some complex circumstance (e.g., noise and speaking from multiple persons) is still a challenging task, if not impossible. In many academic settings, different people from different countries/regions tend to have different accents, resulting in a very difficult task of speech recognition. Therefore, it is infeasible to depend completely on speech recognition to make the transcript extraction. However, we still believe that transcripts of videos are good corpus that can significantly improve various features of video search engines such as indexing or crawling. Therefore, even if extracted transcripts contain a lot of noises and errors, with proper post-processing, a wealth of valuable information can be unearthed. Second, the best accuracy of extracting transcripts is currently achieved when the audio channel is played at a regular speed (as if a speaker talks in real time). However, in the academic video search engine, the number of videos to transcribe ranges from thousands to hundreds of thousands of videos. Therefore, such a slow scheme to extract transcripts would not work. One needs to devise more scalable schemes such as the parallel execution of the speech recognition routine in a cluster environment. Another solution is to reduce the number of videos to process. For instance, instead of transcribing 60 minutes-long videos, one can identify smaller segments of more importance to have faster transcription. In the sub-section, we discuss our early idea toward this approach.

2.4. Abstracting Videos

Video abstraction is defined as a short summary of a long video consisting of a set of still or moving images, which can be selected and reconstructed from an original video with concise information about the contents. Techniques in video abstraction can be categorized into two main approaches, i.e., static video summary and dynamic video skimming. In the static video summary (e.g., [1, 4, 13, 18]), a collection of static images or key frames of video segments are selected and reconstructed. Since the approach allows nonlinear browsing of video contents, the time constraint cannot be preserved. On the other hand, dynamic video skimming (e.g., [2, 9, 10, 14, 17]) provides a shorter version of a video arranged in time. Since it browses certain portions of video contents, the time constraint can be preserved.

For both approaches, segmentation of video into proper units (i.e., shots or scenes) plays a very important role in maximizing the quality of video abstraction. However, due to the nature of academic videos, i.e., captured by a single camera operation without stop, there is no clear boundary of shot in the video. In order to address this, we segment a video into a number of small clips using the time gaps in speaker's speech, which do not have any audio information, i.e., silence or pause. For example, there exist time gaps when a speaker changes presentation slides, brings a new topic, or prepares demonstrations. If the time gap exceeds a certain threshold, we regards the gap as a boundary of segments. Each segment is called as "Slice" (or video slice) to distinguish it from other segments used in general videos, such as shot, scene or event. If a segment is too long, it is re-segmented into "Sub-slices".

After a video is segmented into a number of slices (or sub-slices), we select and reconstruct the slices for abstracting video that is used for efficient processing of transcripts extraction mentioned in Section 2.3. For the abstraction, we exploit the idea of dynamic video abstraction proposed in [8], and then extend it into abstracting academic videos. Unlike [8], where only visual contents are considered for video summarization, our approach in three steps uses both audio as well as visual features: (1) We first segment a video into slices (or sub-slices) by using time gaps of audio information; (2) We next construct a scene tree with the detected sections, which illustrate the hierarchical contents of the video, and generate multi-level scenarios using the constructed scene tree which provides various levels of video abstractions; and (3) Finally, using the multi-level scenarios, we generate two different types of abstraction: multilevel highlights and multi-length summarizations.

The generated abstractions are used to extract transcripts from video efficiently. In other words, the transcripts are extracted only from the abstracted video rather than entire
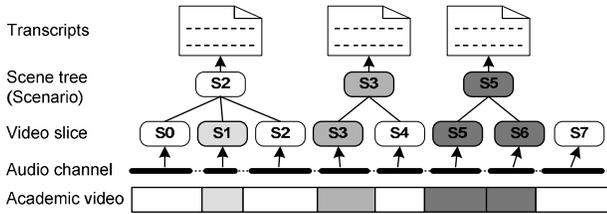
Figure 1. Abstracting academic videos.

Figure 3. The impact of the length of transcripts toward the accuracy in classification.

one. Furthermore, the summarized video can be used for the better interface of LeeDeo. For example, the summarized video can be displayed as 'Preview' or 'Trailer'. Also, several key frames that represent the abstracted video can be used for thumbnail images in LeeDeo. Figure 1 shows an example of it. In the example, an academic video (i.e., colored bar) is segmented into a number of slices (i.e., small rectangles) by using time gaps in audio information (i.e., solid lines). The video slices are used for a scene tree construction, and consequently the video is abstracted as $S2$, $S3$, and $S5$ slices. To reduce the processing time of extracting transcript from a video, we can extract transcripts from selected slices (i.e., $S2$, $S3$, and $S5$) rather than an entire video.

2.5. Classifying Videos

In this step, from many input videos, we aim at classifying and detecting only "academic" videos (or "colloquium" videos in the case of prototype in Section 3). In the preliminary experimentation, we used 64 positive (i.e., academic) and 100 negative (i.e., non-academic) sample videos with Support Vector Machine (SVM) as the main classification framework. As the feature of videos, no metadata are used for now. Rather, only transcripts extracted from the audio channels of videos are used. That is, we treated transcripts of videos as if they are document corpus and applied conventional document classification techniques using the SVM^{light} implementation[13].

Using randomly selected 50% as the training and the remaining 50% as the testing set, overall, we got 100% precision and 18.7% recall in classifying academic videos. As shown in Figure 3, when there are more information available in transcripts, it is easier to classify videos. For instance, a hour-long academic videos will yield lengthy transcripts with a lot of informative tokens, which in turn is likely to yield high rate of success in determining true positives and negatives. Since the preliminary experimentation is still very limited, more extensive empirical study is needed. For instance, clearly, other metadata such as URL o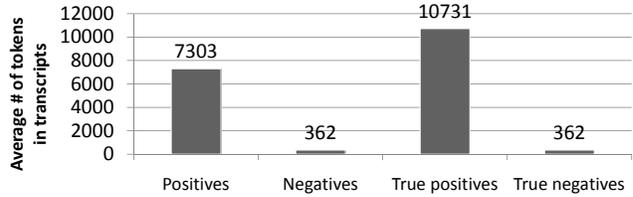r title of videos would be helpful in the classification process and need to be exploited judiciously, and videos in diverse genres need to be used in the experimentation for conclusive analysis. We are currently carrying out such an experimentation.
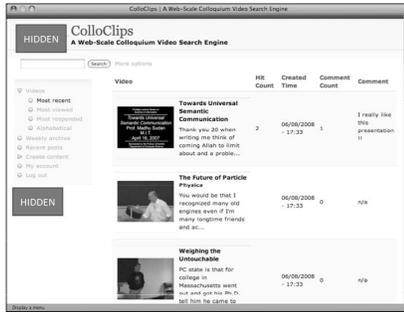
3. ColloClips as a Prototype

As a proof-of-concept of the LeeDeo project, first, we built an academic video search engine for colloquium videos only, termed as ColloClips. ColloClips is built using the Drupal[14] as the content management system and FlashVideo[15] as the video management module in the Drupal. The initial seed pages for ColloClips are found by search engines using queries like "`colloquium OR talk OR video site:*.edu`".

A few screenshots of ColloClips are shown in Figure 2, where (a) shows the main interface of ColloClips, (b) shows the page of an individual colloquium video, and (c) shows the results of search. In (b), bottom area shows a few automatically extracted metadata of the colloquium video (e.g., author name, video length) while scrollable panel on the right shows the extracted transcript of the corresponding video, with each 2-minute slice in different colors. In (c), 10 matches (out of 49) to the keyword string "Physics" are shown. Note that search is performed against the extracted transcripts, instead of metadata of videos (as a demonstration). Since extracted transcripts of videos provide richer corpus for indexing, instead of conventional CBIR, ColloClips provides indexing and searching based on keyword matching on transcripts by default. In the future, we will combine the indexing on both the metadata and transcripts of videos. The index in Drupal is maintained as a UNIX cron job when new contents are added. In a content management system like Drupal, everything is stored in the database (e.g., MySQL). An individual video embedded in each page of ColloClips, shown as Figure 2(b), is treated as a file and stored in the database.
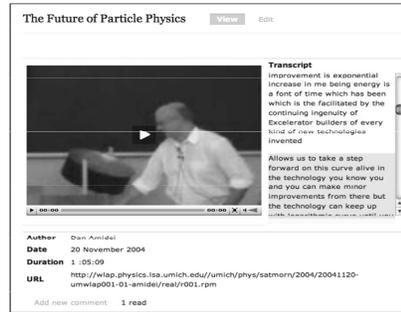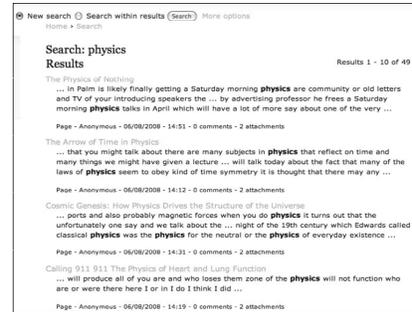
---

[13]http://svmlight.joachims.org/

[14]http://drupal.org/
[15]http://drupal.org/project/flashvideo

| (a) Main window | (b) Play window | (c) Search window |

Figure 2. The screenshots of ColloClips**.**

4. Conclusion

In this paper, we discussed our preliminary proposal for academic video search engines, termed as the LeeDeo, and their related issues. LeeDeo is different from existing search engines in that: (1) it focuses on "academic" or "educational" videos only, (2) it extracts various video metadata from videos themselves as well as from web pages from which videos are crawled, and (3) it extracts transcripts from the audio channel of videos using the speech recognition facility and uses them as the rich corpus for advanced indexing. Brief discussion on our initial design is provided as a motivation and a prototype system for colloquium videos, termed as the ColloClips, is presented. However, much research is needed to resolve many open challenging issues.

References

[1] Z. Cernekova, C. Nikou, and I. Pitas. "Entropy Metrics used for Video Summarization". In *18th Spring Conf. on Computer graphics*, 2002.

[2] H.-W. Chen, J.-H. Kuo, W.-T. Chu, and J.-L. Wu. "Action Movies Segmentation and Summarization based on Tempo Analysis". In *ACM Int'l Workshop on Multimedia Information Retrieval (MIR)*, 2004.

[3] M. G. Christel, T. Kanade, M. Mauldin, R. Reddy, M. A. Sirbu, S. M. Stevens, and H. D. Wactlar. "Informedia Digital Video Library". *ACM Comm. ACM*, 38(4):57–58, 1995.

[4] G. Ciocca and R. Schettini. "Dynamic Storyboards for Video Content Summarization". In *ACM Int'l Workshop on Multimedia Information Retrieval (MIR)*, 2006.

[5] H. Kim, J. Lee, H. Liu, and D. Lee. "Video Linkage: Group Based Copied Video Detection". In *ACM Int'l Conf. on Image and Video Retrieval (CIVR)*, Niagara Falls, Canada, Jul. 2008.

[6] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. "Video Copy Detection: A Comparative Study". In *ACM Int'l Conf. on Image and Video Retrieval (CIVR)*, Amsterdam, The Netherlands, July 2007.

[7] D. Lee, B.-W. On, J. Kang, and S. Park. "Effective and Scalable Solutions for Mixed and Split Citation Problems in Digital Libraries". In *ACM SIGMOD Workshop on Information Quality in Information Systems (IQIS)*, Jun. 2005.

[8] J. Lee, J. Oh, and S. Hwang. "Scenario based Dynamic Video Abstractions using Graph Matching". In *ACM Int'l Conf. on Multimedia (MM)*, 2005.

[9] B. Li and I. Sezan. "Event Detection and Summarization in American Football Broadcast Video". In *SPIE Conf. on Storage and Retrieval for Multimedia Databases*, San Jose, CA, Jan. 2002.

[10] R. Lienhart. "Abstracting Home Video Automatically". In *ACM Int'l Conf. on Multimedia (MM)*, Orlando, FL, Oct. 1999.

[11] G. Marchionini and G. Geisler. "The Open Video Digital Library". *D-Lib Magazine*, 8(12), 2002.

[12] F. McCown and M. L. Nelson. "Evaluation of Crawling Policies for a Web-Repository Crawler". In *ACM Conf. on Hypertext*, Odense, Denmark, Aug. 2006.

[13] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. "Video summarization and scene detection by graph modeling". *IEEE Trans. on Circuits and Systems for Video Technology*, 15(2):296–305, Feb. 2005.

[14] N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki. "Time-Compression: Systems Concerns, Usage, and Benefits". In *CHI Conf. on Human Factors in Computing Systems*, 1999.

[15] B.-W. On, D. Lee, J. Kang, and P. Mitra. "Comparative Study of Name Disambiguation Problem using a Scalable Blocking-based Framework". In *ACM/IEEE Joint Conf. on Digital Libraries (JCDL)*, Jun. 2005.

[16] I. Pandis, N. Karousos, and T. Tiropanis. "Semantically Annotated Hypermedia Services". In *ACM Conf. on Hypertext*, Salzburg, Austria, Sep. 2005.

[17] F. Shipman, A. Girgensohn, and L. Wilcox. "Generation of interactive multi-level video summaries". In *ACM Int'l Conf. on Multimedia (MM)*, 2003.

[18] Y. Takeuchi and M. Sugimoto. "Video Summarization using Personal Photo Libraries". In *ACM Int'l Workshop on Multimedia Information Retrieval (MIR)*, 2006.

[19] W. Wolf, B. Liu, A. Wolfe, M. M. Yeung, B.-L. Yeo, and D. Markham. "Video as Scholary Material in the Digital Library". In *IEEE Advances in Digital Libraries (ADL)*, McLean, VA, USA, 1995.