10-2008

Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis

Feng WANG

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Ting-Chuen PONG

Citation

1

Pattern Recognition 41 (2008) 3257 – 3269

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis

Feng Wang[a,*], Chong-Wah Ngo[b], Ting-Chuen Pong[a]

[a] *Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong*
[b] *Department of Computer Science, City University of Hong Kong, Hong Kong*

ARTICLE INFO

Article history:
Received 19 October 2006
Received in revised form 31 December 2007
Accepted 19 March 2008

Keywords:
Topic detection
Video text analysis
Super-resolution reconstruction
Synchronization of lecture videos and electronic slides

ABSTRACT

This paper presents an unified approach in analyzing and structuring the content of videotaped lectures for distance learning applications. By structuring lecture videos, we can support topic indexing and semantic querying of multimedia documents captured in the traditional classrooms. Our goal in this paper is to automatically construct the cross references of lecture videos and textual documents so as to facilitate the synchronized browsing and presentation of multimedia information. The major issues involved in our approach are topical event detection, video text analysis and the matching of slide shots and external documents. In topical event detection, a novel transition detector is proposed to rapidly locate the slide shot boundaries by computing the changes of text and background regions in videos. For each detected topical event, multiple keyframes are extracted for video text detection, super-resolution reconstruction, binarization and recognition. A new approach for the reconstruction of high-resolution textboxes based on linear interpolation and multi-frame integration is also proposed for the effective binarization and recognition. The recognized characters are utilized to match the video slide shots and external documents based on our proposed title and content similarity measures.

1. Introduction

Multimedia database management has been an active research area in the past several years. The structuring, indexing and retrieval of video databases are essential tasks in many applications. These issues are intensively studied [1,2] and several working systems have been developed [3,4]. In this paper, we address the issue of video structuring on a particular type of video archive, i.e., videotaped lectures (or lecture videos), which are automatically taped during lectures without human intervention. Neither external software nor hardware is used to obtain the cross-media information. The recorded videos are basically unstructured and the relationship with external documents (e.g., lecture slides) is absent. These videos, in general, are not easy to browse and read, given the fact that they are poor in visual quality and only linear scanning of content is affordable. To upgrade the use of these videos, one way is to structure them by uncovering the linking between video segments and external documents. This not only enables the cross-reference browsing of videotaped documents, but also allows the remedy of video quality with the aid of external documents, as demonstrated in Ref. [5].

Fig. 1 shows a typical framework of structuring lecture video content for effective indexing. The inputs to the framework are an external document and a videotaped lecture that consists of the audio–visual information of presenting the document. Initially the video is partitioned into segments called *slide shots* according to the presentation. Each slide shot basically captures one page of the document. Video texts embedded in each slide shot are detected and recognized, while audio information is analyzed for speech recognition. Since video text and speech analysis are usually error-prone, the texts in the document can be extracted to guide the text and speech recognition. The recognized words from video and audio are then matched and synchronized with the document so as to construct the "linking indices" that model the cross-referenced relationship among them.

In this paper, we propose approaches to constructing the "linking indices" between low-quality lecture videos and external documents by video text analysis. Similar efforts have been done in Refs. [6–11], but not specifically with text cues, since texts in lecture videos are hard to be recognized by machines. In our database, the lecture videos are taped in the university classrooms. The external documents, which are the electronic copies of presentation slides (e.g. PowerPoint slides and .pdf documents), are projected to the screen in front of the classroom. The lecturers are free to move before the screen to present the documents to the students. A stationary

[*] Corresponding author. Tel.: +852 27 194 159.
E-mail address: wfeng@cs.ust.hk (F. Wang).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.03.024

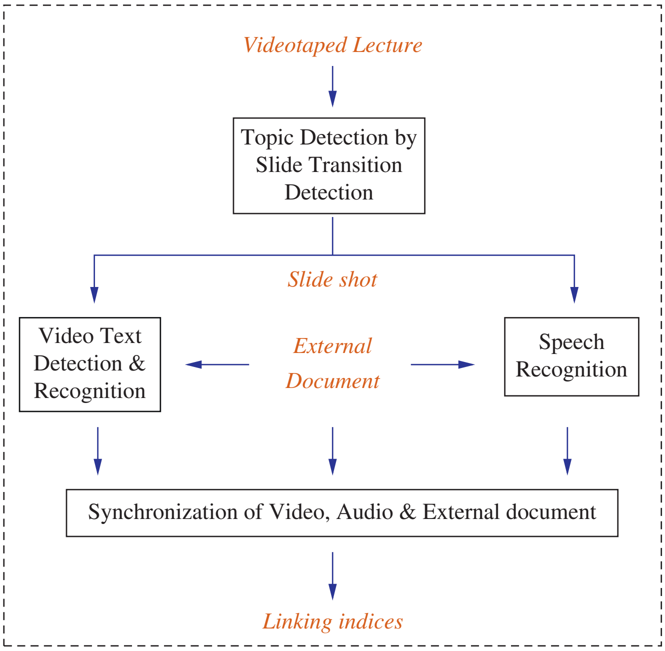

Videotaped Lecture
Topic Detection by Slide Transition Detection
Slide shot
Video Text Detection & Recognition
External Document
Speech Recognition
Synchronization of Video, Audio & External document
Linking indices

Fig. 1. A framework for lecture video indexing.

camera is amounted in the classroom and the screen is always captured. As a result, the texts in videos become the most reliable information for synchronizing lecture videos and external documents.

The flow of our approaches is similar to Fig. 1 except that in this paper only text analysis is employed. As seen in Fig. 1, speech is another important modality for lecture video indexing and can be incorporated with text. There are also numerous related issues to address, e.g. how to deal with environmental noise in speech recognition, and how to fuse speech with text. However, these issues are beyond the scope of this paper. In this work, we focus on the research issues in video text analysis, including topical event detection, high-resolution textbox reconstruction, and the linking of video shots and document pages by text matching.

Compared with other videos, lecture videos pose some new challenges to video structuring. For topic detection, the presence of a new page causes only very slight visual changes in the slide region. On the other hand, to improve the users' experience in learning with lecture videos, it is necessary to capture the lecturer's teaching activities such as gestures and postures together with the screen. The lecturer's motion results in much more significant frame difference. The slide transition detection based on visual cues is therefore greatly affected by the lecturer's motion. The resolution and quality of the texts are usually very low. The existing optical character recognition (OCR) softwares cannot be directly applied to lecture videos. In this paper, we do both topic detection and synchronization by video text analysis. For the former, we propose a new approach that employs the salient visual changes of text regions for slide shot transition detection based on foreground/background segmentation and text region localization. For the latter, we propose an algorithm to reconstruct high-resolution textboxes by integrating multiple frames, which can significantly improve the text recognition results.

Since only video texts are utilized, we do not consider the cases when the topics of discussion are not aligned with the page being shown. From our experience in lecture capture, these cases do not happen frequently. The documents shown in class are usually designed to contain just the outline of the lecture. During presentation, the lecturer then explains the content in detail to the students. For instance, the lecturer can give more examples, some insight thought, and answer the students' questions. The oral presentation may not be exactly the same, but related to the current slide most time. Non-alignment between the oral presentation and the slides may happen, but will not be frequent, e.g., the lecturer forgets to show the new slide when moving on, or simply mentions a previous slide without showing it again. For these cases, natural language processing, audio and speech analysis are necessary for the structuring of video content, and can be incorporated with text to further improve this work.

2. Related works

2.1. Topical event detection

In the past few years, issues in multimedia authoring of live presentation have attracted numerous research attentions. Topical event detection has been actively addressed since it serves as the first fundamental step towards the semantic structuring of lecture video content [8–10,12,13]. Topical event detection is usually achieved by slide matching [9] or slide transition detection [8,10,13]. Slide matching is carried out by matching the content of slide regions located in videos with the real electronic slides. This approach is usually computationally slow since repeated or redundant matching is required in order to locate an exact boundary.

Slide transition detection, on the other hand, is achieved by measuring the dissimilarity of adjacent frames. The term "slide transition" refers to the flipping of slides either manually by hand or electronically by pressing a button. Slide transitions, unlike conventional shot transitions [14,15], do not show significant color changes in most cases. Since most presenters tend to apply the same design to all electronic slides in one presentation, the color content of adjacent slides could be very similar. Traditional shot boundary detectors [14,15] may miss such transitions; on the other hand, they can easily cause false alarms due to the motion of a presenter. In Refs. [8,10], the visual cues like the number of pixel changes [8] and the percentage of outlier points [10] due to motion warping of adjacent frames are utilized to locate slide boundaries. These approaches, like Refs. [14,15], can trigger false alarms especially when a presenter moves and occludes part of the projected slides.

Our approach for topical event detection is based on slide transition detection. To discount the effect due to the motion of a presenter, foreground (presenter) and background (projected screen) scenes are segmented. The salient changes of texts and figures in the slide region, instead of the visual changes in the whole frame as used in conventional shot detection [14,15], are numerically computed for slide transition detection.

2.2. Synchronization of lecture videos and external documents by text analysis

To date, synchronization of lecture videos and external documents can be a manual and labor-intensive task if the cross-reference of multiple sources is not explicitly synchronized by special software or hardware. In certain systems [6,11], the instructors need to manually edit time stamp information in order to match the electronic slides with the relevant audio–video clips. To tackle this problem, several approaches have been proposed to automate the synchronization by matching the content difference, spatial layout and background color of slides and videos [9,10,13]. Since electronic slides in one presentation are normally designed under one master template, the geometric and visual hints alone, in general, are not enough for reliable matching. The analysis of spatial layout in Refs. [9,16] or content difference in Ref. [13], for instance, is robust for matching slides that contain diagrams and images, but may not be appropriate for slides that contain mainly the textual information. In Ref. [9], speech transcription and spoken document retrieval techniques are also employed for synchronization. Nevertheless, the result of synchronization is heavily dependent on the content of speech and a presenter's accent and pronunciation.

In this paper, we present an automatic way of synchronization by video text analysis. Correct recognition of video text is a difficult task [17]. However we show that the recognition can be greatly improved by reconstructing the high-resolution video textboxes obtained from multiple keyframes. Video text analysis, more specifically video OCR, includes mainly the detection, segmentation and recognition of video texts. In general, text appearing in videos can be classified into two groups: scene text and artificial text [18]. Scene text is part of the scene to be captured, e.g. street names or shop names in the scene. Artificial text is produced separately from the video and is superimposed to it in a later stage, such as the headlines in a news program. Artificial texts could be noised only during video compression, but not during video capturing process. Text in lecture videos mainly belongs to scene texts. Noise in scene text may come from both video compression and video capture, e.g. due to varying lighting conditions. Thus, scene texts are usually more difficult to be detected and recognized compared with artificial texts.

To date, techniques in video text detection can be broadly categorized into two major groups: geometric-based [18–21] and texture-based [22–25]. Representative works in video text segmentation include adaptive threshold [26,27], clustering [28,29] and character extraction filter [30]. Compared with text detection and segmentation, relatively few works have been reported for video text recognition [17,30]. In fact, most approaches directly apply commercial OCRs for character recognition. As reported in Ref. [17], for superimposed captions, only about 50% of recognition accuracy is attained by commercial OCRs. From our experiments as shown in Table 5, the direct OCR recognition accuracy for lecture videos is even lower: only 38% of the texts in titles and 3% in content can be correctly recognized. Some examples of lecture video text can be found in Figs. 6, 10 and online [31].

To achieve high text recognition accuracy, some early efforts have been devoted to performing better binarization by adaptive threshold [26,27]. These methods are difficult to segment texts with similar colors to the background. In Ref. [29], a bi-color clustering algorithm is proposed by assuming there are only two colors: a foreground text color and a background color. However, such assumption is not true in many cases especially when the texts are embedded in complex background. In Ref. [28], after considering complex background, all the pixels are classified into K segments. Texts are extracted and recognized for different K values ($K = 2, 3, 4$). All the OCR results are verified and combined to produce the final text file. The above methods focus on text segmentation for improving recognition results and address the issue of dealing with complex background in the text segmentation phase. However, this is not the main problem in lecture videos since the slides are usually designed with clear background so that students in class can easily recognize the texts.

For text recognition in lecture videos, the difficulty lies in the low resolution and visual quality of texts. To solve this problem, we reconstruct high-resolution textboxes by integrating multi-frames before text segmentation and recognition. Super-resolution has been studied for decades. In the traditional image super-resolution problem [32], only one single image is available. Video super-resolution can take advantage of multiple frames that contain the same objects or scenes. In Refs. [33,34], different approaches are proposed to eliminate the blur due to the motion of the camera and the object. Several works have also addressed specifically text enhancement for improving video OCR performance. In Ref. [35], linear interpolation is employed to increase the resolution of texts. In Refs. [30,36,37], both linear interpolation and multi-frame integration are used for video text enhancement. Most approaches assume the foreground text to be static over a long sequence of frames while the background pixels keep changing. In Ref. [36], a multi-frame integration is proposed to smooth the background scene by registering and averaging textboxes of different frames over time. In Refs. [30,37], by assuming that the foreground captions are known with white (or black) colors, a minimum (or maximum) operator is applied to reduce the variation of the background scene in news videos. In Ref. [38], textboxes with higher background/foreground contrast are thought to have better visual quality and readability. HCF (high contrast frame) and HCB (high contrast block) are selected from different frames with the same text, and fed to OCR for text recognition. Because of the moving background assumption, the approaches in Refs. [30,36–38] are found to be effective particularly for recognition of artificial texts such as headlines in news videos. For scene texts, when text and background are both static such as in our videos, the reconstructed or selected textbox from multi-frames by these approaches only changes little and sometimes the effect is even worse for OCR recognition, compared with the textbox from a single frame.

Unlike superimposed captions, the noise that appears in lecture videos is mainly due to the factors such as lighting variance, un-unique lighting distribution and screen reflection, the frequent refreshment of the screen, and the shadow and occlusion of the presenter. These factors usually create varying noise effects particularly on the regions near the boundaries of foreground characters and background scene. In our approach, we also adopt linear interpolation and multi-frame integration but without the assumptions of moving background or foreground character color like Refs. [30,36]. Initially, we expand the size of a textbox with sub-pixel accuracy by linear interpolation. Then, by considering the local information, i.e. color distributions inside and outside the textbox, each pixel is classified as either lying entirely on the foreground/background or near the boundary of background/foreground. Instead of simple averaging, maximum or minimum operator [30,36], the proposed multi-frame integration selects appropriate values based on classification to enhance the color contrast between texts and background. This approach is found to be effective in improving the results of binarization and OCR recognition.

3. Overview of proposed approaches

Our approaches in topical event detection and synchronization are mainly based on the analysis of video texts. To detect the transitions of slide shots, text and figure changes are numerically computed. To synchronize videos and external documents, texts are extracted and binarized for recognition. A critical technique that guarantees the robustness of the proposed approach is the reconstruction of high-resolution video texts prior to recognition. We show the significant difference in recognition accuracy between the high and low-resolution characters.

For topical event detection, our approach is based on slide transition detection. We take into account the background (projected screen) and foreground (presenter) information. The foreground and background scenes are first segmented. The text regions in the segmented background are then located and utilized to detect the slide transitions. To be efficient, the segmentation is conducted on an image volume formed by hundreds of video frames as a whole at each iteration.

Multiple keyframes are extracted from each slide shot for text recognition and synchronization. Firstly we employ the statistical analysis of geometric properties in characters to detect textboxes. The same textboxes extracted from multiple keyframes are then integrated and reconstructed as a high-resolution textbox. The textbox is binarized and input directly to commercial OCR package for recognition. The recognized characters are utilized to match slide shots and external documents. In this paper, we propose a two-stage matching algorithm based on title and content similarity measures.

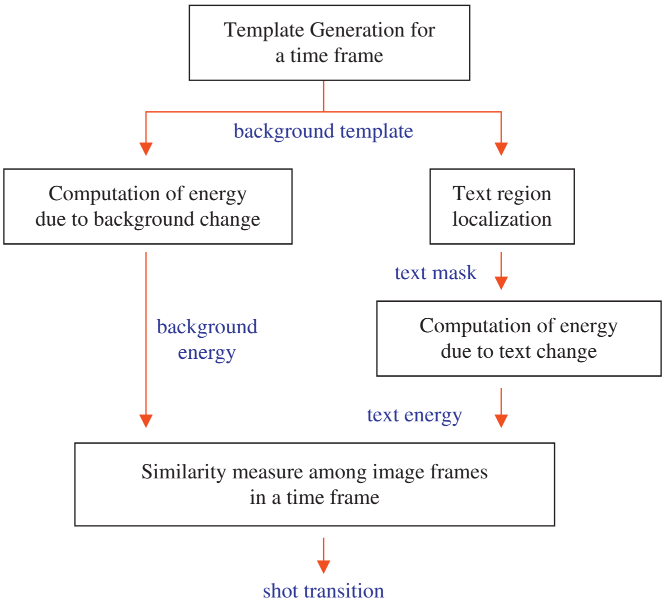

Template Generation for a time frame
background template
Computation of energy due to background change
Text region localization
text mask
background energy
Computation of energy due to text change
text energy
Similarity measure among image frames in a time frame
shot transition

Fig. 2. Framework for shot transition detection.

This approach not only speeds up the matching time, but also enhances the quality of matching since the reconstructed high-resolution titles can always be reliably recognized.

For efficiency, shot transition detection is carried out directly in the MPEG compressed domain. Since the detection of slide shot boundaries does not require character recognition, a rough analysis of DCT coefficients can already give us good enough accuracy. Synchronization, in contrast, requires detailed analysis since the resolution of video texts can have great impact for OCR. In our case, the selected keyframes will be decompressed for video text detection, binarization and recognition.

4. Topical event detection

Topical event detection mainly relies on the text and background cues in videos. The background cues include the visual changes in images, graphics and diagrams. To speed up the processing time, we adopt two strategies: (i) a video is processed in volumes and each volume corresponds to a time interval composed of a group of DC frames, (ii) AC coefficients are directly utilized to locate the text regions.

Initially, a video is temporally partitioned into divisions of fixed intervals. Each division contains a set of DC images. For convenience, we refer a division as a time interval. Fig. 2 shows the overview of our proposed framework. The algorithm works directly in the MPEG compressed domain. Initially, a set of templates are computed from a time interval for background and foreground segmentation. The resulting background template is used as a mask to locate text regions and to compute energy due to background change. A text mask is also generated for the computation of energy due to text change. Both background and text energies are utilized to decide if a time interval contains slide transitions. Once a transition is suspected, the text and background regions similarity among the image frames within a time interval are investigated to detect the exact slide transitions.

For the ease of understanding, the following notations are used in this section:

- The kth time interval of a video is denoted as $\mathbf{T}_k$. The total number of image frames in a time interval is $\#\mathbf{T}_k$. The tth DC image frame of $\mathbf{T}_k$ is denoted as $f_k(t)$. The pixel value of $f_k(t)$ at location (i, j) is written as $f_k(i, j, t)$. The size of a DC image frame is assumed as $M \times N$.
- The energy computed by background change detector is denoted as $\mathbf{E}_b$, while the energy computed by text change detector is denoted as $\mathbf{E}_c$.
- The DCT coefficients are denoted as ρ_{uv}, where ρ_{00} denotes DC coefficient while ρ_{uv} denotes AC coefficients for $u, v \neq 0$. The coefficients of a DCT block at $f_k(i, j, t)$ is simply indexed as $\rho_{uv}(i, j, t)$.

4.1. Template generation

To robustly segment the background and foreground scenes, the statistical mean and standard deviation of pixels along the time dimension are modeled. Two templates, namely mean μ_k and standard deviation σ_k templates, are generated to represent the statistical changes at time interval $\mathbf{T}_k$. The templates μ_k and σ_k are computed as

$$\mu_k(i, j) = \frac{1}{\#\mathbf{T}_k} \sum_{t \in \mathbf{T}_k} f_k(i, j, t) \tag{1}$$

$$\sigma_k(i, j) = \frac{1}{\#\mathbf{T}_k} \sqrt{\sum_{t \in \mathbf{T}_k} \{f_k(i, j, t) - \mu_k(i, j)\}^2} \tag{2}$$

Since our videos are taped with static cameras, we can assume that the background pixels are statistically unchanged unless they are occluded by foreground objects. A background template which contains only binary values is generated as

$$b_k(i, j) = \begin{cases} 0 & \sigma_k(i, j) > 2 \times {}_k \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where ${}_k = 1/(M \times N) \sum_i \sum_j \sigma_k(i, j)$ is the mean standard deviation of time interval $\mathbf{T}_k$ and $M \times N$ is the size of an image. The background template will be used as a mask for computing $\mathbf{E}_b$ and $\mathbf{E}_c$. The energy $\mathbf{E}_b$ due to background change is

$$\mathbf{E}_b = \frac{1}{\#b_{k-1}} \sum_i \sum_j b_{k-1}(i, j) \times |\mu_k(i, j) - \mu_{k-1}(i, j)| \tag{4}$$

where $\#b_k = \sum_i \sum_j b_k(i, j)$ is a normalizing term, and $\mu_{k-1}(i, j)$ is the mean template at time interval $\mathbf{T}_{k-1}$.

4.2. Approximate text region localization

Text regions are generally composed of a unique texture pattern. This pattern is due to the horizontal intensity variations caused by the characters within a text line and the vertical intensity variations caused by the spacing between text lines. We adopt the approach in Ref. [25] that utilizes AC coefficients ρ_{uv} to approximately but rapidly locate the text regions. At each time interval $\mathbf{T}_k$, the potential text regions are characterized by the horizontal EH_k and vertical EV_k text energies where

$$EH_k(i, j) = \frac{b_k(i, j)}{\#\mathbf{T}_k} \times \sum_{t \in \mathbf{T}_k} \sum_{v=2}^{6} |\rho_{0v}(i, j, t)| \tag{5}$$

$$EV_k(i, j) = \frac{b_k(i, j)}{\#\mathbf{T}_k} \times \sum_{t \in \mathbf{T}_k} \sum_{u=1}^{6} |\rho_{u0}(i, j, t)| \tag{6}$$

Notice that the background mask b_k is utilized in order to avoid characterizing the foreground regions as texts. Based on Eq. (5), a DCT block indexed by (i, j) is detected as a potential text candidate

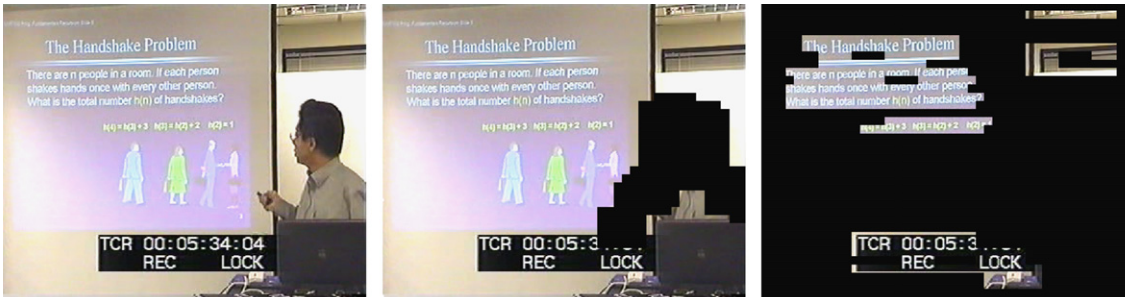

The Handshake Problem
There are n people in a room. If each person shakes hands once with every other person. What is the total number h(n) of handshakes?
TCR 00:05:34:04
REC LOCK

Fig. 3. (left) Original image frame; (middle) segmented background; (right) detected text regions.

if $EH_k(i,j)$ is greater than 1.45 times the average horizontal text energy of all the corresponding DCT coefficients in $\mathbf{T}_k$. The detected text regions are then refined by two morphological operators: a *closing* operator followed by an *opening* operator. This processing step basically removes most of the isolated noisy blocks and merges the nearby detached text blocks into coherent regions. A larger structural element size can remove larger noisy blocks (which might be text) and merge two blocks that are farther away from each other (which might not belong to the same text region). The structural element size is set to be 1 × 3 by following Ref. [25]. Subsequently, contour tracking and connectivity analysis are implemented to segment text regions. Finally, three criterions are used to assess the validity of a text region: (1) the ratio of width over height is at least 1.5; (2) the size must cover at least 10 DCT blocks; (3) the total EV_k of a text region is at least 80. The empirical parameters such as the number of AC coefficients used in text detection are based on the setting in Ref. [25]. These parameters work well in most cases. As discussed in Ref. [25], some failure examples may happen for large-scale text (e.g. occupies half of the screen) which do not possess local texture. However, this is not the case in this application since very few large font texts are used in the slides. Furthermore, since we do not require the precise detection of text regions, few missing or falsely inserted text regions will not seriously affect the slide transition detection as most text regions can be detected. Fig. 3 shows the segmented background and detected text regions of a video frame.

The detected text regions are utilized as a text mask. Denote c_k as the text mask for $\mathbf{T}_k$, the energy $\mathbf{E}_c$ due to text change is computed as

$$\mathbf{E}_c = \frac{1}{\#c_{k-1}} \sum_i \sum_j c_{k-1}(i,j) \times |\mu_k(i,j) - \mu_{k-1}(i,j)| \tag{7}$$

where $\#c_k = \sum_i \sum_j c_k(i,j)$, $c_k(i,j) = 1$ if the corresponding location belongs to a text region and $c_k(i,j) = 0$ otherwise.

4.3. Slide shot transition detection algorithm

The algorithm for locating the exact slide transition is given in Fig. 4. For each time interval $\mathbf{T}_k = \{f_k(0), f_k(1), \ldots, f_k(m)\}$, energies due to background and caption changes are computed, respectively. In the implementation, two adjacent time intervals should overlap, i.e., $\mathbf{T}_{k-1} \cap \mathbf{T}_k \neq \emptyset$. This is to prevent false negatives which can happen if a slide transition resides in the last few frames of $\mathbf{T}_k$. This is due to the fact that the statistical mean template μ_k in Eq. (1) is used to represent $\mathbf{T}_k$. In the algorithm, if either the background or caption energy exceeds a pre-defined threshold, the exact transition will be determined by measuring the similarity among the image frames within $\mathbf{T}_k$. Denote Sim_c and Sim_b as the caption and background region similarities, respectively, between two image frames $f_k(t)$ and $f_k(t+1)$, we have

```
1. Generate new templates for time frame $\mathbf{T}_k$.
2. Compute the background change energy $\mathbf{E}_b$ as shown in Eqn (4).
3. Detect text change regions and generate text mask.
4. Compute the text change energy $\mathbf{E}_c$ as shown in Eqn (7).
5. If either $\mathbf{E}_b$ or $\mathbf{E}_c$ is greater than a threshold
   (a) Locate the exact transition by similarity measure
       among the image frames in $\mathbf{T}_k$ based on Eqn (8) and Eqn (9).
   (b) Update the templates in $\mathbf{T}_k$.
6. Repeat Step 1 for $\mathbf{T}_{k+1}$.
```

Fig. 4. Shot transition detection algorithm.

$$Sim_c = \frac{\sum_i \sum_j c_{k-1}(i,j) \times |f_k(i,j,t) - f_k(i,j,t-1)|}{\sum_i \sum_j c_{k-1}(i,j) \times |f_k(i,j,t+1) - f_k(i,j,t)|} \tag{8}$$

$$Sim_b = \frac{\sum_i \sum_j b_{k-1}(i,j) \times |f_k(i,j,t) - f_k(i,j,t-1)|}{\sum_i \sum_j b_{k-1}(i,j) \times |f_k(i,j,t+1) - f_k(i,j,t)|} \tag{9}$$

In principle, the similarity value is low if the value of the denominator is large. The numerator is a weighting factor such that only local minimum will be detected as a shot transition. A frame $f_k(t+1)$ is determined as the beginning of a new page presentation if $Sim_c < Sim_T$ or $Sim_b < Sim_T$ where the threshold $Sim_T = 0.4$ is empirically determined. Since the visual change is always very small when the slide is not changed, and relatively much larger when a new slide is shown, the detection result is not sensitive to the threshold. In Eqs. (8), (9), Sim_c and Sim_b are normalized by the background and caption masks, and will not be affected by different occlusion scales. Furthermore, since the ratio of visual changes in two consecutive frame pairs, instead of the absolute frame difference is used, the similarity measure can cope with different illumination conditions. After a detection, the current templates including c_k, b_k, μ_k and σ_k need to be updated since they are computed from the image frames of $\mathbf{T}_k$ which may contain shot transitions. Notice that in Eqs. (4), (7)–(9), the mask c_{k-1} or b_{k-1} is employed instead of c_k or b_k. This is simply because c_{k-1} and b_{k-1} are computed based on a time interval that contains no shot transition and hence are more reliable to measure the changes. In step 5(a) of the algorithm, the search for an exact transition is only conducted in $\mathbf{T}_k$ instead of both $\mathbf{T}_{k-1}$ and $\mathbf{T}_k$. This is because the templates in $\mathbf{T}_k$ will be re-computed in step 5(b) when a transition is detected.

5. Video text analysis

Unlike the detection of slide shot transitions, the processing of video text is aimed for character recognition and hence requires detailed analysis. For each slide shot, five frames are evenly extracted along the time dimension. These frames are treated as keyframes which represent the essential content of the current slide shot, and decompressed for video text analysis. In this application, we do not research which keyframe is better for text recognition, but collect multiple frames along the time axis for integration. In each keyframe, video text detection is carried out to extract textboxes. The textboxes obtained from multiple keyframes are integrated for super-resolution reconstruction. High-resolution textboxes are then binarized and recognized by commercial OCR.

5.1. Text detection

The aim of text detection is to locate the exact text region in images or videos. Some factors including complex background, text-like scenes and the low contrast of foreground texts to background scenes, will affect the results of detection. The current algorithms for text detection can be separated into two categories: geometry-based and texture-based approaches. Compared with the latter one, the geometry-based approach is easier to implement and more efficient, but much attention should be paid to noises. In our system, we employ a geometry-based algorithm [19,20] to detect text regions in the keyframes. In Refs. [19,20], the algorithm is developed for sign text recognition. Although the scenes might be complex, the background of the frame containing sign texts is usually designed to be unique so that people can easily recognize the signs. The geometry-based text detection is thus appropriate for sign detection. In our application, the slides are also designed to have relatively simple background. Our experiments show that this algorithm works for text detection in lecture videos.

The algorithm operates as follows. The LOG (Laplacian of Gaussian) is employed to detect edges in keyframes. We obtain the rectangles surrounding the edges, and then an attribution set is computed for each rectangle. The attributes here include the center, height and width of the rectangle, the edge intensity inside the rectangle, the mean and variance corresponding to the foreground and background color distribution. After getting the edges and their attributes, the following criteria are used to exclude non-text regions: (i) one or both dimensions of the textbox are too large or small; (ii) the edge intensity is too low; (iii) the edge inside the region is too simple. The attributes and criteria we use are the same as in Ref. [20, p. 89] and they are quite effective in removing false alarms.

The remaining edges after excluding non-text regions 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 are used to check whether they are possibly of the same character/word. A GMM (Gaussian mixture model) is used to represent background and foreground. Since characters in the same context share some common properties, they are used to analyze the layout and refine detection results. The details of algorithm can be found in Ref. [20, pp. 91–92].

The results of text detection may vary for different keyframes of a shot. This is mainly due to the changes of lighting condition, shadow and the movement of a presenter. Some texts may be hidden in one frame, while appear in the other frames. We integrate the detection results from multiple keyframes. All the textboxes detected in every frame will be counted in the final result. This may include some noise in the detected textboxes and add a little workload to the OCR recognition, but usually will not affect synchronization.

5.2. Super-resolution 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 the super-resolution based approach. Our approach is composed of two steps: (i) linear interpolation, and (ii) multi-frames integration. The first step linearly expands the detected textboxes in keyframes. The second step integrates the expanded textboxes of multiple keyframes by enhancing the contrast of foreground texts to background scene.

5.2.1. Linear interpolation

Denote L as a low resolution textbox, and $\mathscr{S}$ as the high-resolution textbox of L. Let (X, Y) as the pixel index to $\mathscr{S}$ and (x, y) as the pixel index to L. The relationship between $\mathscr{S}$ and L is

$$\mathscr{S}(X,Y) = L\left(\frac{X}{a}, \frac{Y}{a}\right) = L(x', y') \tag{10}$$

where a is the interpolation factor, (x', y') is a sub-pixel index to L, and $x \leqslant x' < x+1$ and $y \leqslant y' < y+1$. By linear interpolation, we have

$$L(x, y') = L(x,y) + (y'-y) \times (L(x, y+1) - L(x,y))$$

$$L(x+1, y') = L(x+1, y) + (y'-y) \times (L(x+1, y+1) - L(x+1, y))$$

By further manipulating the above equations, a high-resolution textbox is reconstructed as follows:

$$\begin{aligned}\mathscr{S}(X,Y) &= L(x', y') \\ &= L(x, y') + (x'-x) \times (L(x+1, y') - L(x, y'))\end{aligned} \tag{11}$$

5.2.2. Multi-frame integration

In our camera setting, although there is no background motion, some difference always exists between keyframes, particularly on the boundaries between texts and background. These noises come from lighting variation, uneven lighting distribution, screen reflection, the frequent refreshment of screen, the movement of foreground objects and shadows. To remedy the noise effects, we adopt a multi-frame integration strategy to combine the results of textboxes obtained by linear interpolation. In this strategy, the value of each pixel in a high-resolution textbox is updated depending on whether it lies on a character, background or near the border of character and background. The update is aimed to enhance the foreground and background contrast of a textbox.

Let $\mathscr{S}_k$ denote the high-resolution textbox of the kth keyframe. For each pixel indexed by (X, Y), we compute the statistical information of the textboxes as follows:

$$\mu_k(X,Y) = \frac{1}{|W|} \times \sum_{p,q \in W} \mathscr{S}_k(X-p, Y-q) \tag{12}$$

$$\mu(X,Y) = \frac{1}{k} \times \sum_{k} \mu_k(X,Y) \tag{13}$$

$$\sigma(X,Y) = \frac{1}{|W|} \max_{k} \sqrt{\sum_{p,q \in W} \{\mathscr{S}_k(X-p, Y-q) - \mu_k(X,Y)\}^2} \tag{14}$$

where W is a 5×5 local support window and $|W|$ is the cardinality of the window. $\mu(X, Y)$ measures the mean of $\mu_k(X, Y)$ across k textboxes, while $\sigma(X, Y)$ characterizes the edge intensity near a pixel. Denote $\mathscr{S}'$ as the multi-frame integrated high-resolution textbox, we update the pixel values in $\mathscr{S}'$ based on the computed statistical information. We consider two cases: (i) a pixel lies entirely on the background or a character, (ii) a pixel lies near the boundary of background and a character. These two cases are determined by the value of $\sigma(X, Y)$. If the value of $\sigma(X, Y)$ is smaller than a predefined threshold, case (i) is assumed. Otherwise, case (ii) is considered. In both cases, we define a small region R surrounding the textbox T as shown in Fig. 5. The region R is assumed to lie in the background scene. Let μ_f and μ_b denote the mean pixel values of foreground and background scenes, μ_R and μ_T denote the mean values of R and T, σ_T denote the standard deviation of T, respectively. The details of two cases are as follows.

In case (i), a pixel lies entirely in the background or a character. If $\mu_R < \mu_T$, apparently $\mu_f > \mu_b$. Otherwise, $\mu_b > \mu_f$. Assume $\mu_f > \mu_b$, to guess whether a pixel (X, Y) lies in the background or foreground scene, the values of $\mu(X, Y)$, μ_T, and σ_T are compared. Intuitively, if $\mu(X, Y) > \mu_T + \sigma_T$, the pixel at (X, Y) should belong to foreground. Otherwise, the pixel belongs to background. Based on this information, we can easily enhance the contrast of foreground and background scenes in a new high-resolution textbox $\mathscr{S}'$ as follows:

$$\mathscr{S}'(X, Y) = \begin{cases} \max\limits_k \mathscr{S}_k(X, Y) & \text{if } \mu(X, Y) > \mu_T + \sigma_T \\ \min\limits_k \mathscr{S}_k(X, Y) & \text{if } \mu(X, Y) < \mu_T - \sigma_T \end{cases} \tag{15}$$

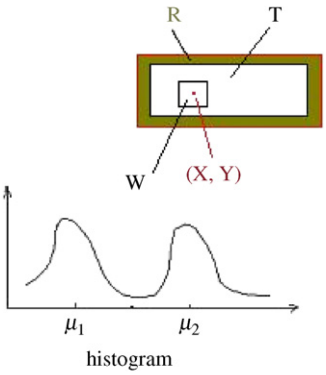


Fig. 5. A procedure to check whether the pixel (X, Y) lies on a character.

By Eq. (15), the pixels that apparently lie on the text edges which draw the skeleton of the texts can be highlighted, while the background is smoothed.

In case (ii), a pixel at (X, Y) lies near the boundary of text edge and background. In this case, comparing $\mu(X, Y)$ and a global value μ_T cannot robustly determine whether a pixel belongs to the foreground or background. This is mainly because noise usually exists on the boundaries of foreground characters and background scene. Instead, a local histogram is computed in the support window W of a pixel at (X, Y) to model the color distribution, as shown in Fig. 5. The distribution is basically characterized by two peaks (μ_1 and μ_2 in the figure) that correspond to the means of foreground (μ_f) and background (μ_b) scenes. If $\mu_R > (\mu_1 + \mu_2)/2$, $\mu_b = \mu_2$ and $\mu_f = \mu_1$. Otherwise $\mu_b = \mu_1$ and $\mu_f = \mu_2$. By measuring the distances from $\mu(X, Y)$ to μ_f and μ_b, we can determine whether a pixel lies on a character or the background. Similar to Eq. (15), assume $\mu_f > \mu_b$, we update $\mathscr{S}'$ as follows

$$\mathscr{S}'(X, Y) = \begin{cases} \max\limits_k \mathscr{S}_k(X, Y) & \text{if } |\mu(X, Y) - \mu_f| \\ & \quad < |\mu(X, Y) - \mu_b| \\ \min\limits_k \mathscr{S}_k(X, Y) & \text{otherwise} \end{cases} \tag{16}$$

Fig. 6 shows an example to compare the difference among the low-resolution (original) textboxes, the reconstructed high-resolution textboxes before and after multi-frame integration. As shown in the figure, multi-frame integration can effectively enhance the quality of text binarization after increasing the contrast between foreground and background scenes. By comparing the OCR outputs of different textboxes, apparently multi-frame integration achieves significantly higher recognition accuracy than linear interpolation. Fig. 7 shows another example that compares our approach with the algorithm proposed in Ref. [36]. As seen in this figure, in lecture videos, when the text and background are both static, the averaged textbox from multiple frames does not help most of the time. Our approach employs the local information in color distribution for high-resolution reconstruction, which proves useful in this case.

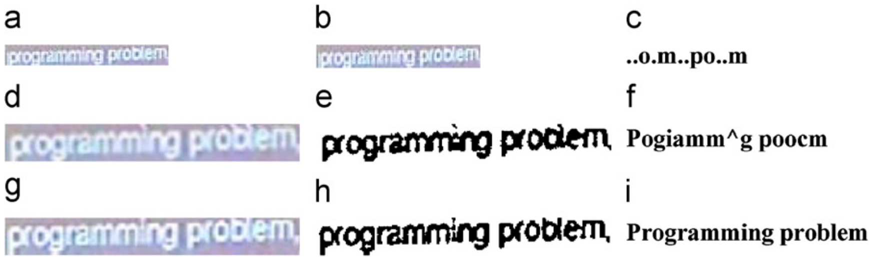


Fig. 6. Super-resolution reconstruction for OCR. (a) & (b) are textboxes in low resolution, (d) & (g) are the reconstructed high resolution text boxes, (e) & (h) are the binarized textboxes, (c), (f) & (i) are OCR output characters.

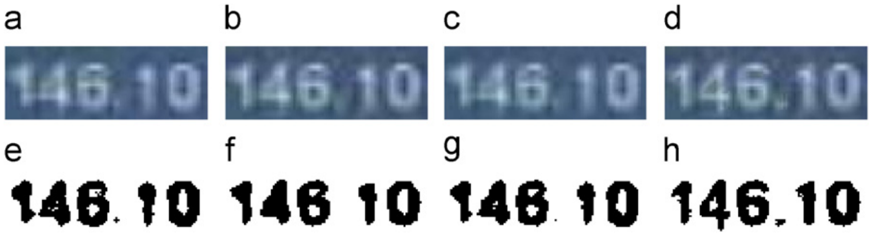


Fig. 7. Comparison between our approach and the algorithm proposed in Ref. [36]. (a) and (b) are two text boxes in low resolution, (c) is the averaged textbox by the algorithm in Ref. [36]. (d) is the high-resolution textbox reconstructed by our approach. (e)–(h) are the binarized textboxes of (a)–(d).

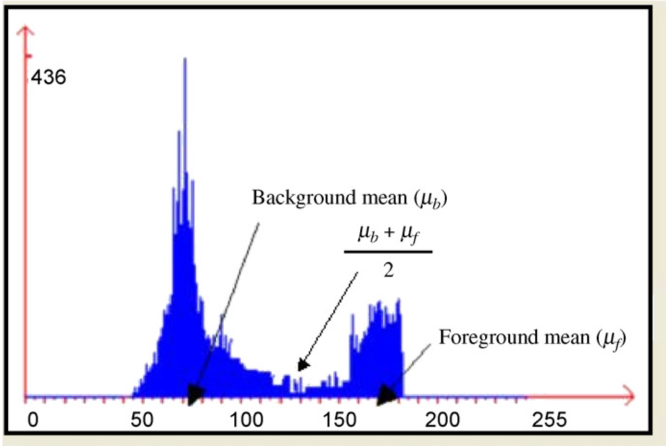


Fig. 8. The intensity distribution of a textbox in I space.

5.3. Text binarization

Since most OCR systems use binary images as input, binarization 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 makes it easy to segment text and to describe the characters using the marginal distribution in a color space.

We utilize R/G/B/H/I components for text binarization. Fig. 8 shows the pixel intensity distribution of a textbox 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 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:

$$C_i = \frac{|\mu_b^i - \mu_f^i|}{\sigma_b^i + \sigma_f^i} \tag{17}$$

$$C_H = \frac{\min(|\mu_b^H - \mu_f^H|, 256 - |\mu_b^H - \mu_f^H|)}{\sigma_b^H + \sigma_f^H} \tag{18}$$

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. As shown in Fig. 8, we select the value, $(\mu_f + \mu_b)/2$, for binarization. The binarized text boxes are fed to OCR system for character recognition. In our experiment, we use the commercial OCR system in Ref. [39] and the recognition results can be found in Table 3.

6. Synchronization by constructing cross-reference linking

The extracted texts from videos are used to synchronize the video shots and external documents. The texts from videos and documents are separated into titles and contents. The similarity between a video shot and a document page is based on the title and content similarities. Given a shot, a page with the highest similarity is linked to it for indexing.

Both title and content similarities are based on word matching. First, the extracted texts of title and content are separated into a list of words. Given two words w_1 and w_2, the edit distance is calculated. The definition and algorithm to compute the edit distance of two strings can be found in Ref. [40]. We define the matching of two words $M(w_1, w_2)$ as

$$M(w_1, w_2) = \begin{cases} 1 & \text{if } \max(Ed(w_1, w_2), Ed(w_2, w_1)) \\ & \quad \leqslant \dfrac{\min(len(w_1), len(w_2))}{4} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $len(w)$ is the length of a word. We say w_1 and w_2 are matched, i.e., $M(w_1, w_2) = 1$, if the edit distance between two words is small compared with their lengths.

6.1. Title similarity

Titles usually have a larger font size, as a result, the chance of success recognition is usually high. Let W_v and W_p denote the word sets, respectively, from a video shot v and a document page p. The set of matched words, W_m, between W_v and W_p is defined as

$$W_m = \{w_1 | w_1 \in W_p, \exists w_2 \in W_v, M(w_1, w_2) = 1\} \tag{20}$$

The similarity between the titles of a video shot v and a document page p is then defined as follows:

$$Sim_T(v, p) = \frac{1}{2} \times \left\{ \frac{\sum_{w_1 \in W_m} len(w_1)}{\sum_{w_2 \in W_p} len(w_2)} + \frac{\sum_{w_1 \in W_m} len(w_1)}{\sum_{w_2 \in W_v} len(w_2)} \right\} \tag{21}$$

In Eq. (21), a longer matched word will contribute more to the similarity. This is because the longer words are usually less frequently used and can help distinguish different pages if they are correctly recognized. Furthermore, the error alarms from OCR will have less effect on longer words. If there is no title in either v or p, $Sim_T(v, p)$ is set to 0.

6.2. Content similarity

The content matching is similar to title matching. However, compared with titles, the size of characters in content is usually smaller and with lower visual quality. The recognition of content characters is thus less reliable. To avoid using the wrongly recognized characters for similarity measure, the content similarity is defined as

$$Sim_C(v, p) = \frac{\sum_{w_1 \in W_m} len(w_1)}{\sum_{w_2 \in W_p} len(w_2)} \tag{22}$$

where W_v, W_p and W_m have the similar meaning as Eq. (21); however, the words are not of the titles, but the content.

To reduce the amount of computation, content similarity of a shot and a page is performed only when the title similarity between them is below 0.7 or there is no title in the shot or page. The final similarity between a shot and a page is defined as the sum of the title and content similarities.

7. Experiments

We conduct experiments on five lecture videos taped in different classrooms. The duration of each video is about 45 to 60 min. The five videos consist of nine different presentations. All the external documents are prepared by speakers with PowerPoint. Basically a variety of master templates are used in different presentations. In the first three videos, most pages contain only texts. In the last two videos, most pages are mixed with texts, images, tables and figures. The flipping time of most pages involves less than 10 frames. During lectures, the documents are projected to the screen by an LCD projector. The lecturer can move freely in the class and make any gestures for presentation. The first three videos consist of one speaker

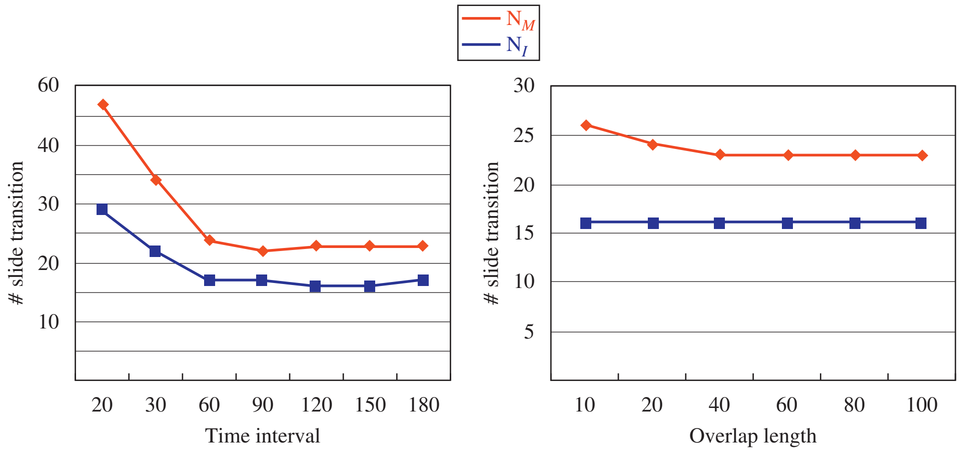


Fig. 9. Sensitivity of topic detection to different time intervals and overlap lengths. (N_M: number of missed transitions; N_I: number of falsely inserted transitions.)

Table 1
Results of topic detection

Lecture video	N_c	N_I	N_M	Accr.	Error rate	Precision	Recall
1	40	0	7	0.85	0.15	1.00	0.85
2	26	2	1	0.89	0.10	0.93	0.96
3	38	3	6	0.80	0.19	0.93	0.86
4	49	6	5	0.80	0.18	0.89	0.90
5	43	5	4	0.81	0.17	0.90	0.91

Table 2
Performance comparison of topic detection

Lecture video	Proposed approach		Frame diff.		Color Histo.	
	Precision	Recall	Precision	Recall	Precision	Recall
1	1.00	0.85	0.93	0.55	0.75	0.13
2	0.93	0.96	0.83	0.74	0.88	0.26
3	0.93	0.86	0.81	0.30	0.84	0.11
4	0.89	0.90	0.76	0.72	0.76	0.52
5	0.90	0.91	0.71	0.85	0.79	0.70

each, while the last two videos consist of three different presenters, respectively. Each speaker in the last two videos presents for approximately 20 min.

7.1. Topic detection

We compare and contrast our proposed topic detection approach with two other methods: frame difference [14] and color histogram difference [15]. All the tested approaches operate directly in the YCbCr color space of the MPEG domain. In the implementation, a slide shot transition is detected if the value of frame difference or color histogram difference is a local maximum. For the proposed approach, each time interval is composed of 120 DC image frames (5 s). Two adjacent time intervals are overlapped by 60 images. The time interval should be long enough so that a complete slide transition is included and the visual changes due to the transition are calculated. On the other hand, a large interval may contain more than one slide transition. The interval overlap is to avoid the slide transition happening at the boundary of an time interval which might be missing. In a presentation, it takes more than 30 s for the lecturer to present one page most time, and normally a slide transition can be finished in less than 10 frames. Thus, the parameter settings can deal with most presentations. We conduct experiments on the five videos with totally 219 slide transitions to test the sensitivity of parameter settings. Fig. 9 shows the numbers of missed and falsely inserted slide transitions for different time intervals and overlap lengths. As seen in Fig. 9, when the time interval and overlap can contain a complete slide transition, the result is not sensitive to these two parameters.

To evaluate the performance, we count the numbers of actual transitions N_T, falsely inserted transitions N_I, missed transitions N_M and correctly detected transitions N_C. The following performance measures are employed

$$\text{Recall} = \frac{N_C}{N_C + N_M}, \quad \text{Precision} = \frac{N_C}{N_C + N_I}$$

$$\text{Accuracy} = \frac{N_T - (N_M + N_I)}{N_T} = \frac{N_C - N_I}{N_T}$$

$$\text{Error rate} = \frac{N_M + N_I}{N_T + N_I} = \frac{N_M + N_I}{N_C + N_M + N_I}$$

The values of recall, precision and error rate are in the range of [0, 1]. Low recall values indicate frequent occurrence of false deletions, while low precision values indicate the frequent occurrence of false alarms. Error rate puts more penalty to false deletion than false insertion, meanwhile accuracy has negative value if $N_C < N_I$.

Table 1 shows the performance of our proposed approach, while Table 2 shows the comparison of the three approaches in terms of recall and precision. As indicated in the tables, our proposed method significantly outperforms the other two approaches. In the first three videos, both color histogram and frame difference approaches suffer from low recall. The former approach fails since the color of text captions and the design template of slides are similar. The latter approach, on the other hand, fails because it is equally sensitive to the difference between projected screens and the motion of foreground objects. As a result, local maxima may not be found when pages are flipped. In the last two videos, there are more images, figures and tables. The recall values of color histogram and frame difference are better compared with the previous three videos. Both approaches are effective when images are included in slides. Nevertheless, for figures that contain only lines and curves, and for tables that contain only texts, color histogram is not effective in detecting the changes.

The performance of our proposed approach, as shown in Table 2, is consistently better than the color histogram and frame difference methods. In most cases, whenever there are changes in figures or images, $\mathbf{E}_b$ will possess large value. Similarly, $\mathbf{E}_c$ will show large value whenever there is a change of texts. In the experiments, false insertions are caused by the sudden change of illumination. False deletions are mainly due to the low contrast between slides as well as the low resolution of video quality. Few transitions are not

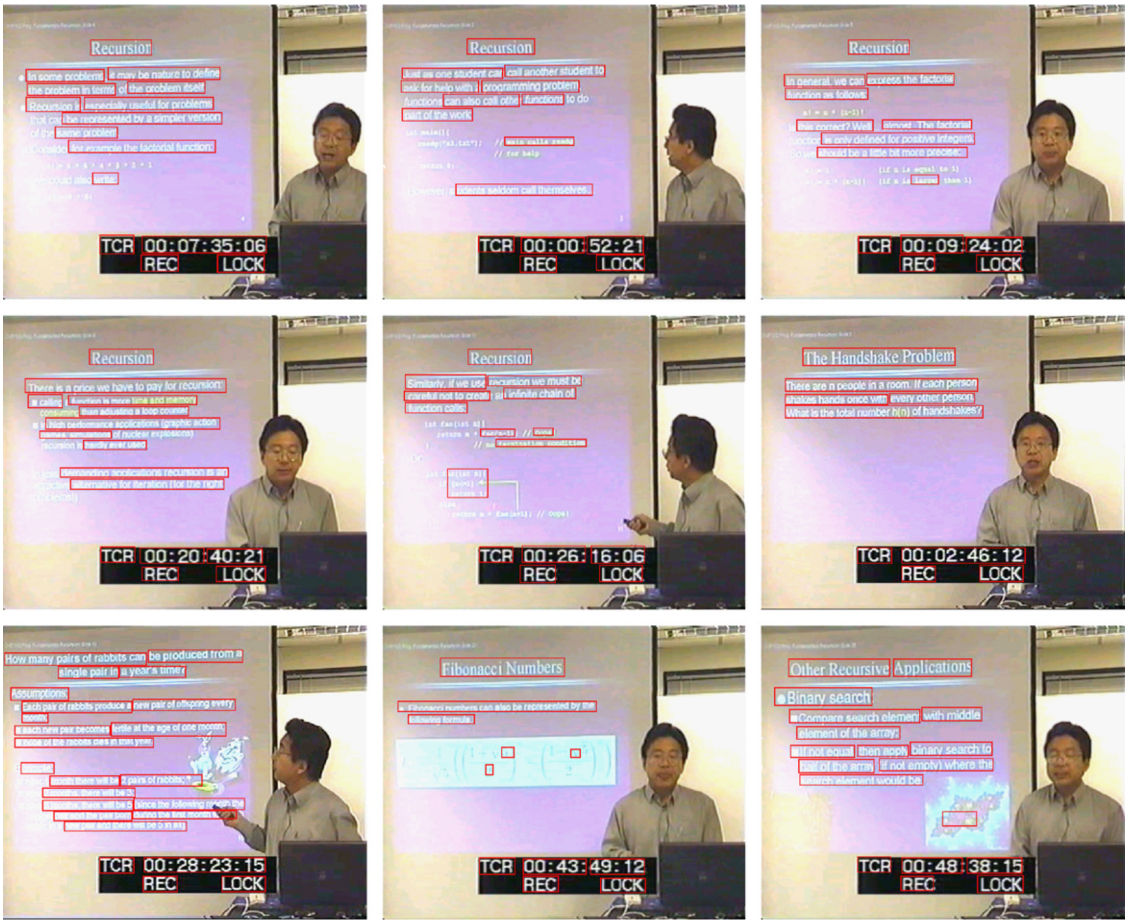
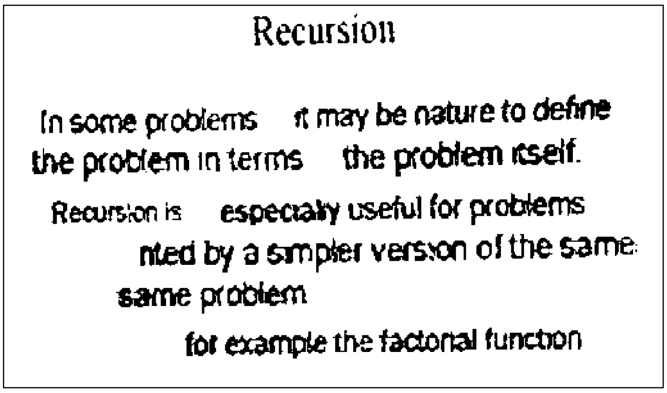


Fig. 10. Experimental results for video text detection.



Recursion

In some problems it may be nature to define

the problem in terms the problem itself.

Recursion is especially useful for problems

nted by a simpler version of the same

same problem

for example the factorial function

Fig. 11. Extracted high-resolution text of the top left image in Fig. 10.

detected because the main title of adjacent slides is same, while the main content is very similar. In the current implementation, the proposed approach can process approximately 70 frames per second on a Pentium-IV platform.

7.2. Video text recognition

For each shot in the videos, we evenly extract five keyframes along the time dimension. The number of keyframes can be changed. According to our experiment, 5–10 frames is considered as appropriate. The textboxes from multiple frames are integrated and reconstructed as one high-resolution textbox before text binarization. The binary textboxes are then fed to the OCR system. Fig. 10 shows the detected text boxes of several keyframes. We can see that when the background is not too complicated, the text detection algorithm works well. Some noise may be included if the text connects with other edges.

Fig. 11 shows the binarized high-resolution textboxes of a keyframe. Fig. 12 further shows some of the high-resolution textboxes obtained from keyframes in Fig. 10. In fact, either low or high-resolution, most of the characters in titles can be segmented correctly. The difference between two resolutions lies in two aspects: (i) the edges of high-resolution characters are much smoother; (ii) the adjacent characters are better separated in high resolution textboxes. These two factors can make great impact for OCR recognition. Compared with titles, the texts in content are usually more difficult to segment due to the small character size and over-illumination. Nevertheless, the results from high-resolution textboxes are much better than low-resolution ones.

To measure the performance of character recognition, we compute the value of *recall* (or accuracy) as N_c/N_g, where N_c is the number of characters recognized by OCR and N_g is the number of characters in the external documents. Tables 3 and 4 compare the OCR results for the high resolution texts reconstructed by our approach and the low resolution. The recognition accuracy for high-resolution titles is about 80% to 90%, much better than the accuracy of 20% to 50% for low resolution. The recognition of texts in the main content is a difficult task. In our experiment, due to the

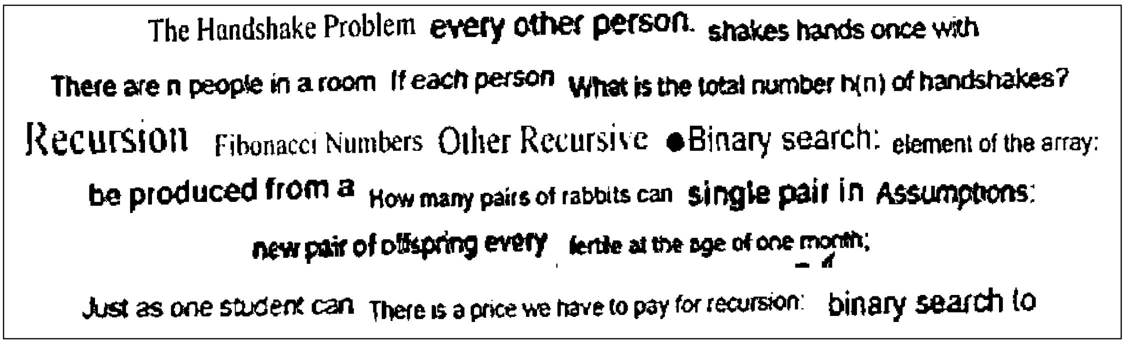

The Handshake Problem
every other person.
shakes hands once with
There are n people in a room
If each person
What is the total number h(n) of handshakes?
Recursion
Fibonacci Numbers
Other Recursive
●Binary search:
element of the array:
be produced from a
How many pairs of rabbits can
single pair in
Assumptions:
new pair of offspring every
fertile at the age of one month;
Just as one student can
There is a price we have to pay for recursion:
binary search to

Fig. 12. Some of the high-resolution text boxes extracted from the video frames shown in Fig. 10.

Table 3
Results of video text recognition (high resolution)

Lecture video	Title					Content					
	N_g	N_c	N_{ocr}	Recall	Precision	N_g	N_c	N_{ocr}	N_h	Recall	Precision
1	620	494	582	0.80	0.85	4117	432	1660	1586	0.10	0.26
2	230	218	230	0.95	0.95	3162	739	2037	1388	0.23	0.32
3	560	515	552	0.92	0.93	5058	1124	2029	1875	0.22	0.55
4	849	792	840	0.93	0.94	5282	1657	3549	2802	0.31	0.47
5	705	673	701	0.95	0.96	4238	1182	2578	2135	0.28	0.46

N_g: number of ground-truth characters, N_c: number of correctly recognized characters, N_{ocr}: number of characters output by OCR, N_h: number of characters recognized by human.

Table 4
Video text recognition (low resolution)

Lecture video	Title		Content	
	Recall	Precision	Recall	Precision
1	0.19	0.69	0.00	0.12
2	0.22	0.58	0.00	0.00
3	0.43	0.78	0.00	0.10
4	0.41	0.59	0.05	0.14
5	0.52	0.56	0.06	0.18

Table 5
Comparison of text recognition for different algorithms

Algorithm	Title		Content	
	Recall	Precision	Recall	Precision
Direct OCR	0.38	0.62	0.03	0.13
Y. Zhang [35]	0.50	0.70	0.12	0.21
H. Li et al. [36]	0.52	0.73	0.11	0.21
X. S. Hua et al. [38]	0.56	0.74	0.14	0.23
Our approach	0.91	0.93	0.24	0.43

low video quality, more than half of the characters are indeed not recognized by human. The OCR fails to recognize almost all the low resolution characters. However, approximately 30% to 60% of human-recognized characters are successfully recognized by the OCR when the high-resolution characters are reconstructed. In overall, about 10% to 30% of characters in the content of external documents are correctly recognized.

Table 5 compares OCR results for different algorithms on the 5 videos. Direct OCR approach is to feed the low-resolution textboxes to OCR directly. Linear interpolation [35] achieves some improvement for text recognition. The OCR performance is also improved by employing the algorithm in Ref. [36]. However, this improvement is mainly due to the linear interpolation phase in the algorithm, while the multi-frame integration contributes little. As discussed in Section 2, when both text and background are static, multiframe averaging does not help much for text enhancement. By selecting HCF and HCB, the algorithm in Ref. [38] performs a little better. Since there is not much difference between frames, further improvement is difficult. Our approach also employs linear interpolation, and the OCR performance is significantly improved after multi-frame integration by considering local color distribution.

Table 6
Synchronization results

Lecture video	Number of shots	Number of pages	Shots not in external documents	# of correctly matched shots (title/content)	Accuracy
1	48	23	16	27 (22/5)	0.84
2	28	17	5	18 (12/6)	0.78
3	45	27	8	34 (26/8)	0.92
4	55	46	0	51 (18/33)	0.93
5	48	43	0	47 (38/9)	0.98

The (title/content) in fifth column shows the number of pages that are matched by title and content.

7.3. Synchronization

In our approach, the effectiveness of synchronization is heavily dependent on the recognition of characters in videos. No temporal assumption or other visual feature such as shape or color is used for matching. The selected test videos were actually taped in different classrooms of varying settings. Table 6 shows the results of synchronizing video shots and external documents. Each video shot is linked with a document page that is most similar to it. The performance is evaluated by *accuracy* $= N_c/N_p$, where N_c is the number of shots that can be correctly linked, and N_p is the total number of shots that capture the prepared document pages.

In the first three videos, because the presented lectures are programming courses, some shots contain only the snapshots of programming codes shown in Borland C++. These programming codes are given in class on the fly and thus not included in the set of prepared documents. In the experiments, we set one confidence level, based on the sum of title similarity in Eq. (21) and content similarity in Eq. (22), to exclude the shots without corresponding slides. A shot is excluded from linking if the sum of title and content similarities with the most similar page is below the confidence level. In the experiments, we successfully filter all the shots showing Borland C++ code snippets.

In the tested videos, notice that the number of shots is larger than the number of actual pages. This is simply because some pages are shown more than once by flipping backward and forward during the presentation. Thus, multiple shots may be linked to one page. The backward transitions or random access of pages in time will not cause problems since we do not assume temporal smoothness in matching.

As indicated in Table 6, by using video text for synchronization, we can achieve the accuracy of approximately 80% to 100% for the five tested videos. The number of shots that are matched by title similarity and by content similarity are also indicated in the table. In overall, about 66% of shots are matched by titles, while 34% of shots are matched by contents. In the experiment, few shots are mismatched due to: (i) the titles or contents of some pages are similar, (ii) not enough texts are extracted or recognized, especially for those pages without titles, the texts from main contents are too few for matching. The experimental results indicate that the effect of lighting can influence the results of video text analysis. However, as long as 80% of the characters in titles and 10% to 30% of the characters in contents can be recognized, the shots and pages are most likely to be linked correctly.

8. Conclusion

We have presented our end-to-end approaches for the structuring and indexing of lecture video content. The novelty of our approach lies in the utilization of scene texts embedded in videos for topical event detection, content recognition and matching. The proposed algorithm for slide shot transition detection is effective and capable of operating in real time. Currently, our approach cannot handle the slide shows with animation. To avoid the spurious detection of shots due to animation, techniques similar to the traditional shot boundary detectors [14,15], e.g., dissolve and wipe detectors, need to be developed. The proposed video text analysis is robust when incorporating with super-resolution reconstruction. Experimental results indicate that the matching of shots and external documents solely based on textual information are effective. Apparently, the accuracy of matching can be further improved particularly for the pages with little text and more graphics, if other cues like the spatial layout and color are jointly considered. Speech is another important modality for lecture video indexing. For instance, when the oral presentation is not aligned with pages being shown, speech can be integrated with text to further improve our work in this paper. Semantic information, instead of string matching, needs to be employed for the content matching of speech and external documents. Because the text and speech recognition, in general, play a critical role in this kind of applications, the content guided approaches such as the utilization of textual information in external documents can also be exploited to tolerate possible errors made during recognition.

Acknowledgement

The work described in this paper was partially supported by the grants DAG01/02.EG16, HIA01/02.EG04, SSRI99/00.EG11, and a grant from the Research Grants Council of the Hong Kong Special Administrative Regions, China (CityU 118906).

References

[1] J.Y. Chen, C.A. Bouman, J.C. Dalton, Hierarchical browsing and search of large image databases, IEEE Trans. Image Process. (2000) 442–455.

[2] T.C.T. Kuo, A.L.P. Chen, Content based query processing for video databases, IEEE Trans. Multimedia (2000) 1–13.

[3] S.F. Chang, W. Chen, H. Meng, H. Sundaram, VideoQ: an automated content based video search system using visual cues, ACM Multimedia (1997).

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, in: Computer, 1995, pp. 23–32.

[5] F. Wang, C.W. Ngo, T.C. Pong, Gesture tracking and recognition for lecture video editing, in: International Conference on Pattern Recognition, 2004.

[6] G.D. Abowd, C.G. Atkeson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney, M. Tani, Teaching and learning as multimedia authoring: the classroom 2000 project, ACM Multimedia (2000) 187–198.

[7] S.G. Deshpande, J.-N. Hwang, A real-time interactive virtual classroom multimedia distance learning system, IEEE Trans. Multimedia 3 (4) (2001) 432–444.

[8] S.X. Ju, M.J. Black, S. Minneman, D. Kimber, Summarization of videotaped presentations: automatic analysis of motion and gesture, IEEE Trans. Circuits and Syst. Video Technol. 8 (5) (1998) 686–696.

[9] T.F.S. -Mahmood, Indexing for topics in videos using foils, in: International Conference on Computer Vision and Pattern Recognition, 2000, pp. 312–319.

[10] S. Mukhopadhyay, B. Smith, Passive capture and structuring of lectures, ACM Multimedia (1999).

[11] L.A.
Rowe, J.M. Gonzlez, BMRC Lecture Browser ⟨http://bmrc.berkeley.edu/frame/projects/lb/index.html⟩.

[12] D. Phung, S. Venkatesh, C. Dorai, High level segmentation of instructional videos based on content density, ACM Multimedia (2002).

[13] T. Liu, R. Hjelsvold, J.R. Kender, Analysis and enhancement of videos of electronic slide presentations, in: International Conference on Multimedia and Expo, 2002.

[14] B.L. Yeo, B. Liu, Rapid scene analysis on compressed video, IEEE Trans. Circuits Syst. Video Technol. 5 (6) (1995) 533–544.

[15] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, Automatic partitioning of full-motion video, ACM Multimedia Syst. 1 (1) (1993) 10–28.

[16] B. Erol, J.J. Hull, D.S. Lee, Linking multimedia presentations with their symbolic source documents: algorithm and applications, ACM Multimedia (2003).

[17] H. Aradhye, C. Dorai, J.-C. Shim, Study of embedded font context and kernel space methods for improved videotext recognition, IBM Research Report RC 22064, 2001.

[18] R. Lienhart, Automatic text segmentation and text recognition for video indexing, Multimedia Syst. Mag. 8 (2000) 69–81.

[19] X. Chen, J. Yang, J. Zhang, A. Qaibel, Automatic detection of signs with affine transformation, IEEE Workshop on Application of Computer Vision, December 2002.

[20] X. Chen, J. Yang, J. Zhang, A. Qaibel, Automatic detection and recognition of signs from natural scenes, IEEE Trans. Image Process. 13 (1) (2004).

[21] J.C. Shim, C. Dorai, R. Bolle, Automatic text extraction from video for content-based annotation and retrieval, in: International Conference on Pattern Recognition, 1998.

[22] X.S. Hua, W. Liu, H.J. Zhang, Automatic performance evaluation for video text detection, in: International Conference on Document Analysis and Recognition, 2001, pp. 545–550.

[23] H. Li, D. Doerman, O. Kia, Automatic text detection and tracking in digital video, IEEE Trans. Image Process. 9 (1) (2000).

[24] R. Lienhart, Localizing and segmenting text in images and videos, IEEE Trans. Circuits Syst. Video Technol. 12 (4) (2002).

[25] Y. Zhong, H.J. Zhang, A.K. Jain, Automatic caption localization in compressed video, IEEE Trans. Pattern Anal. Mach. Intell. 22 (4) (2000) 385–392.

[26] O.D. Trier, A. Jain, Goal-directed evaluation of binarization methods, IEEE Trans. Pattern Anal. Mach. Intell. 17 (2) (1995) 1191–1201.

[27] C. Wolf, J.M. Jolion, F. Chassaing, Text localization, enhancement and binarization in multimedia documents, in: International Conference on Pattern Recognition, 2002, pp. 1037–1040.

[28] D. Chen, J.M. Odobez, H. Bourlard, Text detection and recognition in images and video frames, Pattern Recognition 37 (3) (2004).

[29] E.K. Wong, M. Chen, A new robust algorithm for video text extraction, Pattern Recognition 36 (2003) 1397–1406.

[30] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, Video OCR for digital news archive, in: ICCV Workshop on Image and Video Retrieval, 1998.

[31] ⟨http://vireo.cs.cityu.edu.hk/LEdit/htdocs/Synchronization.htm⟩.

[32] S. Baker, T. kanade, Limits on super-resolution and how to break them, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002).

[33] M. Ben-Ezra, A. Zomet, S.k. Nayar, Video super-resolution using controlled subpixel detector shifts, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005).

[34] E. Shechtman, Y. Caspi, M. Irani, Space-time super-resolution, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005).

[35] Y. Zhang, T.S. Chua, Detection of text captions in compressed domain video, ACM Multimedia Workshop, 2000, pp. 201–204.

[36] H. Li, D.S. Doerman, Text enhancement in digital video using multiple frame integration, ACM Multimedia (1999) 19–22.

[37] T. Sato, T. Kanade, E.K. Huges, M.A. Smith, S. Satoh, Video COR: indexing digital NRES libraries by recognition of superimposed caption, ACM Multimedia Syst. 7 (5) (1999) 385–395.

[38] X.S. Hua, P. Yin, H.J. Zhang, Efficient video text recognition using multiple frame integration, in: International Conference on Image Processing, 2002.

[39] OmniPage Pro 12 ⟨http://www.scansoft.com/omnipage/⟩.

[40] E. Ukkonen, Algorithms for approximate string matching, Inform. Control 100–118 (1985).

About the Author—FENG WANG received his PhD in Computer Science from the Hong Kong University of Science and Technology in 2006. He received his BSc in Computer Science from Fudan University, Shanghai, China, in 2001. Now he is a Research Fellow in the Department of Computer Science, City University of Hong Kong. Feng Wang's research interests include multimedia content analysis, pattern recognition, and IT in education.

About the Author—CHONG-WAH NGO (M'02) received his Ph.D in Computer Science from the Hong Kong University of Science & Technology (HKUST) in 2000. He received his MSc and BSc, both in Computer Engineering, from Nanyang Technological University of Singapore in 1996 and 1994, respectively.

Before joining City University of Hong Kong as assistant professor in Computer Science department in 2002, he was a postdoctoral scholar in Beckman Institute of University of Illinois in Urbana-Champaign (UIUC). He was also a visiting researcher of Microsoft Research Asia in 2002. CW Ngo's research interests include video computing, multimedia information retrieval, data mining and pattern recognition.

About the Author—TING-CHUEN PONG received his Ph.D. in Computer Science from Virginia Polytechnic Institute and State University, USA in 1984. He joined the University of Minnesota—Minneapolis in the US as an Assistant Professor of Computer Science in 1984 and was promoted to Associate Professor in 1990. In 1991, he joined the Hong Kong University of Science & Technology, where he is currently a Professor of Computer Science and Associate Vice-President for Academic Affairs. He was an Associate Dean of Engineering at HKUST from 1999 to 2002, Director of the Sino Software Research Institute from 1995 to 2000, and Head of the W3C Office in Hong Kong from 2000 to 2003. Dr. Pong is a recipient of the HKUST Excellence in Teaching Innovation Award in 2001.

Dr. Pong's research interests include computer vision, image processing, pattern recognition, multimedia computer, and IT in Education. He is a recipient of the Annual Pattern Recognition Society Award in 1990 and Honorable Mention Award in 1986. He has served as Program Co-Chair of the Web and Education Track of the Tenth International World Wide Web Conference in 2001, the Third Asian Conference on Computer Vision in 1998, and the Third International Computer Science Conference in 1995.