Structuring Lecture Videos for Distance Learning Applications

Chong-Wah Ngo
Deparment of Computer Science
City University of Hong Kong
Tat Chee Avenue, Kowloon
cwngo@cs.cityu.edu.hk

Feng Wang & *Ting-Chuen Pong*
Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon
{wfeng,tcpong}@cs.ust.hk

Abstract

This paper presents an automatic and novel approach in structuring and indexing lecture videos for distance learning applications. By structuring video content, we can support both topic indexing and semantic querying of multimedia documents. In this paper, our aim is to link the discussion topics extracted from the electronic slides with their associated video and audio segments. Two major techniques in our proposed approach include video text analysis and speech recognition. Initially, a video is partitioned into shots based on slide transitions. For each shot, the embedded video texts are detected, reconstructed and segmented as high-resolution foreground texts for commercial OCR recognition. The recognized texts can then be matched with their associated slides for video indexing. Meanwhile, both phrases (title) and keywords (content) are also extracted from the electronic slides to spot the speech signals. The spotted phrases and keywords are further utilized as queries to retrieve the most similar slide for speech indexing.

1. Introduction

The teaching and learning in a traditional classroom can be viewed as a multimedia authoring activity. The main streams of activity include what is heard (audio), what is seen (video) and what is discussed (presentation slides) in a classroom. One essential goal of distance learning is to provide a quality of learning that is both compatible and comparable to the traditional classroom environment. To achieve this goal, a fundamental problem is how to effectively present and index the activities in the classrooms for on-line courses. A typical approach is to record and encode the activities in a classroom as multimedia documents such as in the audio and video formats [1, 4]. The multimedia documents, together with the associated electronic slides, could then be streamed over the network for on-line presentation.

Besides effective presentation, a more sophisticated way of distance learning is to support the semantic querying of multimedia documents [5, 11]. For instance, to provide the facility to support the querying of topics of interest from a set of lecture videos that have been recorded for a semester. This application typically requires the "linking indices" that can explicitly link every video segment with its associated electronic slide. By these linking indices, the problem of semantic querying can be turned into the traditional information retrieval problem. In other words, the remaining step is to index the keywords found in electronic slides in a database for retrieval. When a query is input, a set of relevant slides are retrieved. When a relevant slide is selected to view, the associated video segment will be displayed. The main challenge of this application, nevertheless, is the modeling of the relationship (or finding the linking indices) between the recorded multimedia documents and electronic slides.

Figure 1 presents our proposed framework for structuring lecture video content for effective indexing. The input to the framework is the presentation slides and a video that contains the visual and audio information. The outputs are the "linking indices" that link the relationship among them. Three major techniques involved are the detection of slide transitions, the analysis of textual information embedded in videos (refer to as video text) and the recognition of speech signal.

In the framework shown in Figure 1, initially, a video is partitioned into segments according to the topic of discussion. This is achieved by detecting the transition of slides through the analysis of video content. We refer each segment as a shot[1]. Frames in one shot normally capture a projected slide as their background scene. After detecting slide transitions, keyframes are extracted from each shot for video text analysis. Texts embedded in videos are useful cues for conjecturing the topics of discussion. Thus,

[1] A shot is originally referred to as a sequence of frames with continuous camera motion. In this paper, we refer a shot as a sequence of frames that capture a same electronic slide.

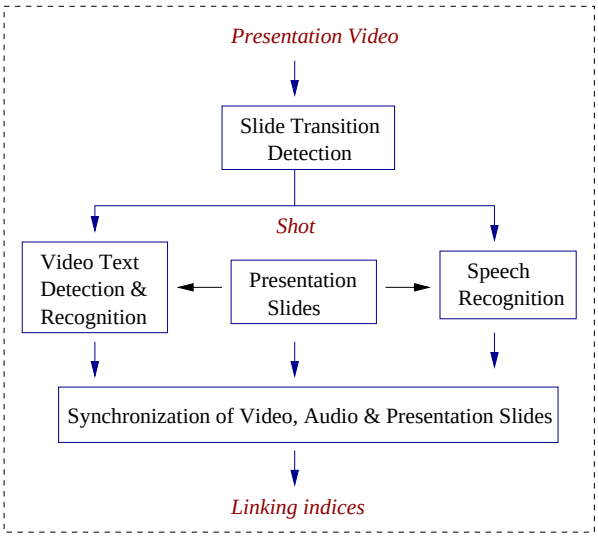

Figure 1. A framework for lecture video indexing

our task is to extract and segment the text lines in videos for OCR recognition. Video texts are not easily recognized since they usually suffer from poor visual quality and compression artifacts. To tackle this problem, we propose a super-resolution technique to reconstruct the high-resolution video texts from the low-resolution video texts extracted from multiple keyframes. The adoption of super-resolution technique has greatly improved the performance of OCR recognition. Besides recognizing video texts, audio signal is also processed for speech recognition. Since speech recognition is usually error-prone due to background noise and speaker's accent, we adopt phrase and keyword spotting approach. Instead of recognizing speech in an unconstraint domain, phrases and keywords are extracted directly from electronic slides to guide the speech recognition. The final step in this framework is to match or synchronize the recognized text and speech with the titles and contents extracted from presentation slides. Algorithms like fuzzy string matching [13] could be adopted for this application.

In this paper, we mainly focus topics on the analysis of video text and speech signal. The remaining paper is organized as follows. Section 2 presents the related works. Section 3 introduces our approach for slide transition detection. Section 4 presents the details in detecting, reconstructing and segmenting video texts. Section 5 describes our keyword spotting approach for speech recognition. Section 6 presents the experimental results, while Section 7 concludes this paper.

2. Related Works

To date, numerous efforts have been attempted to construct structured multimedia documents from live presentations [1, 4, 5, 12]. The produced documents, ideally, should contain synchronized audio, video, image and text. Although recent research has led to advances in software for education, creating multimedia presentation documents remains primarily a manual and labor-intensive process. Existing representative systems include Classroom 2000 [1] and Interactive Virtual Classroom [4]. The goal of Classroom 2000 project developed at Georgia Tech is to automate the authoring of multimedia documents from live presentations, but in a structured environment. In this project, audio and video links need to be manually generated from videotaped lectures.

To automate structuring and indexing, major research issues include the detection of slide transition, the detection of text regions in viewgraph, recognition of characters and words, tracking of pointers and animation, gesture analysis, speech recognition, and the synchronization of videos, audios and presentation slides. Related works include [5, 7, 11, 12, 14].

The detection of slide transitions has been actively addressed since it serves as the first fundamental step towards the semantic structuring of lecture video content [7, 11, 12, 14]. The term "slide transition" refers to the flipping of slides either manually by hand or electronically by pressing a button. Traditional shot boundary detection techniques [20, 21] such as frame difference and color histogram have been applied for detecting slide transitions, however, yield poor results [12, 14] especially for slides flipped electronically. This is mainly due to the fact that most presenters tend to use the same design template for all electronic slides in one presentation. As a result, the contrast between the adjacent slides is normally too low to be detected. This problem, in fact, has motivated the studies of approaches in analyzing the layout and the content of video texts [7, 11, 12, 14], not only for detecting slide transitions, but also to facilitate the matching of videos and electronic slides.

Video OCR is an area of intensive exploration recently. The process of video OCR mainly includes the detection, segmentation and recognition of video texts. Texts embedded in the presentation videos mainly belong to scene texts. Compared with artificial or superimposed captions, they are relatively hard to be detected and recognized. Techniques in video text detection can be broadly categorized into three major groups: learning-based [8, 10], geometric-based [3, 9, 16], and texture-based [6, 22]. Representative works in video text segmentation include adaptive thresholding [17, 18], clustering [19] and character extraction filter [15]. Compared with text detection and segmenta-

tion, relatively few works have been reported for video text recognition [2, 15]. In fact, most approaches directly applied commercial OCRs for character recognition. As reported in [2], only about 50% of recognition accuracy (mainly for super-imposed captions) is attained by commercial OCRs.

While video texts analysis has attracted the researchers' attentions for lecture video applications, relatively few works were reported on how to index lecture videos by speech information. One interesting approach was reported in [11] recently. The author integrated the approaches in speech recognition and spoken document retrieval literatures for searching the transcribed audios with the keywords extracted from electronic slides. Our proposed approach is different from [11] in the following aspects: i) both phrases and keywords are extracted from electronic slides for retrieval, ii) instead of recognizing continuous speech signals, the signals are spotted by the extracted keywords and phrases, iii) the spotted keywords and phrases in audio signals are used directly to retrieve the most similar slide in an electronic document.

3. Slide Transition Detection

Intuitively, a new slide can be detected whenever there is a change of title, content, figure, or design template. In our approach, slide transitions are detected by both the background (figure and design template) and caption (title and content) cues on the projected slides. Our system is set up as follows. A camera is mounted in the lecture hall so as to capture the presenter and the projected electronic slides (see Figure 3 for an illustration). The position of camera is fixed and it stays stationary throughout a lecture. A presenter can move freely in front of the projected screen and use pointers to explain or highlight important concepts. The capture videos are encoded in MPEG format.

Figure 2 illustrates the overview of our approach. A video is initially partitioned into divisions of fixed interval. These divisions are referred to as time frames. They are composed of a sequence of images. A background template is computed from each time frame based on the foreground and background segmentation algorithm. The computed background template is used as a mask to detect caption and to compute energy due to background change. A text mask is also generated to compute energy due to caption change. The background and caption energies are utilized to determine whether a time frame contains slide transitions. If a transition is suspected, the caption and background similarity among the image frames of the time frame will be further compared to detect the exact slide transition.

Our approach operates directly in MPEG compressed domain. Instead of processing the original size image frames, DC image sequence extracted directly from DC co-

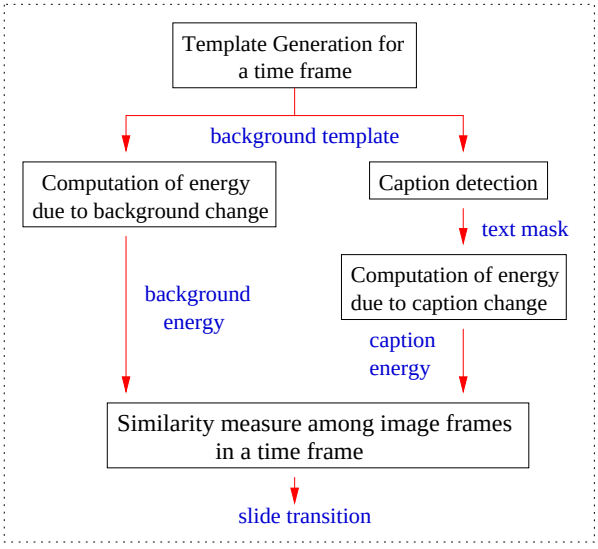

Figure 2. Approach for slide transition detection

efficients are used for template generation and energy computation. Besides DC coefficients, AC coefficients are also utilized for caption detection. The detailed of our approach can be found in [14]. This approach is efficient, it can process approximately 70 frames per second on a Pentium-IV platform. Figure 3 shows the segmented background and the detected caption regions of a video frame for slide transition detection.

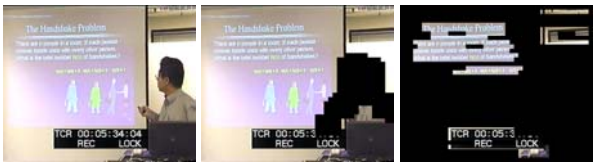

Figure 3. (left) Original image frame; (middle) segmented background; (right) detected captions for slide transition detection.

4. Video Text Analysis

After the slide transition detection, multiple keyframes are extracted from each shot[2]. The low-resolution texts embedded in multiple keyframes are then detected, reconstructed and segmented as high-resolution texts prior to OCR recognition. Unlike slide transition detection, all operations are carried out in uncompressed domain. For transition detection, a rough analysis of visual hints has already

[2]No specific keyframe selection algorithm is employed since the camera is static throughout a presentation. The number of selected keyframes in each shot is five in our experiments.

given us good enough accuracy. For video text analysis, nevertheless, a detailed analysis is necessary since the resolution of video texts can significantly affect the OCR accuracy.

4.1. Video Text Detection

The aim of text detection is to localize the exact text region in images or videos. Some factor including complex background, text-like scenes and the contrast between foreground texts and background scenes, will affect the result of detection. To detect text in lecture videos, we have tried both geometric-based and texture-based approach. From our experiments, we find that the geometric-based approach likes [3] gets better results, and meanwhile, is very efficient.

Our algorithm operates as follows. The LOG (Laplacian of Gaussian) is employed to detect edges in keyframes. The rectangle text boxes that surround the edge sets are then obtained. An attribute set is then computed for each surrounding rectangle. The attributes include the center of the rectangular text box, the mean and variance vectors corresponding to the foreground and background color distribution, and the threshold vector corresponding to the colors. After detecting the edges and computing their attributes, the next step is to exclude non-text regions using the following criteria: i) one or both dimension of the text box are too large; ii) the edge intensity is too low; iii) the edge density inside the region is too low.

The remaining edges are regarded as belonging to some characters. Since each character/word may consist of several edges or components, a loop is done to combine all edges that belong to the same character/word. The attributes obtained from the second step are used to check whether they are possibly of the same character/word.

With LOG, we can obtain enhanced correspondences on different edge scales by using a suitable deviation. A GMM (Gaussian mixture model) is used to represent background and foreground. Since characters in the same context share common properties, they are used to analyze the layout and refine detection results.

The text detection results may vary for different keyframes extracted from a shot. This is mainly due to the changes of lighting condition, shadow, and the movement of a presenter. In our approach, we integrate the results extracted from multiple keyframes and obtain the best possible text boxes in a shot.

4.2. Super-Resolution based Reconstruction

The main problem of recognizing video texts is the poor visual quality due to low image resolution. For instance, in our lecture videos, the height of a character is usually no more than 10 pixels which is too small for the commercial OCR systems. To improve the resolution, we employ super-resolution based approach. Our approach is mainly lain in two aspects: i) linear interpolation to expand a textbox, ii) multi-frames integration to smooth background scene while enhancing the contrast of foreground texts to background scene.

Denote L as a low resolution textbox, and $\mathcal{S}$ as the high-resolution textbox of L. The relationship between $\mathcal{S}$ and L is

$$\mathcal{S}(X,Y) = L(\frac{X}{a}, \frac{Y}{a}) = L(x^{'}, y^{'}) \tag{1}$$

where a is the interpolation factor, $x \leq x^{'} < x+1$ and $y \leq y^{'} < y+1$. By linear interpolation, we have

$$\begin{aligned} L(x, y^{'}) &= L(x,y) + (y^{'} - y) \times \\ &\quad (L(x, y+1) - L(x,y)) \end{aligned} \tag{2}$$

$$\begin{aligned} L(x+1, y^{'}) &= L(x+1,y) + (y^{'} - y) \times \\ &\quad (L(x+1, y+1) - L(x+1,y)) \end{aligned} \tag{3}$$

By further manipulating the above equations, we have

$$\begin{aligned} &\mathcal{S}(X,Y) = L(x^{'}, y^{'}) \\ &= L(x, y^{'}) + (x^{'} - x) \times (L(x+1, y^{'}) - L(x, y')) \end{aligned} \tag{4}$$

After linear interpolation, the final high-resolution textbox is obtained by integrating the results of text boxes obtained from multiple keyframes. The approach can enhance the foreground and background contrast. Let $\mathcal{S}_k$ as the high-resolution textbox of k^{th} keyframe, we compute the statistical information of these text boxes as follow

$$\mu_k(x,y) = \frac{1}{|w|} \times \sum_{p,q \in w} \mathcal{S}_k(x-p, y-q) \tag{5}$$

$$\mu(x,y) = \frac{1}{k} \times \sum_{k} \mu_k(x,y) \tag{6}$$

$$\begin{aligned} \sigma(x,y) &= \frac{1}{|w|} \times \max_k \\ &\quad \sqrt{\sum_{p,q \in w} \{\mathcal{S}_k(x-p, y-q) - \mu_k(x,y)\}^2} \end{aligned} \tag{7}$$

where w is a 5×5 local support window and $|w|$ is the cardinality of the window. Denote $\mathcal{S}^{'}$ as the final high-resolution textbox. We update the pixel values in $\mathcal{S}^{'}$ based on the computed statistical information. If $\sigma(x,y)$ is smaller than a predefined threshold, $\mathcal{S}^{'}(x,y) = \mu(x,y)$. Otherwise, $\mathcal{S}(x,y) = \min_k \mathcal{S}_k(x,y)$ or $\mathcal{S}(x,y) = \max_k \mathcal{S}_k(x,y)$ by guessing whether $\mathcal{S}(x,y)$ lies on a character. The guessing is done by checking the pixel values outside a small region of the low-resolution text boxes.

4.3. Video Text Segmentation

Since most current OCR systems use binary images as input, binarization (or segmentation) is a preprocessing step of text recognition. Given a high-resolution text box, the task is to determine whether the pixels belong to foreground characters or just lie in background scene. The high resolution texts usually have distinguishable colors between the foreground and background, and also have a high intensity contrast in a gray scale image. This seems make it easy to segment text and to describe the character using marginal distribution in a color space.

We utilize R/G/B/H/I components for text binarization. Figure 4 shows the histogram of a text box in I space. The foreground mean μ_f, background mean μ_b, foreground variance σ_f, and background variance σ_b are calculated for each component. Then the GMM (Gaussian mixture model) parameters of a text box are calculated and they can reflect how well each component is in segmenting and describing character properties. Each component is associated with a confidence as follows:

$$C_i = \frac{|\mu_b^i - \mu_f^i|}{\sigma_b^i - \sigma_f^i} \quad (8)$$

$$C_H = \frac{\min\left(|\mu_b^H - \mu_f^H|, 256 - |\mu_b^H - \mu_f^H|\right)}{\sigma_b^H - \sigma_f^H} \quad (9)$$

where $i = \{R, G, B, I\}$. The higher the value C, the more confident the corresponding component. The component with the highest confidence is selected to carry out the segmentation of foreground texts and background scene.

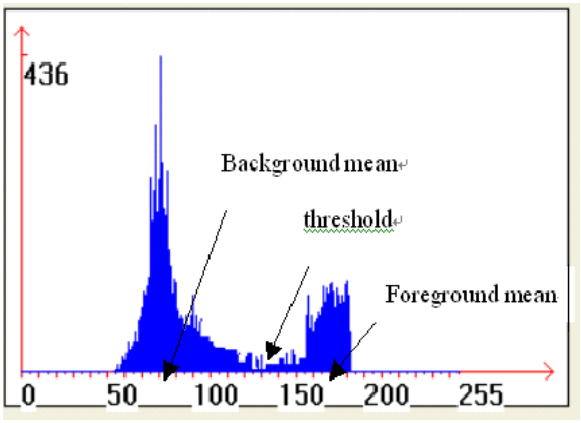

Figure 4. The histogram of a text box in I space.

5. Content Guided Speech Recognition

Besides video texts, what a presenter said during the presentation is another source of useful information. However, the speech, in general, is not as stable as video texts. This is mainly because different people can have different styles when talking, the content of speech is comparatively subjective. Moreover, the performance of speech recognition can be sensitive to the accent and pronunciation of speakers. Despite the disadvantages, speech signal, nevertheless, can still serve as a supplement, in particularly, when there is error happened in video text recognition.

Instead of recognizing speech in an unconstraint environment, we adopt a content guided recognition approach. We restrict ourselves to a simple indication of discussing topic by spotting the speech signal using the phrases and keywords extracted from the presentation slides. Phrases are actually the titles in slides, while keywords are obtained from the content of slides. A stop-word list is used to filter insignificant words in the electronic slides. Figure 5 illustrates the flow of our approach. First, the phrases and keywords extracted from the electronic slides are represented in XML format. Together with the audio signal, the XML file is input to the SR (speech recognition) engine. Based on the content of XML, the SR engine will output the phrases and keywords that are hit or recognized. Based on the hits, the retrieval engine will further compute the confidence and similarity scores to decide which slide is most likely the current discussing topic.

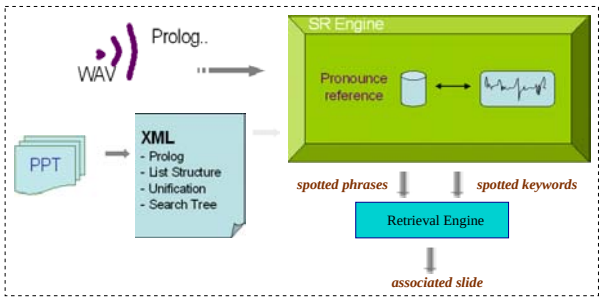

Figure 5. Slide detection by speech recognition.

Given the time index of a shot, the retrieval engine will buffer and collect a sequence of spotted phrases and keywords from the SR engine. These information will be utilized to determine the associated electronic slide of a given shot. Basically, the decision process can be divided into two stages. At the first stages, the retrieval engine computes confidence scores for each slide based on the "phrase hits". Normally most slides will get zero score, except those slides whose titles match the spotted phrase. The confidence score is dependent on how close a title can match with a spotted phrase. A maximum score of 1 will be given for an exact match. Let $Title$ and $Phrase$, respectively, as the sets of keywords in a title and a phrase. The confidence score is measured as

$$C = \frac{Title \cap Phrase}{\max\{|Title|, |Phrase|\}} \quad (10)$$

where $|Title|$ and $|Phrase|$ are, respectively, the number of keywords in a title and a phase. The value of C is in the range of $[0, 1]$. In principle, those slides with relatively high scores will be selected as the candidates for further processing. At the second stage, we adopt an approach similar to the traditional information retrieval techniques, except that the query is a list of spotted keywords. The spotted keywords will be used by the engine to retrieve the most similar slide from a set of candidate slides.

In the current implementation, we employ MSAPI 5.1 (Microsoft Speech API) to develop our application. We use the CnC (Command and Control) mode in MSAPI for speech recognition. The phases extracted from the slides are represented as dynamic grammars in the XML file format. Meanwhile, the extracted keywords are itemized as a reference list for keyword spotting.

6. Experiments

We conduct experiments on four lecture videos. The duration of each video is approximately 30 to 45 minutes. Same design template is used for all the slides in a presentation document. In total, our approach extracts 194 shots from these four videos. The recall and precision of our slide transition detection approach is 0.87 and 0.95 respectively.

For each shot, we evenly extract five keyframes along the temporal dimension for video text analysis. The common text boxes extracted from the keyframes are integrated and reconstructed as one high resolution text box prior to the foreground and background segmentation. Figure 6 shows the detected text boxes in different keyframes, while Figure 7 shows some of the reconstructed high-resolution text boxes from these keyframes. As seen in Figure 6, when the background is not very complicated, our text detection algorithm works very well. Some noises will be included if the text is connected with some other edges of scenes.

We also compare the performance of video segmentation for both low-resolution and high-resolution text boxes. The approach works pretty well for the titles in both resolutions. Almost all characters in the titles are correctly segmented, except few characters that are over-illuminated due to lighting conditions. Nevertheless, the main differences are: i) the borders or edges of high-resolution characters are much smoother, ii) in contrast to low-resolution, two adjacent segmented characters in high resolution text box are normally well separated. We found that these factors can make great impact for OCR recognition. Compare with the title segmentation, the characters embedded in the main content are generally difficult to be segmented due to small font sizes and over-illumination. Nevertheless, the segmentation of high-resolution text boxes is significantly better than the low-resolution text boxes. In low-resolution text boxes, the foreground and background scenes are not well separated. As a results, foreground characters are usually segmented into broken characters that are almost impossible for OCR recognition.

Tables 1 and 2 compare the commercial OCR performance for the low and high resolution texts. We employ the commercial OCR in [23] for this experiment. As indicated in the tables, the improvement of high-resolution texts over low-resolution texts is significant. The OCR can recognized 80% to 90% of the high resolutions titles, but can only recognize 20% to 40% of low resolution titles. The recognition of texts in the main contents is a difficult task. As shown in the tables, more than half of the characters embedded in videos are indeed not recognized by human. In the experiments, the commercial OCR fails to recognize almost all the low resolution characters. Nevertheless, approximately 30% to 60% of characters that are recognized by human are successfully recognized by the OCR when the high-resolution characters reconstructed. To measure the performance of text recognition in a more objective way, we compute the value of $recall$ as $\frac{N_c}{N_g - N_u}$.

The performance of speech recognition can vary significantly depending on the presenters' accent, background noise and the content of speech. To conduct a more objective evaluation, we ask three different people (including male and female speakers) to prepare three different presentations with the topics they are most familiar with. The duration of each presentation is around 20 to 30 minutes. Experimental results indicate that the proposed approach can link and index correctly approximately 65% of electronic slides.

7. Conclusion

We have presented our approach for structuring and indexing lecture videos. The approach mainly relies on the analysis of video texts and speech signals. Since the titles and contents that we expect from the text and speech recognition are always known in a priori, the content guided approach can be adopted to improve the performance of recognition. Experimental results indicate that our approach is effective, in particular when the super-resolution analysis of video texts, and the phrases and keywords spotting of speech signals are incorporated. Our future works include the synchronization and fusion of the recognized video texts and speeches for more reliable video indexing.

Acknowlegement

The work described in this paper was supported by a grant from City University of Hong Kong (Project No. 7100249) and a RGC Grant CityU 1072/02E (Project No. 9040693).

Table 1. Results of video text recognition (Low Resolution). N_g: number of ground-truth characters, N_c: number of recognized characters, N_I: number of characters output by OCR, N_u: number of characters not recognized by human.

Lecture Video	*Title*					*Content*					
	N_g	N_c	N_I	Recall	Precision	N_g	N_c	N_I	N_u	Recall	Precision
1	620	118	172	0.19	0.69	4117	4	33	2531	0.00	0.12
2	230	51	88	0.22	0.58	3162	0	16	1774	0.00	0.00
3	470	210	268	0.44	0.78	4224	0	18	2752	0.00	0.00
4	90	30	39	0.33	0.77	834	4	20	431	0.01	0.20

Table 2. Results of video text recognition (High Resolution).

Lecture Video	*Title*					*Content*					
	N_g	N_c	N_I	Recall	Precision	N_g	N_c	N_I	N_u	Recall	Precision
1	620	494	582	0.80	0.85	4117	432	1660	2531	0.27	0.26
2	230	218	230	0.95	0.95	3162	739	2037	1774	0.53	0.32
3	472	434	461	0.92	0.94	4224	955	1616	2752	0.65	0.59
4	90	81	91	0.90	0.89	843	169	413	431	0.41	0.41

References

[1] G. Abowd *et. al.* , "Teaching and Learning as Multimedia Authoring: The Classroom 2000 Project," *ACM Multimedia*, pp. 187-198, 2000.

[2] H. Aradhye, C. Dorai & J. -C. Shim, "Study of Embedded Font Context and Kernel Space Methods for Improved Video-text Recognition," *IBM Research Report RC 22064*, 2001.

[3] X. Chen *et. al.* , "Automatic Detection of Signs with Affine Transformation," *Proc. WACV*, Dec, 2002.

[4] S. G. Deshpande & J.-N. Hwang, "A Real-time Interactive Virtual Classroom Multimedia Distance Learning System," *IEEE Trans on Multimedia*, vol. 3, no. 4, pp. 432-444, 2001.

[5] L. He *et. al.* , "Auto-Summarization of Audio-Video Presentations," *ACM Multimedia*, pp. 489-498, 1999.

[6] X-S. Hua, W. Liu, H. J. Zhang, "Automatic Performance Evaluation for Video Text Detection," *Int. Conf. on Document Analysis and Recognition*, pp. 545-550, 2001.

[7] S. X. Ju *et. al.* , "Summarization of Videotaped Presentations: Automatic Analysis of Motion and Gesture," *IEEE Trans on CSVT*, vol. 8, no. 5, pp. 686-696, 1998.

[8] H. Li, D. Doerman and O. Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Trans. on Image Processing*, vol. 9, no. 1, 2000.

[9] R. Lienhart, "Automatic Text Segmentation and Text Recognition for Video Indexing," *Multimedia System Magazine*, vol. 8, pp. 69-81, 2000.

[10] R. Lienhart, "Localizing and Segmenting Text in Images and Videos," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, np. 4, April, 2002.

[11] T. F. S. -Mahmood, "Indexing for topics in videos using foils," *Computer Vis. and Pattern Recog.*, pp. 312-319, 2000.

[12] S. Mukhopadhyay & B. Smith, "Passive Capture and Structuring of Lectures," *ACM Multimedia*, pp. 477-487, 1999.

[13] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computer Survey*. pp. 32-86, Vol. 33, No. 1, 2001.

[14] C. W. Ngo, T. C. Pong & T. S. Huang, "Detection of Slide Transition for Topic Indexing," *Int. Conf. on Multimedia Expo*, 2002.

[15] T. Sato *et. al.*, "Video OCR for Digital News Archieve," *ICCV Workshop on Image and Video Retrieval*, 1998.

[16] J. C. Shim, C. Dorai & R, Bolle, "Automatic Text Extraction from Video for Content-based Annotation and Retrieval," *Int. Conf. on Pattern Recognition*, 1998.

[17] O. D. Trier & A. Jain, "Goal-Directed Evaluation of Binarization Methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 1191-1201, 1995.

[18] C. Wolf *et. al.* "Text Localization, Enhancement and Binarization in Multimedia Documents," *International Conference on Pattern Recognition*, pp. 1037-1040, 2002.

[19] E. K. Wong *et. al.* "A New Robust Algorithm for Video Text Extraction," *Pattern Recog.*, vol. 36, pp. 1397-1406, 2003.

[20] B. L. Yeo & B. Liu, "Rapid Scene Analysis on Compressed Video," *IEEE Trans. on CSVT*, vol. 5, no. 6, pp. 533-44, 1995

[21] H. J. Zhang *et. al.* , "Automatic Partitioning of full-motion video," *ACM Multimedia Sys.*, Vol. 1, No. 1, pp. 10-28, 1993.

[22] Y. Zhong *et. al.* , "Automatic Caption Localization in Compressed Video," *IEEE Trans. on PAMI*, vol. 22, no. 4, 2000.

[23] OmniPage Pro 12, http://www.scansoft.com/omnipage/

Figure 6. Experimental results for video text detection.

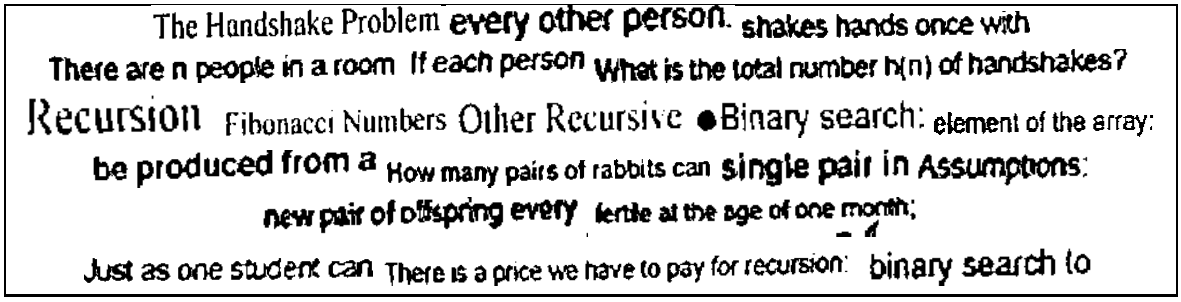

Figure 7. Some of the high-resolution text boxes extracted from the video frames shown in Figure 6.