ONTOLOGY-DRIVEN ANNOTATION AND ACCESS OF PRESENTATION VIDEO DATA

[1]Aijuan Dong, [2]Honglin Li

[1]Department of Computer Science, Hood College , Frederick, MD 21701
[2]Truveo / AOL Video Search, 333 Bush Street, 23rd Floor, San Francisco, CA 94104
Email: [1]dong@hood.edu

ABSTRACT

The tremendous growth in video data calls for efficient and flexible access mechanisms. In this paper, we propose an ontology-driven framework for presentation video annotation and access. The goal is to integrate ontology into video systems to improve users' video access experience.

To realize ontology-driven video annotation, the first and foremost step is video segmentation. Current research in video segmentation has mainly focused on the visual and/or auditory modalities. In this paper, we investigate how to combine visual and textual information in the hierarchical segmentation of presentation video data. With average F-scores over 0.92, our experiments show that the proposed segmentation procedure is effective.

After a video is segmented, video annotation data can be extracted. To extract annotation data from a video and its segments, and to organize them in a way that facilitates video access, we propose a multi-ontology based multimedia annotation model. In this model, a domain-independent multimedia ontology is integrated with multiple domain ontologies. The goal is to provide multiple, domain-specific views of the same multimedia content and, thus, better address different users' information needs.

With extracted annotation data, ontology-driven video access explores domain knowledge embedded in domain ontologies and tailors the video access to the specific needs of individual users from different domains. Our experience suggests that ontology-driven video access can improve video retrieval relevancy and, thus, enhance users' video access experience.

Keywords: *video segmentation, video annotation, video access, ontology, and information retrieval.*

1. INTRODUCTION

In the past decade, we have witnessed unprecedented advances in multimedia technology. As a result, an unprecedented amount of multimedia data is being generated. Among the myriad types of multimedia data, presentation videos from lectures, conferences and seminars, and corporate trainings are of particular interest to the research reported here.

The need for specific solutions in this field comes from the popularity of e-learning systems. Recent years, there have been extensive efforts at both universities and colleges on developing e-learning systems to support distant learning. By 2003, 84 percent of US colleges have e-learning programs (Kariya, 2003). The demand for e-learning continues to grow. Some analysts predict that up to one-half of traditional campus programs will soon be available online (Frydenberg, 2000; Bishop & Spake, 2003; Dunn, 2000; Winsboro, 2002). In addition, there are e-learning systems for military, medical, and cooperate trainings (Smith, Ruocco, & Jansen, 1999; Fan, Luo, & Elmagarmid, 2004). For example, Microsoft supported 367 on-line training lectures with more than 9000 online viewers in the year of 1999 alone (He, Grudin, & Gupta, 2000). These e-learning systems enhance learning experiences and augment teachers' work in and out of traditional classrooms (Abowd, Brotherton, & Bhalodai, 1998; Flachsbart, Franklin, & Hammond, 2000). Working professionals as well embrace e-learning programs due to their convenience and flexibility (Kariya, 2003). However, due to unstructured and liner features of videos, the essential instructional content of most e-learning systems, the presentation videos, has not been fully exploited. People often feel difficulties in locating a specific piece of information in a presentation video. Sometimes they have to play back and forth several times to locate the right spot. To ensure effective exploitation of these video assets,

efficient and flexible access mechanisms must be provided.

Video annotation data play a critical role in video systems. The richer the annotation data are, the more flexible the video access becomes, and thus the more effective the video data can be utilized. We view video annotation as a two-step process: video segmentation, and video annotation data extraction and organization. The former divides a continuous video stream into a set of meaningful and manageable segments, and the latter extracts various annotations from these segments and organizes them in a way that facilitates efficient video access. In the following sections, we will examine the existing systems or approaches with regards to these two aspects.

A variety of techniques have been proposed to segment presentation videos. Earlier work from the Cornell Lecture Browser (Mukhopadhyay & Smith, 1999) uses feature differences between binary slide images to segment a slide video stream. Later, Yamamoto et al. (2003) propose topic segmentation of lecture videos by computing the similarity between topic vectors obtained from a textbook and a sequence of lecture vectors obtained from a lecture speech. In another paper, a content density function is proposed based on the observation that topic boundaries coincide with the ebb and flow of the "density" of content shown in videos (Phung, Venkatesh, & Dorai, 2002). Using various visual filters, Haubold and Kender (2003) utilize key frames in instructional video segmentation. Extracted key frames are first assigned a media type. Key frames are then clustered based on visual contents. Recently, Lin et al. (2005) investigate a linguistics-based approach for lecture video segmentation. Multiple linguistic-based segmentation features from lecture speech are extracted and explored. Similar approach has been explored in this paper, where segmentation positions are estimated with comparisons of successive indexes using dynamic programming (Kanedera, Sumida, Ikehata, & Funada, 2006). Related work in this field also include these (Onishi, Izumi, & Fukunaga, 2000; Rui, Gupta, Grudin, & He, 2002; Liu & Kender 2002; Ngo, Wang, & Pong 2003). Despite many successes, most approaches described above focus on linearly segmenting video streams into smaller units using information from single modality. In this paper, we investigate how to combine visual and textual information in the hierarchical segmentation of presentation videos.

After a video is segmented, video annotation data can be extracted from the video and its segments. In our study, we find that one of the problems in current video systems is that there exists a gap between users' information needs and video content representation. On one hand, users from different domains or with different backgrounds perceive video content from different angles and are only interested in particular type of information. On the other hand, most existing video systems have only one representation of video content. Thus, it is very difficult for these systems to provide multiple and customized views to users from different domains. As a result, the degree of video retrieval relevancy is low. Another overlooked problem in most video systems is the organization of video annotation data. Syntactic relations and semantic constraints are not sufficiently enforced in current annotation data organization. Thus, it is difficult to extract relevant information from the ever-growing multimedia data collection.

Research has been conducted to address these problems. Relevance feedback has been used widely in image retrieval to adjust user queries and provide better approximation to the users' information needs (Rui, Huang, Mehrotra, & Ortega, 1998; Cox, Miller, Omohundro, & Yianilos, 1996; Cox, Miller, Minka, & Yianilos, 1998; Papathomas, et al., 1998; Minka & Picard, 1997). However, this technique is proposed under the assumption that high-level semantic concepts can be captured by low-level multimedia features, which is not always the case (Cox, et al., 1996), such as high-level abstract concepts in scientific domains. Therefore, relevance feedback cannot be used to approximate users' information needs under these situations.

With the development of semantic web, several ontologies have been developed to annotate and represent multimedia content in recent years (Khan & McLeod, 2000; HyvÄonen, Styrman, & Saarela, 2003; Schreiber, Dubbeldam, Wielemaker, & Wielinga, 2004; Bao, Cao, Tavanapong, & Honavar, 2004; Hauptmann, 2004; Tsinaraki, 2004; Hollink, Worring, & Schreiber, 2005). Despite many initial successes, one problem with most existing approaches is that one ontology targets one specific domain or data collection. A new ontology is generated by combining domain specific knowledge with a multimedia ontology. The ontology is then used to annotate multimedia data in an effort to integrate domain knowledge into multimedia access and

increase the degree of retrieval relevancy. As a result, such an ontology only works for users from one specific domain and it cannot meet the information needs of a variety of users. In this paper, we propose multi-ontology based multimedia annotation. Although this multi-ontology annotation model applies to multimedia in general, we focus on our discussion on presentation video data.

Based on the discussion above, we propose a framework for ontology-driven presentation video annotation and access in this paper. The rest of paper is organized as follows. Section 2 introduces the framework for ontology-driven video annotation and access. Section 3 discusses multi-mode video segmentation. We detail the hierarchical segmentation of presentation videos through visual and text analysis. Section 4 proposes multi-ontology based video annotation. After video is segmented and metadata is extracted, Section 5 describes ontology-driven video access. Section 6 implements an experimental video access platform to demonstrate the idea. Section 7 concludes the research and highlights the opportunities for future work.

2. THE ONTOLOGY-DRIVEN FRAMEWORK

In this section, we present the ontology-driven framework. The framework provides the readers with a high-level view of the research and lays the foundation for subsequent discussions.

The ontology-driven framework is proposed based on annotation-driven video systems. In a typical annotation-driven video system, video data is the combination of video production data (i.e., video raw data) and video annotation data (i.e., video metadata). Users interact with annotation data and then locate raw video data through the time stamps that are associated with the annotation data. As can be seen, it is the availability of the video annotation data that determines the functionalities and flexibilities of a video system.

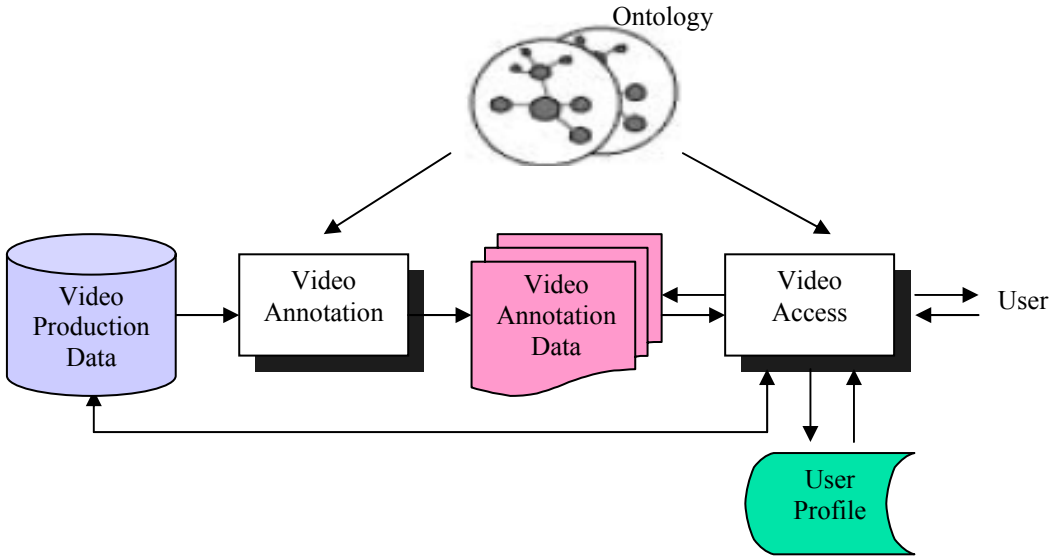


Figure 2.1. The framework of ontology-driven video annotation and access.

The ontology-driven framework integrates ontologies into video annotation and access. In this framework (Figure 2.1), video data consists of both video production data and video annotation data, the same as that of annotation-driven video access. However, video annotation (i.e., the process of assigning various indices or annotations to a video) interacts with both video production data and ontology; video access operates on video production data, video annotation data, and ontology. Depending on applications, multiple ontologies can be incorporated. The goal is to integrate ontology into video systems in an effort to improve users' video access experience. We argue that ontology-driven video annotation and access can improve users' video access experience. The integration of ontology has the following advantages. First, ontology describes concepts and

their relationships in a formal way. By semantically refining video queries based on these relationships, relevant concepts can be extracted. Since this relevancy information is extracted directly from ontology where domain knowledge is embedded, there is a potential to increase the degree of video retrieval relevancy. Second, in the ontology-driven framework, multiple ontologies can be integrated. Different ontology describes the same video content from different perspective. This enables multiple content representations of the same video content. Thus, different users' information needs are addressed. Third, the controlled vocabulary of ontology is exploited to annotate and access video data, which alleviates the problem of inconsistency in annotation data and thus enables information sharing and exchange among different parties. In other words, ontology facilitates information retrieval over collections of heterogeneous and distributed information sources. Finally, ontology represents knowledge in a machine-processable format, which means that we can use computer programs/user agents to process information and infer knowledge. This is especially important when a large amount of videos are disseminated over the web.

3. HIERARCHICAL SEGMENTATION OF PRESENTATION VIDEOS THROUGH VISUAL AND TEXT ANALYSIS

Video segmentation addresses the issue of granularity and answers the question of what to index. Thus, video segmentation is the first and one critical step towards automatic annotation of digital video sequences. In our study, we observe that a presentation usually consists of many topics, and each topic covers several slides. This inherent structure enables hierarchical segmentation, indexing, and access of presentation videos. Moreover, most presentations have the following two data sources: PowerPoint slide video stream (i.e., the video stream captures slide activity during presentation) and PowerPoint slide file, also called a PPT file. Both of them contain rich information about the video content. Thus, it is logical to use both of them in the segmentation of presentation videos.

3.1. Overview of the Approach

Based on the discussion above, this section proposes a hierarchical segmentation procedure for presentation videos through visual and text analysis (Figure 3.1). Specifically, a two-level video segmentation is investigated: topic-level and slide-level. Slide-level segmentation operates on slide video streams captured by a stationary camera, while topic-level segmentation makes use of extracted slide text.

Figure 3.1.shows that the first step in topic-level segmentation is text-based segmenting through Topic Words Introduction (TWI) that will be discussed later. TWI generates a sequence of slide blocks, each of which discusses one topic. To associate each slide block with its corresponding topic-level video segment, the temporal relationship between a slide video stream and slides must be established. This is accomplished by matching slide images converted from PowerPoint slides with key frames extracted from slide-level video segments. Based on timing information of each slide, slide

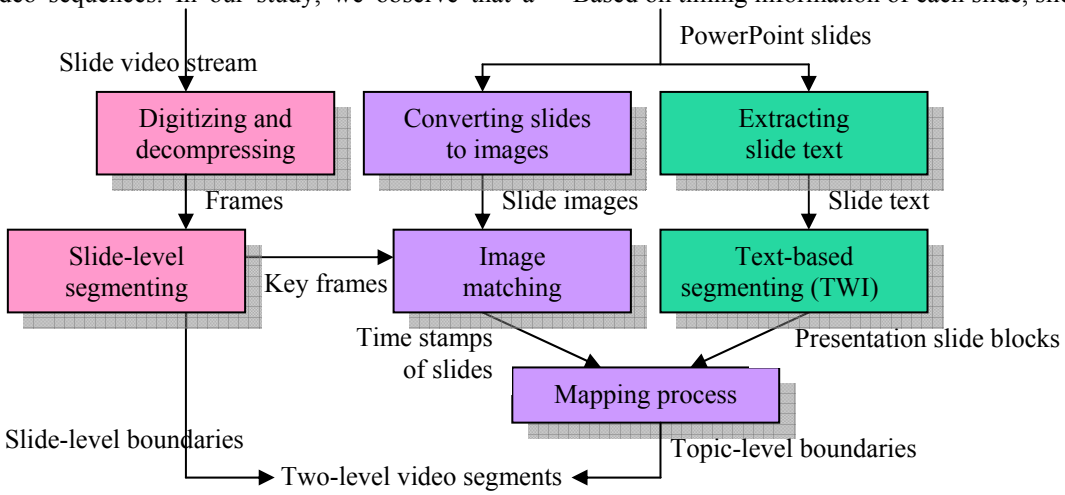


Figure 3.1. The segmentation of presentation videos.

blocks can be mapped with topic-level video segments, thus achieve hierarchical video segmentation. In the following subsections, we discuss in detail slide-level segmentation and topic-level segmentation.

3.2. Slide-level Video Segmentation

Slide-level segmentation divides a continuous slide video stream into a set of video segments, each of which matches one slide. More formally, given a presentation video stream v and a set of n slides, compute a set of video segments $VS_{slide-level} = \{vs_{0,slide-level}, vs_{1,slide-level}, ..., vs_{i,slide-level}\}$, such that the projected slide image of each video segment $vs_{i,slide-level}(0 \le i \le m)$ does not change.

Notice that this definition only requires that each video segment vs_i displays the same slide, but it does not impose that two adjacent segments display different slides. Thus, extra segments (false positives) are acceptable. If the matching process detects the same slide is shown in two consecutive video segments, then these segments will be combined. By allowing extra segments, it is less likely that slide transitions go undetected.

To segment presentation videos at slide-level, the feature of local color histogram is employed. We compare the local color histograms of adjacent successive frames. When the difference exceeds the pre-defined threshold, a slide-level boundary is declared. This approach is simple, but works well for presentation videos. This is because most slide transitions are abrupt cuts, and presentation videos do not have special video effects, such as fading, dissolve, and wipe.

3.3. Topic-level Video Segmentation

In our study, we observe that most presentations tend to follow a basic structure in spite of differences in contents and formats. A typical presentation, especially a conference presentation, starts with a title slide, then an outline/overview slide, which is followed by a number of content slides. The outline/overview slide of a presentation summarizes major topics that will be covered in content slides. In addition, the first-time introduction of a new topic in the content slides generally uses terms that are the same as or very similar to what occur in the outline/overview slides. Actually, most presenters intentionally construct such a structure in an effort to guide their presentation and engage the audience. Based on this observation of the presentation structure, we propose a text-based segmentation algorithm—Topic Words Introduction (TWI).

TWI segments a presentation into topically coherent slide blocks. More formally, given a presentation p and a set of n content slides, compute a set of slide blocks $SB = \{sb0, sb1, ..., sbk\}$, such that the topic of each $sbi(0 \le i \le k)$ does not change.

TWI algorithm works on slide text that is automatically extracted. Specifically, for each PPT slide file, extract slide content from its outline/overview slide and slide titles from its content slides. With the extracted text, TWI algorithm consists of three main phases: morphological analysis, lexical score determination, and boundary identification.

Phase one: *morphological analysis*. The purpose of this phase is to determine the terms to be used in the following phases. Two major processes in this phase are tokenization and stemming.

Tokenization refers to the process of dividing the input text into individual lexical units. With a regular expression recognizer and a stop-word[1] list, punctuation and uninformative words are removed. And the remaining slide text is converted to streams of tokens, including words, numbers, and symbols. Stemming is the process of reducing tokens to their roots, also called stems. The Porter's stemming algorithm (Porter, 1980) is used here for this purpose. It removes the common morphological and inflected endings from English words. Thus, the result of it is a set of word stems. These stems are considered the registered terms of a presentation.

An example output of morphological analysis for extracted slide text is illustrated in Figures 3.2 and 3.3. Line numbers are manually added for clarity. In Figure 3.2, each line correlates to one bullet/list in the overview/outline slide, while in Figure 3.3, each line associates with the slide title of a content slide.

[1] A stop-word is a word that lacks significance to the determination of the subject of a document.

1. background
2. barrier
3. experi gridblast keck center
4. lesson learn
5. acknowledg

Figure 3.2. Slide content from the outline/overview slide of a presentation.

.....¶
4. barrier¶
5. keck center¶
6. origin nongridawar configure keck center¶
7. scal large compare genom¶
8. schemat view web portal¶
9. current gridawar configure¶
10. compon gridblast¶
.....¶

Figure 3.3. Slide titles from the content slides of the same presentation.

Phase two: *lexical score determination*. The purpose of this phase is to measure the similarity between a topic and a slide. Since most presenters summarize their major topics in the outline/overview slides, analyzing extracted text from the outline/overview slide can identify the topics of a presentation. In our study, for each presentation, we take each natural line of text from its outline/overview slide as one topic (Figure 3.2). For example, "lesson learn" is one identified topic. If there is more than one level in the outline/overview slides, then the content of the first level is used. A dictionary of word-stem frequencies is constructed for each line of text and is represented by a vector of frequency counts. These vectors are called topic vectors in our discussion.

Content slides are summarized by their titles. Therefore, in the TWI algorithm, slide titles are used to represent the content slides of a presentation. For example, "scal large compare genom" in Figure 3.3 is such a slide title. Similarly, a dictionary of word-stem frequencies is constructed for each slide title. This is again represented as a vector of frequency counts. These vectors are called content vectors in our discussion.

To segment presentations at the topic level, we calculate the lexical scores between topic vectors and content vectors. Lexical score measures the lexical similarity between two vectors and is represented by cosine similarity measure (Formula 3.1)(Hearst, 1994).

$$score(i, j) = \frac{\sum_t wt, ti \; wt, cj}{\sqrt{\sum_t wt, ti^{\,2} \sum_t wt, cj^{\,2}}}, \qquad (3.1)$$

where t_i is a topic vector, c_j is a content vector, t ranges over all the registered terms of t_i and c_j, w_{t,t_i} is the weight assigned to term t in topic vector t_i, and w_{t,c_j} is the term weight assigned to term t in content vector c_j. Here, the weights on the terms are simply their frequency counts. For a presentation with k topics and n content slides, each topic has n lexical scores, and the total lexical score calculation is $k * n$.

Phase three: *boundary identification*. The method for boundary identification is based on lexical cohesion theory, which states that text segments with similar vocabulary are likely to be in one coherent topic. Thus, the more words two vectors share, the more strongly they are semantically related.

A lexical score between a topic vector and a content vector measures how strong these two are related, and is used here to determine topic boundary. The larger the score, the more likely the boundary occurs at that content slide. Steps for boundary identification are stated in Figure 3.4.

1. For each topic $i\left(0 \le i \le k\right)$
2. If there is lexical score(s) greater than zero
3. Set the boundary i where the first maximum lexical score is
4. Else
5. Locate the boundary $i-1$ and boundary $i+1$
6. Calculate lexical scores of adjacent content vectors within these two boundaries
7. Set a boundary where the lexical score is greater than the threshold $T1$
8. End if
9. End for

Figure 3.4. Boundary identification.

For each topic *i*, if there exists lexical score(s) greater than zero (line 2), then its boundary is set where the first maximum lexical score occurs (line 3), i.e., the position where the topic is first

introduced. Otherwise, if the lexical score equals to zero, the algorithm locates its previous and subsequent boundaries, and calculates the lexical scores of adjacent content vectors within these boundaries. After that, set a boundary where the lexical score is greater than threshold $T1$ (line 4-7). Instead of comparing with zero (line 2), a threshold may be used. Due to limited terms in both topic vectors and content vectors in the case of presentations, we found zero is a reasonable threshold here. This will be demonstrated in the experiment section.

segmentation, image matching between key frames extracted from slide-level video segments and slide images converted from PowerPoint slides is preformed (Figure 3.5). Most key frames extracted from slide-level video segments have borders and/or overlaid presenter images. Unlike slide-level segmentation that works on frames that are captured using the same stationary camera, image matching with local color histogram difference cannot give satisfying results. Thus, image matching reported here is accomplished through image edge detection and analysis.

To map segmentation results of TWI back to video

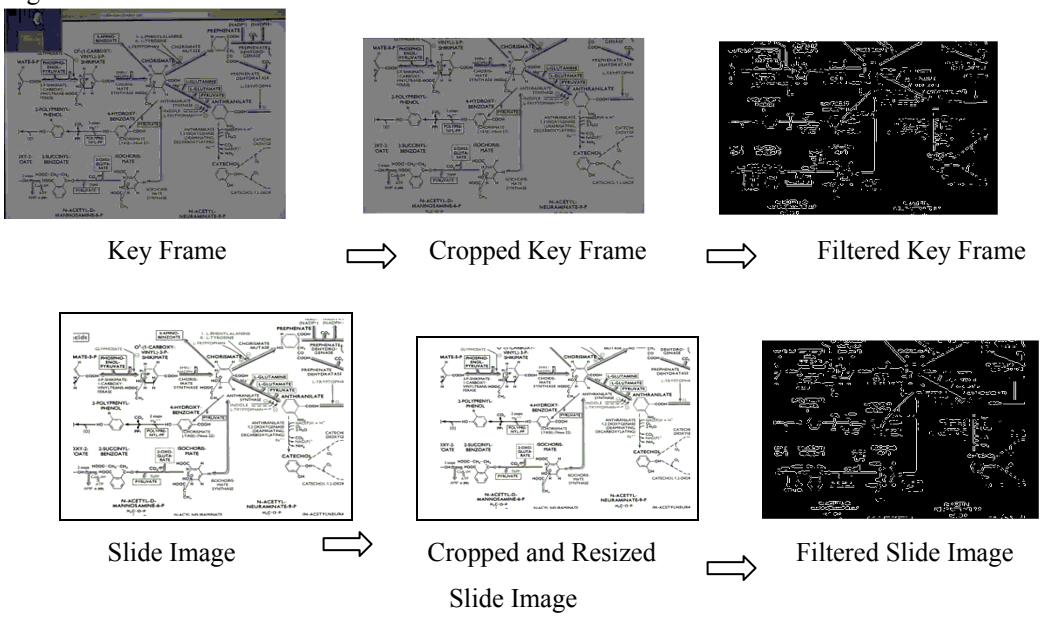


Figure 3.5. Image matching.

The first step in image matching is to align extracted key frames with converted slide images. We first crop key frames and slide images. Since all frames are captured with the same stationary camera, the clipping factors only need to be determined once per presentation. Then we resize the cropped slide image to the same size as the cropped key frames, or vice versa. Bilinear interpolation is applied in this process.

The next step is to extract edge information of both key frames and slide images. There are many ways to perform edge detection. In this paper, we apply Sobel filter on both images. The Sobel method is a gradient method and it finds edges using the Sobel approximation to the first derivative. It returns edges at those points where the gradient is maximum.

Based on work in (Mukhopadhyay, et al., 1999), the difference between a filtered key frame and a filter slide image is then computed as follows:

Given Sobel-filtered key frame $f1$ and Sobel-filtered slide image $s1$, let $b1$ be the number of black pixels in $f1$, $d1$ be the number of black pixels in $f1$ whose corresponding pixel in $s1$ is not black, $b2$ be the number of black pixels in $s1$, and $d2$ be the number of black pixels in $s1$ whose corresponding pixel in $f1$ is not black, then the difference Δ is defined as

$$\Delta = \frac{d1 + d2}{b1 + b2} \qquad (3.2)$$

The pair with the smallest Δ is considered as a

matching pair. When multiple key frames extracted from adjacent video segments match the same slide image, their corresponding segments are combined.

Image matching adds timing information to each slide. Based on this timing information, slide blocks from TWI are associated with topic-level video segments. Moreover, the relationship between slide-level and topic-level video segments can also be inferred.

Formally, given a presentation p, its slide video stream v, and a set of n slides, let $VS_{slide-level} = \{vs_{0,slide-level}, vs_{1,slide-level}, \ldots, vs_{m,slide-level}\}$ be a set of video segments generated from slide-level segmentation, $SB = \{sb_0, sb_1, \ldots, sb_k\}$ be a set of slide blocks produced from Topic Words Introduction, then, $VS_{topic-level} = \{vs_{0,topic-level}, vs_{1,topic-level}, \ldots, vs_{k,topic-level}\}$, where

$$vs_{i,topic-level} = \left\{ \begin{array}{l} vs_{j,slide-level} \mid 0 \le j \le m, \\ \text{and the projected} \\ \text{slides of } vs_{j,slide-level} \in sb_i \end{array} \right\} (0 \le i \le k)$$

Therefore, hierarchical segmentation of presentation videos is realized.

3.4. Evaluation

This section evaluates the performance of slide-level segmentation and topic-level segmentation. F-score is adopted for performance evaluation. It is defined as $F = \frac{2 \cdot P \cdot R}{P + R}$, where p is precision and R is recall. In TWI and slide-level segmentation,

$$p = \frac{number\ of\ correctly\ detected\ segments}{number\ of\ detected\ segments}$$

$$R = \frac{number\ of\ correctly\ detected\ segments}{number\ of\ true\ segments}$$

In image matching,

$$p = \frac{number\ of\ correctly\ matched\ key\ frames}{number\ of\ extracted\ key\ frames}$$

$$R = \frac{number\ of\ correctly\ matched\ key\ frames\ with\ combining}{number\ of\ slide\ images}$$

"with combining" here means that multiple key frames are treated as one if they are extracted from adjacent video segments and match the same slide image. The higher the F-score is, the better the performance is.

In slide-level segmentation, ten presentation videos from the 3rd Virtual Conference on Genomic and Bioinformatics (VCGB) are used (Table 3.1). If two video segments display the same slide, then they are counted as one correctly detected segment. Though slide video streams are captured with stationary cameras, different lighting conditions and accidentally overlaid images do affect the performance. Slide transitions can go undetected if adjacent slides have similar content layout and color distribution

.

Table 3.1. Experimental results of slide-level segmentation.

No.	Video length	No. of true seg.	No. of detected seg.	No. of correctly detected seg.	P	R	F
1	11 m 49 s	11	15	11	0.73	1.00	0.84
2	52 m 6 s	47	50	46	0.92	0.98	0.94
3	55m 49 s	19	19	18	0.94	0.94	0.94
4	50 m 46 s	58	58	54	0.93	0.93	0.93
5	24 m 13 s	22	18	18	1.00	0.82	0.90
6	60 m 0 s	50	60	50	0.83	1.00	0.91
7	59 m 43 s	73	63	63	1.00	0.86	0.92
8	22 m 13 s	27	22	22	1.00	0.81	0.90
9	28 m 46 s	39	39	34	0.87	0.87	0.87
10	22 m 0 s	16	16	16	1.00	1.00	1.00
Ave.							0.92

In topic-level segmentation, nine conference presentations are used. Out of the nine presentations, three are from the 3rd VCGB, one from SPIE AeroSense 2001, three from the 9th CAA Conference 2005, and the other two from other conferences. In spite of differences in subjects and formats, all the presentations have the same basic structure as described in Section 3. In this experiment, threshold *T*1 (Figure 3.4) is set as $m-d$ where m is the mean lexical scores of adjacent content slides and d is the corresponding standard deviation.

In our study, we find most presenters have a strong tendency to clearly restate a topic before they start it. Moreover, they use terms that are the same as or very similar to what they have in the outline/overview slides. Thus, topic-level segmentation achieves an average F-score of 0.97 (Table 3.2). Uses of acronyms affect the segmentation performance if an acronym is not properly introduced. For example, use "Support Vector Machine" only in outline/overview slide and "SVM" only later in content slides.

Table 3.2. Experimental results of Topic Words Introduction algorithm

.

No.	No. of slides	No. of True seg.	No. of detected seg.	P	R	F
1	76	3	3	1.00	1.00	1.00
2	48	3	3	1.00	1.00	1.00
3	28	5	5	1.00	1.00	1.00
4	17	7	9	0.78	1.00	0.88
5	21	4	4	1.00	1.00	1.00
6	36	4	4	0.75	1.00	0.85
7	37	6	6	1.00	1.00	1.00
8	32	9	9	1.00	1.00	1.00
9	32	6	6	1.00	1.00	1.00
Ave.						0.97

In image matching (Table 3.3), key frames are extracted from slide-level video segments and slides images are converted from a corresponding PowerPoint slide file. As discussed in image matching, image matching bases on edge detection and comparison. In effect, this matching method compares shapes of image components including text, graphics, tables, and so on. Therefore, inaccurate cropping factors affect the performance of image matching. In addition, if a slide is skipped during presentation, then there will be no matching video content, and no matching key frame exists. However, the current matching method still returns the most closely matched key frame. This will affect the image matching performance. To solve this problem, a threshold method should be investigated in future work. If the difference Δ is greater than the predefined threshold, then it is very unlikely that the extracted key frame and the converted slide image is a matching pair, thus the problem of slide skipping during presentations can be solved.

Table 3.3. Experimental results for image matching.

No.	No. of slides	No. of key Frames	No. of correct matching w/o combining	No. of correct matching w/ combining	P	R	F
1	11	15	14	10	0.93	0.90	0.91
2	47	50	47	44	0.94	0.93	0.93
3	19	19	19	18	1.00	0.94	0.97
4	58	58	54	54	0.93	0.93	0.93
5	22	18	18	18	1.00	0.82	0.90
6	50	60	60	50	1.00	1.00	1.00
7	73	63	60	60	0.95	0.82	0.88
8	27	22	22	22	1.00	0.81	0.90
9	39	39	39	34	1.00	0.87	0.93
10	16	16	16	16	1.00	1.00	1.00
Ave.							0.94

4. MULI-ONTOLOGY BASED VIDEO ANNOTATION

After a video is segmented, video annotation data can be extracted from the video and its segments. In this section, we propose a multi-ontology based multimedia annotation model in which a domain-independent multimedia ontology is integrated with multiple domain ontologies in an effort to better address different users' information needs. We first describe the process of ontology development and then introduce the strategy to integrate the domain-independent multimedia ontology with multiple domain ontologies. A term extraction procedure is proposed as a mechanism to extract domain-specific annotations.

4.1. Developing Ontology

To realize multi-ontology based multimedia, the first step is to develop ontology. Two types of ontologies are involved: a domain-independent multimedia ontology and domain ontologies.

Multimedia ontologies describe multimedia entities, structure, and content that are shared by all domains. Several multimedia metadata standards have been proposed in the literature (Martinez, 2004; n.d.; The Dublin Core Metadata Initiative, n.d.; Isaac & Troncy, 2004). MPEG-7 (Martinez, 2004), developed by the Moving Picture Expert Group (MPEG), is one of the most widely accepted standards for multimedia content description. MPEG-7 provides a rich set of description tools to describe multimedia assets from various aspects, such as content generation, content description, content management, navigation and access, user interaction, and so on. Several multimedia ontologies have been developed based on the MPEG-7 standards (Hunter, 2001; Tsinaraki et al., 2004; Garcia et al., 2005). Hunter's ontology is the first MPEG-7 ontology and it covers the upper part of the Multimedia Description Scheme (MDS) part of the MPEG-7 standard. Starting from the ontology developed by Hunter, Tsinaraki's ontology covers the full MDS part of the MPEG-7 standard. Compared to the previous ones, Garcia et al. developed the most complete MPEG-7 ontology. The proposed Multimedia Ontology (MO) here is based on MPEG-7 standards but focuses on the aspect of content description.

Three steps are followed to develop MO. First, we identify classes of the ontology. In general, classes describe concepts in the domain and are the focus of most ontologies. There are three types of classes in the proposed multimedia ontology (Figure 4.1): multimedia entities, non-multimedia entities, and descriptor entities. Multimedia entities are further classified into image, video, audio, audiovisual, and multimedia. Non-multimedia entities include agent, place, time, and instrument. Descriptor entities include visual descriptors, audio descriptors, structure descriptors, and semantic descriptors. In Figure 4.1, "Multimedia" refers to composite information that combines other multimedia elements such as image, audio and video. "MultimediaSegment" describes a segment of such media. Figure 4.1 gives the big picture of the MO, and some of the classes are not shown due to the limited space. Classes can have subclasses, for example, video segment is a subclass of Video. The subclass/superclass relationship may go several levels deep depending on the domain. In Figure 4.1, all arrows are labeled with "subClassOf," which depicts this relationship.

.

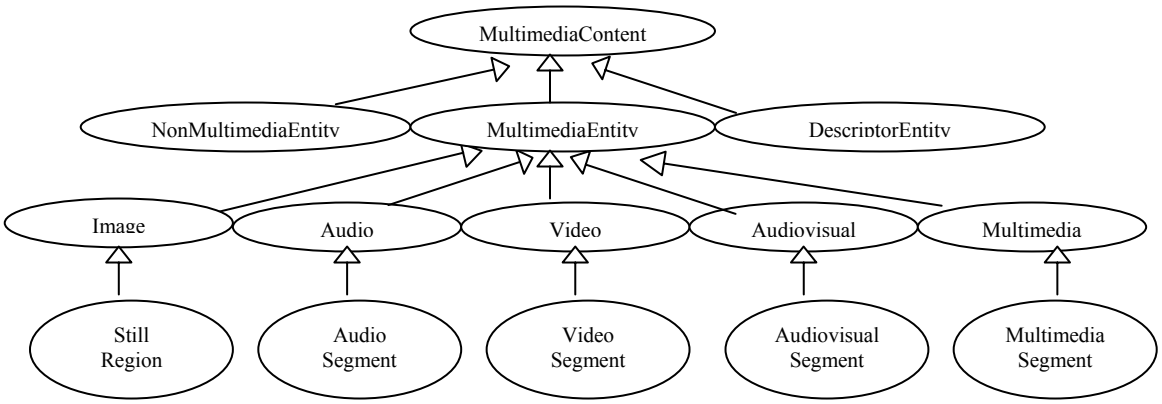


Figure 4.1. The big picture of the multimedia ontology.

We then arrange all classes in a hierarchy. This concept hierarchy describes various relationships among classes, for example, multimedia entities are disjoint with non-multimedia entities and descriptor entities, and video segment is a subclass of video.

The last step is to define properties for each class. These properties further define the permitted relationships among multimedia entities, descriptor entities, and non-multimedia entities. Figure 4.2 shows an example of one video segment property. In Figure 4.2, "hasDominantColor" is a video segment property. This property correlates "VideoSegment" class, a multimedia entity, with "DominantColor" class, a descriptor entity. "DominantColor" is a subclass of "Color" and "Color" is a subclass of "VisualDescriptor." By following this subclass chain, "VideoSegment" class is further related to "Color" class and "VisualDescriptor" class, both of which are descriptor entities. Properties can also be viewed as links among individuals from domain and individuals from range. Properties can have sub properties and each property can have multiple constraints. More details about general ontology development can be found in this paper (Noy & McGuinness, 2001).

Figure 4.2. Defining properties.

Domain ontologies can be either adopted or developed from the scratch. Each domain ontology defines domain concepts, concept properties, and concept relationships that are specific to that domain. These concepts and concept properties, called ontological terms in our discussion, form the controlled vocabulary of that domain ontology.

4.2. Integrating the Multimedia Ontology (MO) with Domain Ontologies

To integrate MO with a domain ontology, we use controlled vocabulary of that domain ontology to annotate multimedia content. Specifically, the ontological terms from a domain ontology are added as properties to instances of multimedia entities at different levels, which allows us to annotate multimedia content with domain-specific concepts at different levels.

Figure 4.3 illustrates the basic idea at ontology structure level. Again, only a small portion is displayed here. In Figure 4.3, "VideoSegment" is a multimedia entity. Three of its properties defined in MO are listed, i.e., "hasStartTime," "hasAbstract," and "hasDominantColor." Data Mining Ontology (DMO) and Gene Ontology (GO) are two domain ontologies that are integrated with MO.

"hasDMOAnnotation" and "hasGOAnnotation," from DMO and GO respectively, are added to "VideoSegment" class as two properties. "hasDMOAnnotation" annotates instances of "VideoSegment" with ontological terms from DMO, while "hasGOAnnotation" annotates instances of "VideoSegment" with ontological terms from GO.

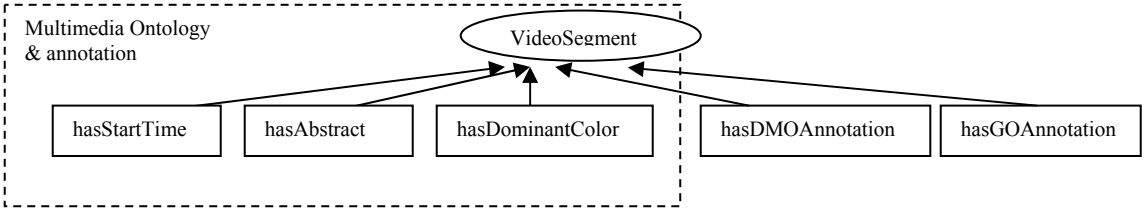


Figure 4.3. Integrating domain ontologies with MO.

MO can integrate with multiple domain ontologies. The relationship between them is one-to-many. In the case of a new ontology joining the system, properties are added to multimedia entities at the right level. This process does not affect other parts of the system.

Since one instance of MO can be annotated with multiple ontological terms of a given domain and one ontological term of a given domain can annotate multiple instances of MO, the relationship between instances of MO and ontological terms of a given domain is many-to-many. The cardinality of the relationship modeled in Figure 4.3 is one-to-one, i.e., one instance of "VideoSegment" is annotated with one and only one ontological term from GO, that is "hasGOAnnotation" is a single-value property. It is a simplified version. The many-to-many relationship described above can be modeled using intermediate relations.

To realize this integration strategy, i.e., adding ontological terms of a domain ontology to instances of MO, we must address the issue of how to automatically annotate multimedia entities with ontological terms of a specific domain ontology. To annotate multimedia with semantic concepts, most approaches in the literature are model-based. Various statistical models are built and used as semantic concept detectors. This approach works well when it is easy to build statistical models and the number of possible concepts is small. In this dissertation, we present a term extraction procedure (Figure 4.4) that can be used to automatically extract ontological terms from multimedia textual resources for situations where it is very difficult to build statistical models, such as conference presentations, and/or the number of terms is so big that it is infeasible to build concept detectors for all possible concepts.

1. Initialization
2. Let $W := \{w_1, w_2, ..., w_n\}$;; W is a word sequence of length n, and w_i, $1 \le i \le n$, is a single word.
3. Let $T :=$ {ontological terms}
4. Let $C := \phi$;; for extracted ontological terms.
5. Let start := 1 ;; start with the first word.
6. while (start $\le n$)
7. Let step := 0 ;; variable for term expansion.
8. while (start + step $\le n$ and $T.size > 1$)
9. pattern := $(\backslash w*[\backslash s-]) + w_{start}([\backslash s-] \backslash w*) + w_{start+1}...w_{start+step}([\backslash s-] \backslash w*) +$
10. Let $T' :=$ matchPattern(pattern, T)
11. if ($T'.size >= 1$)
12. $T := T'$
13. step := step + 1
14. end of while
15. if (step >= 1)
16. add terms in T but not already in C to C
17. start := start + step
18. end of while

Figure 4.4. The term extraction algorithm.

During the development of the term extraction procedure, we realize that professionals do not talk or write with ontologies in mind. Therefore, it is very rare to find exact ontological terms in their writings or talks. The main idea of the term extraction procedure is to find the longest sub-word sequences in input text that partially matches ontological terms. To this end, we utilize regular expression pattern matching in the procedure. Especially, Java regular expression standards[2] are followed. The expression \w matches a word character: [a-zA-Z_0-9], \s a white space character, * zero or more times, and + one or more times. Parentheses are used to group expressions. Before term extraction, uninformative words and punctuations are removed from input text. The detailed procedure is explained as follows.

Given an input word sequence $W := \{w_1, w_2, ..., w_n\}$, the objective is to extract all ontological terms from W and store them in collection C. The procedure starts with initialization (line 1-5). For each word in input text, first design the matching pattern (line 9). Then match the pattern against all ontological terms (line 10). If one or more ontological terms are found to match the pattern, then expand the word or word sequence with the word that follows (line 11-13). Continue doing this expansion until the text input is exhausted or no more ontological terms can be found (line 8). Store extracted ontological terms in collection C. The pattern definition in line 9 finds ontological terms that contain every word of the word sequence $\{w_{start}, w_{start+1}, \ldots, w_{start+step}\}$ in specified order. But sub-sequences are not necessarily continuous sub-strings of ontological terms. For example, "ribosomal large subunit assembly and maintenance" is a matched ontological term for the text input "ribosomal assembly." We apply the above term extraction algorithm with ontological terms from different domain ontologies and, thus, get different domain-specific annotations for the same multimedia content.

With this multi-ontology based multimedia annotation, different sets of annotation data are used in information retrieval. If a user is a biologist, then GO-based annotation is used; if a user is a computer scientist, then DMO-based annotation is used. As a result, multimedia information retrieval can be tailored towards different users' information needs.

[2] http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

5. ONTOLOGY-DRIVEN VIDEO ACCESS

After videos are segmented and annotation data are extracted, this section introduces ontology-driven video access. The key idea is to integrate ontologies into video browsing, searching, and filtering. The goal is to increase video retrieval relevancy and enhance users' video access experiences.

Ontology-driven video access works on video annotation data and refines or generalizes user queries with relevant domain concepts extracted from domain ontologies. To extend ontology-driven video access to external heterogeneous data sources, web services are explored.

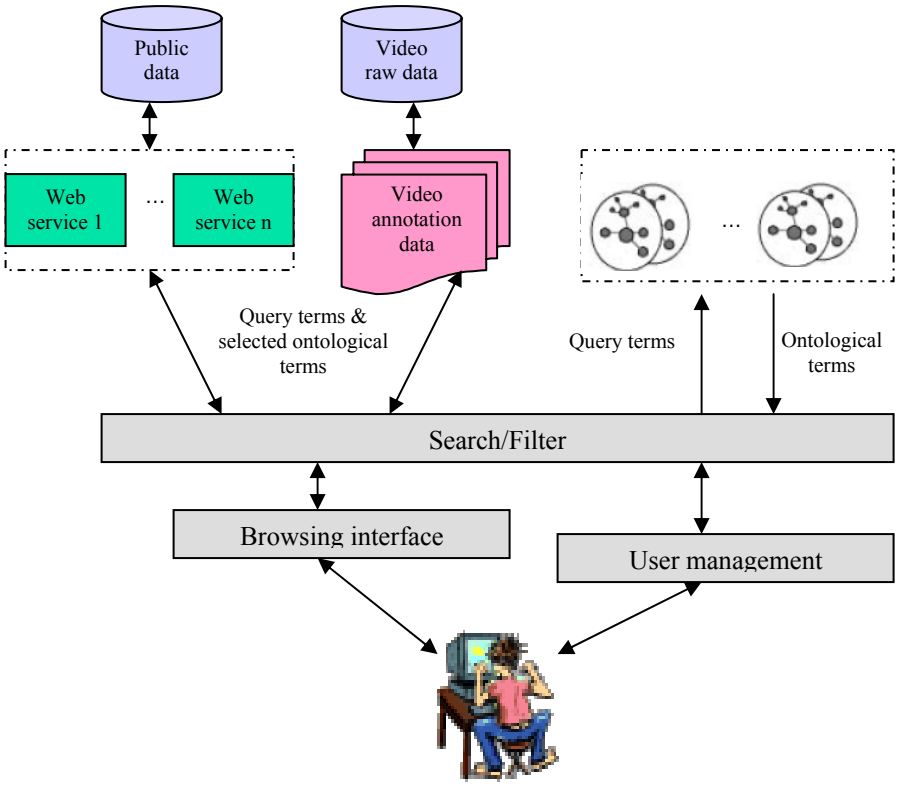


Figure 5.1. The overview of ontology-driven video access.

Given query terms, the system first sends these query terms to a selected domain ontology, and retrieves relevant ontological terms/concepts based on the relationships embedded in that domain ontology. After that, both query terms and extracted ontological terms/concepts are fed into the selected set of video annotation data and web services, which then return relevant materials from both internal video collections and external data sources that are publicly available. Based on the proposed architecture, one single search pulls out all the relevant material both internal and external, which simplifies and reduces work on the users' side.

The purpose of user management in Figure 5.1 is to manage user profiles. By knowing learning habits or access preferences of users, there is a better chance that a video system can present relevant information of more interest to users.

6. EXPERIMENT

To experience ontology-driven video annotation and access proposed, we use VCGB as our test bed and build a Ontology-driven Video Access Platform for Virtual Conferences (OVAP) (http://webdb2.cs.ndsu.nodak.edu/multimedia/).
This system is proposed in the broad context of virtual learning/researching environments, such as virtual conferences, virtual seminars, and virtual classrooms. Virtual learning/researching environments overcome geographical and economical limitations, and enable students and

researchers alike to learn new technologies, participate in high quality meetings, and share research ideas easily.

6.1. Multi-ontology Video Annotation

We use the Multimedia Ontology (MO) developed in Section 4.2, Gene Ontology (GO) (http://www.geneontology.org/) and an experimental Data Mining Ontology (DMO) to demonstrate multi-ontology based multimedia annotation. GO provides controlled vocabularies for describing gene products in terms of their biological process, molecular function, and location in a cellular component. The standardized GO terms facilitate the annotation of gene products and allow for uniform queries to be performed across different scientific databases. The DMO is developed from the scratch, following the same process as described in Section 4.2. Figure 6.1 illustrates this ontology structure with a high-level view. All the arrows in Figure 6.1 are labeled with "subClassOf," which depicts subclass/superclass relationships. Details on classification and ARM (Association Rule Mining) are not shown.

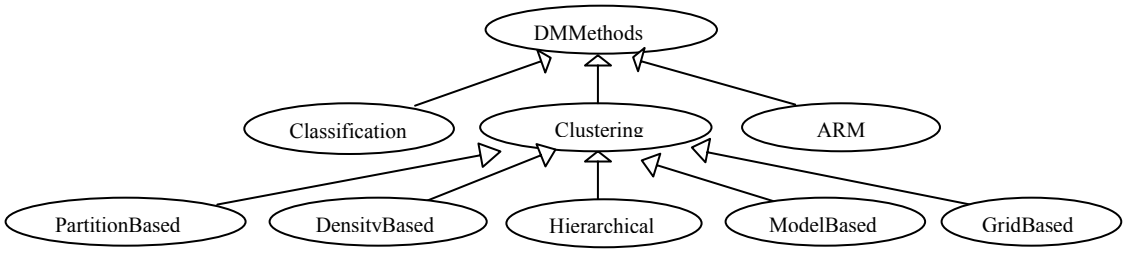


Figure 6.1. The big picture of data mining ontology.

Both MO and DMO are developed using the Protégé 3.1.1 ontology editor (http://protege.stanford.edu/). Due to limited query facilities and non-web query interface of Protege, we convert MO and DMO into database schemas from their ontology XML outputs. As for GO, we use its MySQL format downloaded from http://www.geneontology.org/. To realize the integration of MO with GO and DMO, two intermediate tables are created to model the many-to-many relationships existing between instances of MO and ontological terms from GO and DMO respectively as described in Section 4.2.

After setting up ontologies and their storage structures, the next logical step is to extract MO-based, GO-based, and DMO-based annotations. To perform ontology-driven video annotation, the first step is video segmentation. The segmentation of presentation videos uses the exact same procedure as described in Section 3. Slide-level segmentation operates on slide video streams, while topic-level segmentation makes use of extracted slide text. At the end, slide-level segmentation creates a sequence of slide-level video segments. Within each such segment, the projected slide image does not change. Topic-level segmentation generates a sequence of topic-level video segments, each of which covers one or more slides. Within each such segment, the topic does not change. Since the presenter video stream and the slide video stream have the same presentation timeline, the presenter video stream is segmented as well based on this temporal relationship.

To extract annotation data from a presentation video and its segments, we apply multi-ontology based multimedia annotation as discussed in Section 4. Three levels of annotation data are extracted: presentation-level, topic-level, and slide-level. For MO-based annotation, at the presentation-level, words or terms in presentation titles are used; at the topic-level, words or terms in topic vectors are used; at the slide-level, words or terms in content vectors are used. For GO-based and DMO-based annotation, the term extraction algorithm introduced in Section 4.2 is applied on presentation titles, outline/overview slides, and content slides. Besides these data, other annotation, such as presenter information, presentation durations, video segment start time and end time, key frames, and so on, are all stored in the annotation database.

Before term extraction, uninformative words and punctuations are removed from the extracted slide text. To find uninformative words in input text for GO-based annotation, we perform a word frequency study on GO terms. Gene Ontology of version January 2005 has 19,455 terms and more

than 70,000 words. Some words, such as activity (6657), regulation (1939), biosynthesis (1084), occur much more often than others and are uninformative words to the domain. Based on the word frequency analysis result of GO, we pick those terms with very high frequency and combine them with a common stop-word list, and use the combined list to remove uninformative words from input text. As for DMO-based annotation, we use the common stop-word list only.

6.2. Ontology-driven Video Access

Typical video access involves browsing and searching. To facilitate browsing, OVAP provides links for abstract, full paper, PowerPoint slide, video summary with key frames, and whole presentation for each virtual presentation (if available) in a hierarchical manner according to the ease of access. Regarding the search operation, one search pulls out relevant documents regardless of the format and sources. Search results are presented in multiple levels. In addition, relevant documents from PubMed (Sayers & Wheeler, n.d.) and Google (Google Inc., n.d.) are dynamically extracted with the corresponding web services. Figure 6.2 describes the general search process.

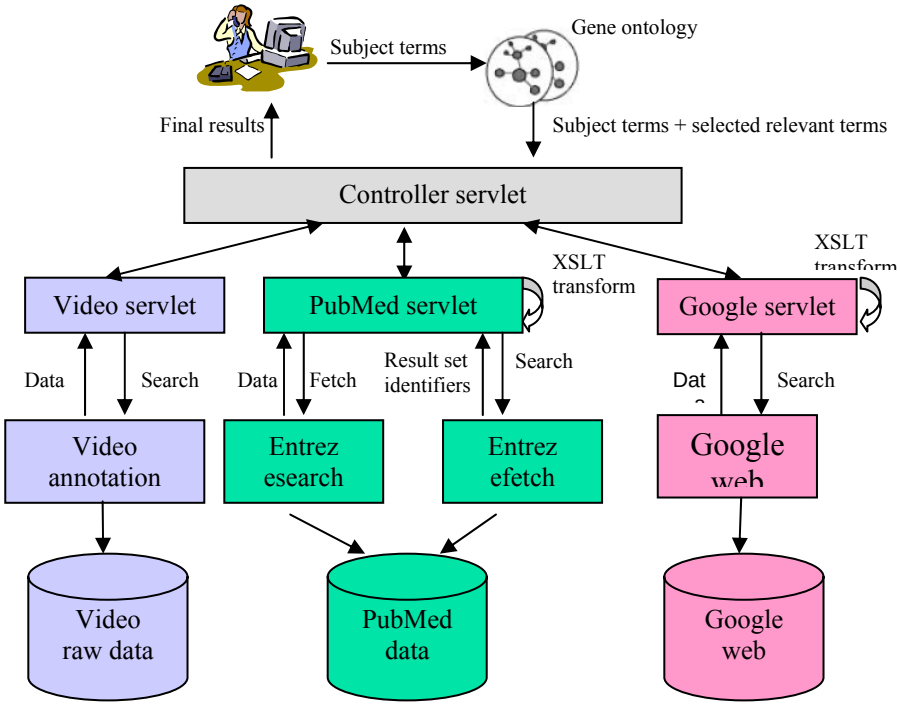


Figure 6.2. The ontology-driven search.

The following interactive steps describe this process:

- A user enters query terms; the application sends them to Gene Ontology.
- The application extracts relevant GO terms from Gene Ontology and displays them to the user an ontology-based browsing space.
- The user selects GO terms of interest, and the application feeds both query terms and relevant GO terms to the controller servlet. Servlet here refers to the software agent developed in Java.
- The controller servlet sends all these terms to VCGB servlet, PubMed servlet, and Google servlet to extract relevant materials from the VCGB video collection, PubMed literature collection, and Google web sources, respectively.

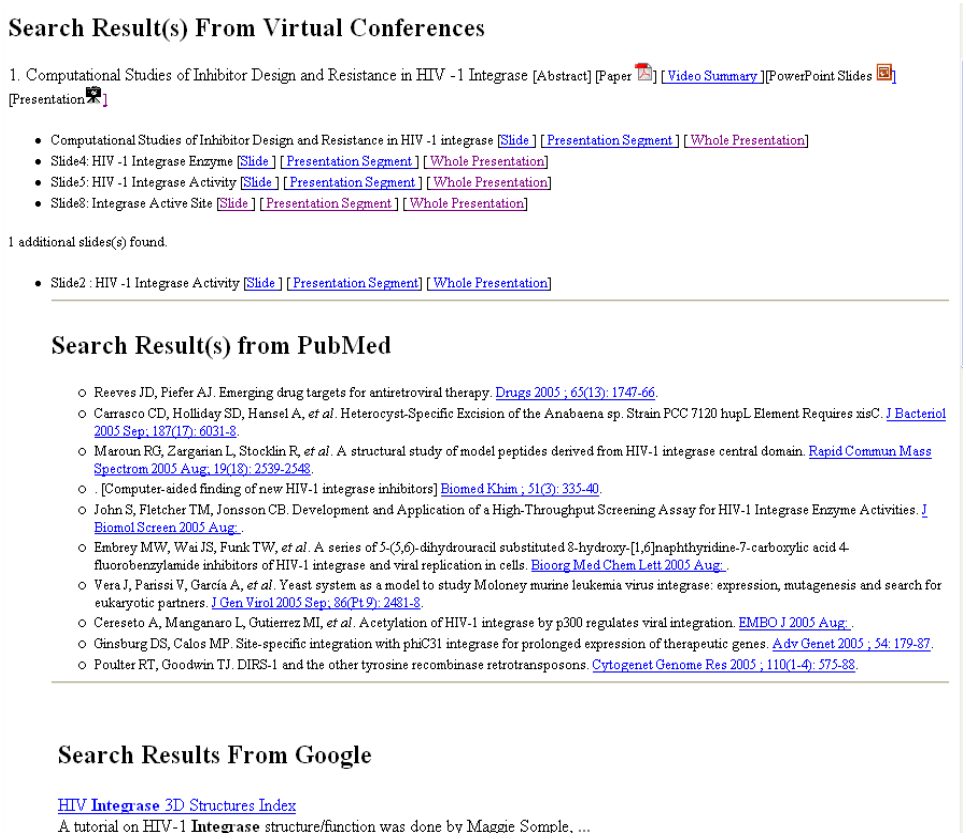
In the VCGB branch, GO-based annotation data are used. As indicated in Section 6.1, GO-based annotation data have three levels: presentation-level, topic-level, and slide-level. The servlet first searches presentation-level annotation data. If there is a match, it searches topic-level and slide-level annotations of that presentation. Slide-level video segments with matching topic-level annotations are searched before those with no matching topic-level

annotations. Then, the servlet searches the rest slide-level video segments with no matching presentation titles. Finally, the VCGB servlet organizes video data using SMIL (W3C, n.d.), RealText, and RealPix (RealNetworks, Inc., n.d.), links it to an HTML page and returns that HTML page to the controller servlet.

In PubMed servlet, the servlet first calls the *esearch* utility of PubMed web service by sending out an HTTP GET request with a URL containing all required parameters, *esearch* returns XML data that contain result set identifiers. PubMed servlet then extracts QueryKey and WebEnv from the XML data and sends out another HTTP GET request with a URL containing all required parameters. This request calls *efetch* utility, and *efetch* returns data to PubMed servlet. PubMed servlet transforms the results to HTML using XSL and returns them to the controller servlet.

In Google servlet, the servlet executes *doSearch* with a Google license key and other query parameters. Google Web APIs sends back query results with structured data format. Google servlet then converts structured data to XML and transforms XML data to HTML using XSL. Finally, it returns HTML results to the controller servlet.

- The controller servlet compiles the results from the three servlets and sends back the final result as an HTML page to the user/ program. Figure 6.3 is an example of HTML pages sent back to users.



Search Result(s) From Virtual Conferences

1. Computational Studies of Inhibitor Design and Resistance in HIV -1 Integrase [Abstract] [Paper] [Video Summary][PowerPoint Slides] [Presentation]

- Computational Studies of Inhibitor Design and Resistance in HIV -1 integrase [Slide] [Presentation Segment] [Whole Presentation]
- Slide4: HIV -1 Integrase Enzyme [Slide] [Presentation Segment] [Whole Presentation]
- Slide5: HIV -1 Integrase Activity [Slide] [Presentation Segment] [Whole Presentation]
- Slide8: Integrase Active Site [Slide] [Presentation Segment] [Whole Presentation]

1 additional slides(s) found.

- Slide2 : HIV -1 Integrase Activity [Slide] [Presentation Segment] [Whole Presentation]

Search Result(s) from PubMed

- Reeves JD, Piefer AJ. Emerging drug targets for antiretroviral therapy. Drugs 2005 ; 65(13): 1747-66.
- Carrasco CD, Holliday SD, Hansel A, *et al*. Heterocyst-Specific Excision of the Anabaena sp. Strain PCC 7120 hupL Element Requires xisC. J Bacteriol 2005 Sep; 187(17): 6031-8.
- Maroun RG, Zargarian L, Stocklin R, *et al*. A structural study of model peptides derived from HIV-1 integrase central domain. Rapid Commun Mass Spectrom 2005 Aug; 19(18): 2539-2548.
- . [Computer-aided finding of new HIV-1 integrase inhibitors] Biomed Khim ; 51(3): 335-40.
- John S, Fletcher TM, Jonsson CB. Development and Application of a High-Throughput Screening Assay for HIV-1 Integrase Enzyme Activities. J Biomol Screen 2005 Aug; .
- Embrey MW, Wai JS, Funk TW, *et al*. A series of 5-(5,6)-dihydrouracil substituted 8-hydroxy-[1,6]naphthyridine-7-carboxylic acid 4-fluorobenzylamide inhibitors of HIV-1 integrase and viral replication in cells. Bioorg Med Chem Lett 2005 Aug; .
- Vera J, Parissi V, García A, *et al*. Yeast system as a model to study Moloney murine leukemia virus integrase: expression, mutagenesis and search for eukaryotic partners. J Gen Virol 2005 Sep; 86(Pt 9): 2481-8.
- Cereseto A, Manganaro L, Gutierrez MI, *et al*. Acetylation of HIV-1 integrase by p300 regulates viral integration. EMBO J 2005 Aug; .
- Ginsburg DS, Calos MP. Site-specific integration with phiC31 integrase for prolonged expression of therapeutic genes. Adv Genet 2005 ; 54: 179-87.
- Poulter RT, Goodwin TJ. DIRS-1 and the other tyrosine recombinase retrotransposons. Cytogenet Genome Res 2005 ; 110(1-4): 575-88.

Search Results From Google

HIV **Integrase** 3D Structures Index
A tutorial on HIV-1 **Integrase** structure/function was done by Maggie Sample, ...

Figure 6.3. Search result presentation.

6.3. Cc

In the last decade, several systems have been developed to automatically record a live presentation and produce rich-media presentation documents for online or offline usage (Brotherton & Abowd, 2004; Bianchi, 1998; Rowe, Harley, Pletcher, & Lawrence, 2001; Müller & Ottmann, 2000). Among these systems, the eClass project and the Authoring on the Fly (AOF) system are closely related to our system (OVAP). They target the education domain and focus on presentation videos. In this section, we compare

OVAP with these systems (Table 6.1).

Table 6.1. Comparison of OVAP with related projects

Present video content in multiple levels	
The eClass project	The content of each lecture is presented at two levels: the whole lecture and its slide-level segments.
The AOF system	No explicit hierarchy exists, but Random Visible Scrolling enables the user to scroll forward and backward at any speed, much like window scrolling.
The OVAP	The content of each presentation is shown at three levels: the whole presentation, the topic-level segments, and the slide-level segments.
Provide rich Annotation	
The eClass project	• For each course, the course title, its instructor, and links to course home page are provided. For each lecture, the lecture date and the lecture title are given. • Time stamps for slide transitions, handwritten annotations, and URLs of web pages visited during a lecture are captured and stored.
The AOF system	• Keywords are extracted from slide text. • To facilitate replay of recorded lectures, synchronization related information (e.g., the time stamps of slide-level video segments and whiteboard actions) is stored.
The OVAP	• For each presentation, the title, the presenter information, and links to the abstract, the PPT slides, and the video are stored. For each topic-level segment, the topic summary is stored. For each slide-level segment, the slide title and slide text are stored. • To facilitate search and retrieval, presentation dates and duration, start time and end time of video segments at different levels, extracted key frames, domain-specific annotation data, etc. are all stored in annotation database.
Support flexible search	
The eClass project	• Search can be performed at multiple levels and it mainly operates on slide text. • Search results are presented at two levels: lectures and slide-level segments. • The existing search system is based on SQL queries of the database without indexing and weighting. The issue of retrieval relevancy is not sufficiently addressed.
The AOF system	• Search is usually performed over all the lectures, and it mainly operates on slide text. • Only slide-level segments are returned. • The retrieval relevancy is addressed with heuristic rules.
The OVAP	• Search is performed at multiple levels: the entire data collection, individual presentations, topic-level segments, and slide-level segments. • Search results are presented hierarchically with presentations followed by topic-level segments and slide-level segments. • Multiple ontologies have been integrated into the system to improve retrieval relevancy.
Extract relevant information from external data sources	
The eClass project	This aspect is not addressed in this system.
The AOF system	This aspect is not addressed in this system.
The OVAP	Web services are exploited to extract relevant information from heterogeneous external data sources.

Table 6.1 shows that there are three major differences between our system, i.e., OVAP, and the other two projects. First, OVAP integrates ontologies into video access. An ontology describes concepts and their relationships in a formal way. By exploring these relationships, domain-specific relevant concepts can be extracted. Since this relevancy information is extracted directly from ontology where domain knowledge is embedded, there is a potential to increase the degree of video retrieval relevancy (Hollink, 2005; HyvÄonen, et.al., 2003; Khan, 2000; Schreiber, 2004). Second, OVAP supports video access with finer granularity. It supports video access of a presentation at three

levels, i.e., presentation-level, topic-level, and slide-level. The added topic-level access provides users with further flexibility. Lastly, OVAP employs web services to extract relevant information from heterogeneous data sources. This is significant in e-learning systems where relevant information can advance the understanding of problems and the acquisition of knowledge and skills.

7. CONCLUSIONS AND FUTURE WORK

The explosive growth of video data demands efficient and flexible access mechanisms. In this paper, we propose an ontology-driven framework for video annotation and video access. The goal is to integrate ontology into video systems in an effort to improve users' video access experience.

We view ontology-driven video annotation as a two-step process: video segmentation, and video annotation data extraction and organization. In video segmentation, we propose multi-mode segmentation procedures for presentation videos. In this procedure, the semantic-rich textual modality is integrated with the visual modality.

To extract annotation data from videos and video segments, and organize them in a way that facilitates video access, we propose a multi-ontology based multimedia annotation model. In this model, a domain-independent multimedia ontology is integrated with multiple domain ontologies. The goal is to provide multiple, domain-specific views of the same multimedia content and thus meet different users' information needs.

With extracted annotation data, we propose ontology-driven video access. In ontology-driven video access, a user can select which ontology to interact with. The selection of ontology determines the set of annotation data and the group of relevant terms/concepts. As can be seen, ontology tailors the video access to users' domain-specific information access needs. To extend ontology-driven video to external heterogeneous data sources, web services are explored in this dissertation. Our experience shows that web service is an effective way to extract relevant documents from assorted, publicly available data sources.

In this paper, we focus our discussion on presentation videos. But the general concept of ontology-driven video annotation and access is applicable to many other areas as well, for example, digital libraries, the Web and corporate video collections. To improve the work and also extend the concept of ontology-driven to other fields, we identify the following areas for future work:

- To apply multi-ontology based multimedia annotation model on different types of multimedia assets, the issue of extracting domain-specific annotation need to be further addressed.
- Information-rich text modality is important in semantic segmentation of videos. We would like to integrate intelligent text analysis techniques that integrate natural language processing, machine learning, and artificial intelligence. We envision that such techniques will provide a viable solution to text-based segmentation.
- Ontology-driven video annotation and access incorporates ontology into video systems. To apply this concept on a large scale, other issues, such as redundant information across ontologies and external ontology inference engine integration, need to be further addressed.
- Formally evaluating a video access system is an important issue. Most approaches in the literature are survey-based. We would like to investigate other ways to assess our system in the future.

Video plays an important role in today's education. With the increasing growth of video data, we envision that there will be more extensive research conducted to effectively segment, annotate, and access these data, thus making them fully benefit the advance of society.

REFERENCES

[1]. Abowd, G. D., Brotherton, J. A., and Bhalodai, J. (1998). Classroom 2000: A system for capturing and accessing multimedia classroom experiences. *CHI '98 Demonstration Paper*, pp.20-21.

[2]. Bao, J., Cao, Y., Tavanapong, W., and Honavar, V. (2004). Integration of Domain-Specific and Domain-Independent Ontologies for Colonoscopy Video Database Annotation. *Proc. of 2004 International Conference on Information and Knowledge Engineering (IKE 04),* pp. 82-90.

[3]. Bianchi, M.(1998). AutoAuditorium: A fully automatic, multicamera system to televise auditorium presentations. *Joint DARPA/NIST Smart Spaces Tech. Workshop*, 1998.

[4]. Bishop, J.S., Spake, D.F. (2003). Distance

education: A bibliographic review for educational planners and policymakers 1992-2002. Journal of Planning Literature, Vol. 17, No. 3, pp. 372-391.

[5]. Brotherton, J. A. and Abowd, G.D. (2004). eClass: Assessing automated capture and access in the classroom. *ACM Trans. on CHI*, pp. 121 - 155.

[6]. Cox, I. J., Miller, M. L., Minka, T. P., & Yianilos, P.N. (1998). An optimized interaction strategy for Bayesian relevance feedback. *Proc. of IEEE conf. on Computer Vision and Pattern Recognition*, pp. 553-558.

[7]. Cox, I. J., Miller, M.L., Omohundro, S.M., and Yianilos, P.N. (1996). Target testing and the pichunter Bayesian multimedia retrieval system. *Advanced Digital Libraries Forum,* pp. 66-75.

[8]. Dunn, S. (2000). The virtualizing of education. *The Futurist*. 34(2): 34-38.

[9]. Fan, J., Luo, H., and Elmagarmid, A.K. (2004). Concept-oriented indexing of video databases: Toward semantic sensitive retrieval and browsing. *IEEE Trans. On Image Processing*, July 2004, pp. 974–992.

[10]. Flachsbart, J., Franklin, D., and Hammond, K. (2000). Improving human computer interaction in a classroom environment using comptuer vision. *Proc. of the 5th international conference on Intelligent user interfaces*, pp. 86-93.

[11]. Frydenberg, J. (2002). Quality standards in eLearning: A matrix of analysis. *The International Review of Research in Open and Distance Learning*, Vol 3, No 2. Available: http://www.irrodl.org/index.php/irrodl/article/view/109/189.

[12]. Garcia, R. and Celma, O. (2005). Semantic Integration and Retrieval of Multimedia Metadata. *Proc. of the 5th International Workshop on Knowledge Markup and Semantic Annotation*, November 2005.

[13]. Google Inc.(n.d.). Google SOAP Search APIs. Available: http://www.google.com/apis/api_faq.html.

[14]. Haubold, A. & Kender, J. R. (2003). Analysis and interface for instructional video. *Proceedings of International Conference on Multimedia and Expo*, pp. 705-708.

[15]. Hauptmann, A. G. (2004). Towards a large scale concept ontology for broadcast video. *Proc. of the 3rd Int. Conf on Image and Video Retrieval (CIVR'04)*, pp. 674-675.

[16]. He, L., Grudin, J., and Gupta, A. (2000). Designing presentations for on-demand viewing, *Proc. of CSCW'00 vhgnb,* pp.127-134.

[17]. Hearst, M. A. (1994). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, Vol. 23, No. 1, pp. 33-64.

[18]. Hollink, L., Worring, M., and Schreiber, G. (2005). Building a Visual Ontology for Video Retrieval. *Proc. of the ACM Multimedia*, pp. 479– 482.

[19]. Hunter, J.(2001). Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. The 1st International Semantic Web Working Symposium (SWWS'01), pp. 261-281.

[20]. HyvÄonen, E., Styrman, A., & Saarela, S. (2003). Ontology-based image retrieval. *Poster Proceedings, the 12th International World Wide Web Conference*.

[21]. Isaac, A., and Troncy, R.,(2004). Designing and Using an Audio-Visual Description Core Ontology. *Proc. of the 14th International Conference on Knowledge Engineering and Knowledge Management*, October 2004.

[22]. Kanedera , N., Sumida, A., Ikehata, T., and Funada, T. (2006). Subtopic segmentation in lecture speech for the creation of lecture video contents. *Syst Comp Jpn*, 37(10), pp.13-21.

[23]. Kariya, S. (2003). Online education expands and evolves. *IEEE Spectrum.* 40(5): 49-51.

[24]. Khan, L. & McLeod, D. (2000). Audio structuring and personalized retrieval using ontologies. *Proc. IEEE Advances in Digital Libraries (ADL2000)*, p. 116.

[25]. Liu, T. and Kender, J. R. (2002). Rule-based semantic summarization of instructional videos. *International Conference on Image Processing*, pp. 601-604.

[26]. Lin, M., Chau, M., Cao, J., & Nunamaker, J. F. (2005). Automated video segmentation for lecture videos. *The International Journal of Technology and Human Interaction* (*IJTHI*), Vol. 1, No. 2, pp. 27-45.

[27]. Rowe, L.A., Harley, D., Pletcher, P., and Lawrence, S. (2001). BIBS: A Lecture Webcasting System. *Center for*

Studies in Higher Education. Paper CSHE4-01. http://repositories.cdlib.org/cshe/CSHE4-01.

[28]. Martinez, J. M. (2004). MPEG-7 Overview. Available: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[29]. Minka, T. P. & Picard, R. W. (1997). Interactive learning with a 'society of models'," *Pattern Recognition*. Vol. 30, No. 4, pp. 565–581.

[30]. Mukhopadhyay, S. & Smith, B. (1999). Passive capture and structuring of lectures. *Proceedings of the 7th ACM International Conference on Multimedia*, pp. 477-487.

[31]. Müller, R. and Ottmann, T. (2000). The 'authoring on the fly' system for automated recording and replay of (Tele)presentations," *ACM/Springer Multimedia Systems Journal*, Vol. 8, No. 3, pp. 158-176, 2000.

[32]. Ngo, C., Wang, F., and Pong, T. (2003). Structuring lecture videos for distance learning applications. *ISMSE*, pp. 215-222.

[33]. Noy, N., McGuinness, D. (2001). Ontology development 101: A guide to creating your first ontology. *Technical Report SMI-2001-0880*, Stanford University School of Medicine.

[34]. Onishi, M., Izumi, M., and Fukunaga, K. (2000). Blackboard segmentation using video image of lecture and its applications. *International Conference on Pattern Recognition*, pp. 615–618.

[35]. Papathomas, T. V., Conway, T. E., Cox, I. J., Ghosn, J., Miller, M. L. , Minka, T. P., et al. (1998). Psychophysical studies of the performance of an image database retrieval system. *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology, Conference on Human Vision and Electronic Imaging III*, pp. 591-602.

[36]. Phung, D., Venkatesh, S., & Dorai, C. (2002). High level segmentation of instructional videos based on content density. *Proceedings of Multimedia '02,* pp. 295-298.

[37]. Picard, R. W. & Minka, T. P. (1995). Vision texture for annotation. *ACM Multimedia Systems*, 3 (1), pp.1-11.

[38]. Porter, M. (1980). An algorithm for suffix stripping. *Program*, Vol. 14, No. 3, pp.130-137.

[39]. RealNetworks, Inc.(n.d.). Real networks production guide. Available: http://service.real.com/help/library/encoders.html#productionguide.

[40]. Rui, Y., Gupta, A., Grudin, J., and He, L. (2004). Automating lecture capture and broadcast: technology and videography. *Multimedia Systems*, pp. 3–15.

[41]. Rui, Y., Huang, T. S., Mehrotra, S., & Ortega, M. (1998). Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transaction on Circuits and Systems and Video Technology*, Vol. 8, No. 5, pp. 644-655.

[42]. Sayers, E. and Wheeler, D. (n.d.). Building customized data pipelines using Entrez programming utilities (eUtils). Available: http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=coursework.chapter.eutils.

[43]. Schreiber, A. T., Dubbeldam, B., Wielemaker, J., & Wielinga, B.(2004). Ontology-based photo annotation. *IEEE Intelligent Systems*, Vol. 16, No. 3, pp. 66-74.

[44]. Smith, T., Ruocco, A., and Jansen, B. (1999). Digital video in education. *Proc. of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*, pp. 122-126.

[45]. The Dublin Core Metadata Initiative. (n.d.). Dublin Core metadata initiatives. Available: http://dublincore.org/.

[46]. Tsinaraki, C., Polydoros, P., and Christodoulakis, S. (2004). Interoperability support for Ontology-based Video Retrieval Applications. *Proc. of 3rd International Conference on Image and Video Retrieval*, pp. 582-591.

[47]. Yamanoto, N., Ogata, J., & Ariki, Y. (2003). Topic segmentation and retrieval systems for lecture videos based on spontaneous speech recognition. *Proc. of EUROSPEECH 2003,* pp. 961-964.

[48]. W3C(n.d.). Synchronized Multimedia Integration Language (SMIL 2.0). Available: http://www.w3.org/TR/SMIL2/.

[49]. Winsboro, I.D.S. (2002) Technology and distance learning lessons from the nation's newest university: Perceptions and reality. *The Educational Forum*. 66(3): 247-252.