Simulating a Smartboard by Real-Time Gesture Detection in Lecture Videos

Feng Wang, Chong-Wah Ngo, *Member, IEEE*, and Ting-Chuen Pong

***Abstract*—Gesture plays an important role for recognizing lecture activities in video content analysis. In this paper, we propose a real-time gesture detection algorithm by integrating cues from visual, speech and electronic slides. In contrast to the conventional "complete gesture" recognition, we emphasize detection by the prediction from "incomplete gesture". Specifically, intentional gestures are predicted by the modified hidden Markov model (HMM) which can recognize incomplete gestures before the whole gesture paths are observed. The multimodal correspondence between speech and gesture is exploited to increase the accuracy and responsiveness of gesture detection. In lecture presentation, this algorithm enables the on-the-fly editing of lecture slides by simulating appropriate camera motion to highlight the intention and flow of lecturing. We develop a real-time application, namely simulated smartboard, and demonstrate the feasibility of our prediction algorithm using hand gesture and laser pen with simple setup without involving expensive hardware.**

***Index Terms*—Gesture detection, lecture video, real-time simulated smartboard.**

I. Introduction

In the past few years, multimedia in education has attracted numerous research attentions. With the aid of hardware and software, researchers attempt to find innovative ways of teaching, and thus enhance the quality of learning. Typical demonstrated systems include Classroom 2000 [1], BMRC lecture browser [20], and SCORM [22]. Due to the popularity of e-learning, lecture videos are widely available for online access. A lecture is captured by different devices equipped in the classroom, such as video cameras, microphones and electronic board. The recorded data is then used for offline video editing or online broadcasting. To automate this process, video content analysis is usually essential to understand the activities during the lecture.

Numerous issues have been addressed for the content analysis of lecture or instructional videos. These issues include topical detection, synchronization, gesture detection, pose estimation, video summarization, and editing. In topical detection, a lecture video is structured according to the topics of discussion by audio [13], visual [10], [13], [16], text [25] or cinematic expressive cues [19]. The detected topics are synchronized (or linked) with external documents for effective indexing, retrieval and editing [4], [10], [16], [25]. To facilitate browsing and summarization, keyframes [9], mosaics [11] and statistical highlights [2], [7] are also extracted. Texts in the projected slides [25] and whiteboard [8] are detected and recognized to be aware of the content under discussion. Gesture detection [3], [26] and pose estimation [18], [27] are employed to recognize the presenter's activities. Based on the results of content analysis, lecture videos are then edited in either offline [5], [29] or online manners [21], [32], [18] to better deliver the quality of teaching and learning.

Advances in video content analysis have brought us opportunities to create not only efficient ways of learning, but also convenient tools for presentation. In this paper, we first propose a gesture prediction algorithm by fusing multimodality cues including visual, speech, and slides. The algorithm is then applied for the real-time simulation of smartboard where the lecture slides projected on the screen are on-the-fly edited by simulated camera motion and gesture annotations. The editing aims to capture the flow of lecturing naturally without involving heavy hardware setup.

Gesture is a popular kind of lecture activity. Most gestures used by presenters belong to *dietic gestures* to direct the audiences' attentions to the content under discussion. In this context, gesture has been addressed in [3], [12], [5], and [18]. Most of them employ simple features such as frame difference and skin-color for gesture detection. In [3], frame difference is employed to detect gestures present in the slide region. Once a gesture is detected, appropriate commands are sent to a camera on the floor to zoom in/out. In [12], although no specific gesture detection is mentioned, some gestures can be noticed by the frame difference calculated by a wide angle camera to control another camera for video capture. In [18], an algorithm is proposed to keep detecting three skin-color blocks in the whiteboard region, which are assumed to be the presenter's face and hands. Gestures are extracted according to the relative locations of the three skin-color blocks.

The challenge in gesture detection is that there is no salient feature to describe the hand in the video due to the small size and irregular shape. Any simple or single feature is not robust enough. Frame difference can be triggered by any moving objects, while some noisy skin-similar colors are usually introduced. In addition, intentional gestures are always intertwined with non-gesture movement of hands, which makes existing approaches less tolerant to noise. For instance, a presenter can move freely in front of class and thus the hand may interact with the slide even when there is no dietic gesture pointing to the screen. This indeed results in the difficulty of determing the intention or the information to be conveyed by the gesture.

Manuscript received April 24, 2007; revised March 9, 2008. First published June 13, 2008; last published July 9, 2008 (projected). This work was supported in part by Grants DAG01/02.EG16, HIA01/02.EG04, SSRI99/00.EG11, and by a grant from the Research Grants Council of the Hong Kong Special Administrative Regions, China (CityU 118906). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hayder Radha.

F. Wang and T. C. Pong are with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: fwang@cs.cityu.edu.hk; tcpong@ust.hk).

C. W. Ngo is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong e-mail: cwngo@cs.cityu.edu.hk).

Digital Object Identifier 10.1109/TMM.2008.922871

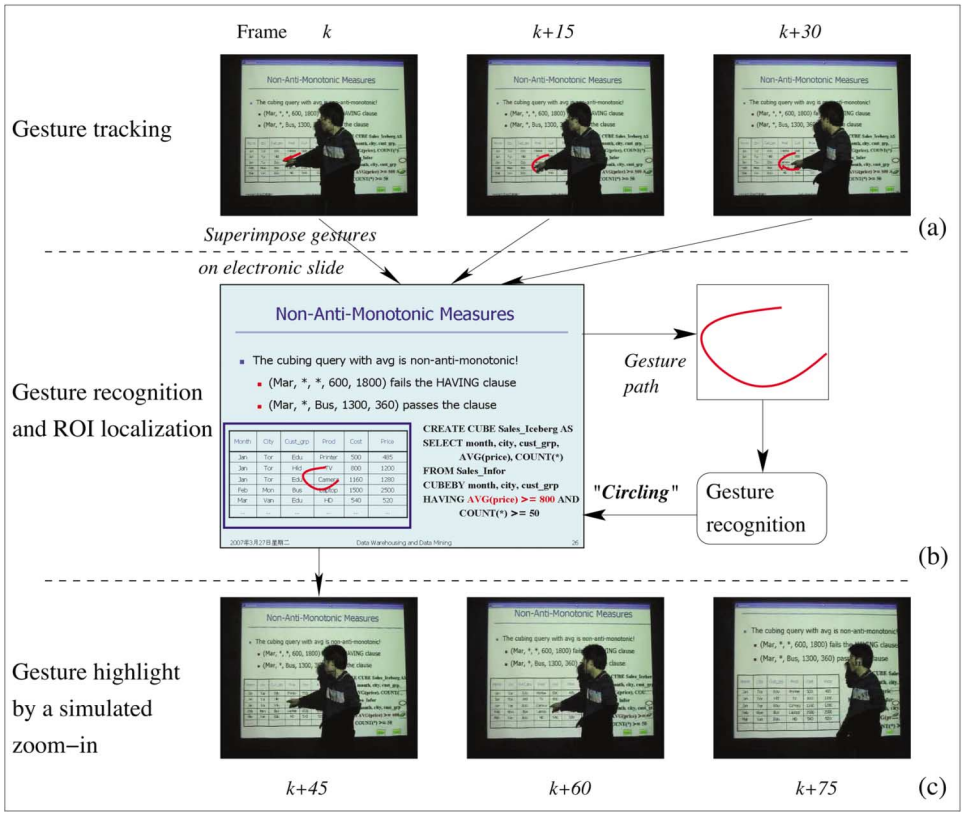

Fig. 1. Real-time simulated smartboard based on gesture prediction.

In [26] and [29], we have proposed an algorithm for off-line gesture detection and recognition in lecture videos. To be robust, frame difference and skin color are combined to detect and track the candidate gestures. Gesture recognition is then employed to recognize and extract intentional gestures from the gesture paths. Here we define a gesture as an intentional interaction between the presenter's hand and some object instances (e.g., a paragraph or a figure) in the slide. Three kinds of gestures (*lining*, *circling*, and *pointing*) are recognized by three hidden Markov models (HMMs). By gesture recognition, we extract intentional gestures and eliminate non-gesture movements. A gesture is detected only when it is verified by gesture recognition.

In this paper, we address the detection and recognition of gestures in lecture videos for real-time applications. The major difference between this work and [26] lies in two aspects. First, to cope with the efficiency requirements of real-time applications, the responsiveness and accuracy of gesture detection are jointly considered. For example, in an automatic camera management system, when a gesture is present, a camera is expected to focus on the region of interaction. The action from the camera should be as rapid as possible so that the interaction can be captured and used for editing in real-time. In [26], [29], for off-line video editing, a gesture is verified when the full gesture path is completed, which is too late for the camera to respond to the gesture. To tackle the responsiveness problem, our new algorithm predicts and reacts as early as possible before a gesture path is fully observed. Second, besides visual cue, we combine speech and electronic slides to improve the accuracy and reduce the response time of gesture detection.

Gesture in lecture videos has been proven to be a very important and useful hint in recognizing the lecture activities for offline [5], [29] or online [21], [32], [18] video editing to provide the students with an efficient and effective way of learning. However, how to employ gestures and other lecture activities to create tools for teaching has seldom been attempted before. In [24], an experimental system called *magic boards* is proposed. Given a lecture video with some interactions around a chalkboard, the user who is familiar with the interactions is presented with a small set of images. Each image represents a single idea written on the board. The user then replaces each image with a picture, video or animation to better represent each idea. A new video is rendered using the replaced pictures and videos in place of the writing on the board. Due to the offline nature, the system cannot be used in course of the presentation. The whole procedure is manual, which could be highly time-consuming.

Based on our initial work in [28] for gesture prediction, we propose and develop a simulated smartboard for real-time lecture video editing. With the smartboard, a presenter can interact, by hand or using a laser pen, with the projected slides to generate a novel view of the slide. The interactions include annotating the slides, controlling the slide show, and highlighting specific compound objects in slides. With response to the interactions, the projected slides are sort of edited automatically by analyzing the video streams captured by a camera mounted at the back of the classroom. Technically, the incomplete gestures in videos are first detected illusively through prediction. The appropriate actions such as camera zoom and superimposition of gesture annotation are then simulated and projected to the screen on-the-fly. This process is illustrated in Fig. 1.

In Fig. 1(a), a candidate gesture interacting with the slide is detected and tracked. In Fig. 1(b), the gesture path is then superimposed on the electronic slide and recognized as "circling" gesture. The region-of-interest (ROI) pointed by the gesture is localized. In the next few frames, the ROI is highlighted by a simulated zoom-in on-the-fly as shown in Fig. 1(c). Due to the real-time capability, the smartboard can be used during presentation and simulate most functions of a real smartboard. Compared to the high expenditure of a smartboard, our setup becomes relatively simple and economic with the content-based solution.

The remainder of this paper is organized as follows. In Section II, we revise the HMMs for complete gesture recognition in our previous work [26] to recognize incomplete gestures. In Section III, speech and electronic slides are fused with visual cue to improve the accuracy and responsiveness of gesture detection. Section IV describes the design and implementation of the simulated smartboard. Experimental results are shown in Section V. Finally, Section VI concludes this paper.

II. Gesture Detection by Visual Cue

To extract intentional gestures, three kinds of gestures frequently used by presenters are defined: *lining, circling,* and *pointing*. Intentional gestures are then recognized and extracted by gesture recognition. Similar to our previous works [26], [29], skin-color and frame difference are fused as evidence for candidate gesture detection and tracking. In [29], gesture recognition is activated after the full gesture path is observed, without considering the responsiveness constraint. In this section, we modify the HMMs in [26] and [29] to recognize the incomplete gestures so that gestures can be detected and verified earlier to cope with the efficiency requirements for real-time applications.

A. HMM for Complete Gesture Recognition

HMM has been proven to be useful in sign gesture recognition [23]. A detailed tutorial on HMM can be found in [31]. In [26], we employ HMM to recognize the dietic gestures in lecture videos. Three HMM models are trained to recognize the three defined gestures.

Given an observation $O = (o_1, o_2, \ldots, o_T)$, where each $o_i (i = 1, 2, \ldots, T)$ is a sampled point on the gesture path, gesture recognition problem can be regarded as that of computing

$$\arg\max_i \{P(g_i \mid O)\} \tag{1}$$

where g_i is the ith gesture. By using Bayes' Rule, we have

$$P(g_i \mid O) = \frac{P(O \mid g_i) P(g_i)}{P(O)}. \tag{2}$$

Thus, for a given set of prior probabilities $P(g_i)$, the most probable gesture depends only on the likelihood $P(O \mid g_i)$. Given the dimensionality of the observation sequence O, the direct estimation of the joint conditional probability $P(o_1, o_2, \ldots \mid g_i)$ from examples of gestures is not practicable. However, if a parametric model of gesture production such as a Markov model is assumed, then estimation from data is possible since the problem of estimating the class conditional observation densities $P(O \mid g_i)$ is replaced by the much simpler problem of estimating the Markov model parameters.

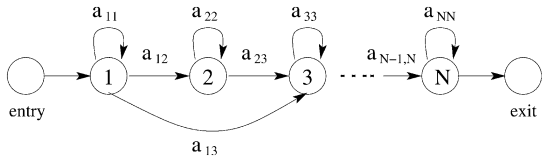

Fig. 2. HMM for complete gesture recognition.

In HMM-based gesture recognition, it is assumed that the observation sequence corresponding to each gesture is generated by a Markov model as shown in Fig. 2. A Markov model is a finite state machine which changes state once for every observation point. The joint probability that O is generated by the model M moving through the state sequence $X = x_1, x_2, \ldots, x_T$ is calculated simply as the product of the transition probabilities and the output probabilities

$$P(O, X \mid M) = \prod_{t=1}^{T} b_{x_t}(o_t) a_{x_t x_{t+1}} \tag{3}$$

where $b_{x_t}(o_t)$ is the probability that o_t is observed in state x_t, and $a_{x_t x_{t+1}}$ is the probability that the state transition from x_t to x_{t+1} is taken. However, in practice, only the observation sequence O is known and the underlying state sequence X is hidden. This is why it is called a *Hidden Markov Model*. Given that X is unknown, the required likelihood is computed by summing over all possible state sequences, that is

$$P(O \mid M) = \sum_X \prod_{t=1}^{T} b_{x_t}(o_t) a_{x_t x_{t+1}}. \tag{4}$$

In [31], by using Bayes' Rule and approximation, (1) is deduced to calculating the likelihood

$$\tilde{P}(O \mid M) = \max_X \left\{ \prod_{t=1}^{T} b_{x_t}(o_t) a_{x_t x_{t+1}} \right\} \tag{5}$$

where $X = x_1, x_2, \ldots, x_T$ is the state sequence that O moves through the model M. Notice that if (1) is computable, then the recognition problem is solved. Given a set of HMM models $\{M_i\}$ corresponding to gestures $\{g_i\}$, (1) is solved by using (2) and assuming that

$$P(O \mid g_i) = P(O \mid M_i). \tag{6}$$

In HMM models, a_{ij} and $b_i(\cdot)$ are estimated by Baum-Welch algorithm in the training phase. For gesture recognition, (5) is calculated by employing the Viterbi algorithm. The details of the two algorithms can be found in [31].

B. Modified HMM for Incomplete Gesture Recognition

The HMM models introduced in Section II-A are used for gesture recognition in offline systems. Each gesture is recognized after the complete path is observed. However, this is too late in some real-time applications. For instance, in a system of automatic camera control for lecture capture, once a gesture is present, the camera is expected to zoom in on the corresponding region to highlight the interaction in a short period. For real-time

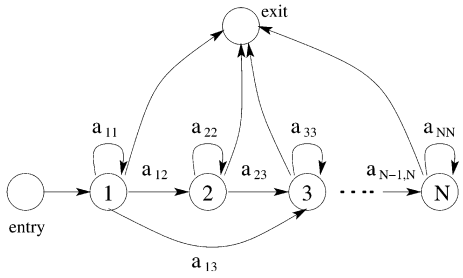

Fig. 3. Modified HMM for incomplete gesture recognition.

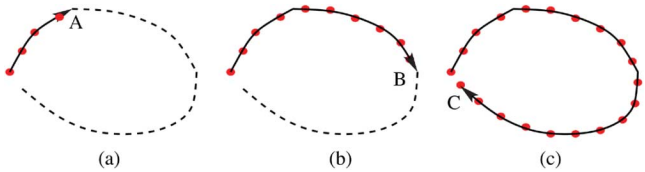

Fig. 4. (a), (b) Incomplete circling gestures. (c) Complete circling gesture.

processing, besides accuracy, the response time is an important criterion. To respond to the presenter's gestures as soon as possible, gesture recognition and verification need to be carried out before the gesture is finally completed. We modify the HMM models in Section II-A to predict and recognize incomplete gestures.

Different from complete gestures, an incomplete gesture usually cannot move through the HMM model to the state *exit* in Fig. 2, but just reaches one of the intermediate states $1, 2, \ldots, N$. Given an observation O which corresponds to an incomplete gesture, the incomplete gesture recognition can be regarded as that of computing

$$\arg\max_{i,S} P(g_{i,S} \,|\, O) \tag{7}$$

where S is the last intermediate state that O reaches when it goes through the HMM model corresponding to gesture g_i. In the topology of HMM model shown in Fig. 2, the final nonemitting state *exit* can be reached only through the last state N, which means a complete gesture must reach the state N before it is completed. To recognize incomplete gestures that may stop at any intermediate state, we modify the HMM models by adding a forward state transition from each intermediate state to the state *exit* in Fig. 3. The joint probability that O moves through an HMM model M and stops at an intermediate state S can be approximated by

$$\tilde{P}(O \,|\, M, S) = \max_X \left\{ \prod_{t=1}^{S} b_{x_t}(o_t) a_{x_t x_{t+1}} \right\} \tag{8}$$

where x_{S+1} is the *exit* state.

Fig. 4 shows the trajectory of a circling gesture. Fig. 4(c) is the complete gesture, and Fig. 4(a) and (b) are two incomplete gestures stopping at different intermediate states. By comparing Fig. 4(a) and (b), we are more confident that the current observation will compose a gesture if it moves further through the HMM model or stops at a state nearer to the state N. Based on this finding, we take into account the stopping state for the probability calculation and modify (8) to be

$$\tilde{P}'(O \,|\, M, S) = \max_X \left\{ \left(\prod_{t=1}^{S} b_{x_t}(o_t) a_{x_t x_{t+1}} \right) \cdot e^{\frac{S}{N}} \right\} \tag{9}$$

where $e^{(S/N)}$ is introduced to assign a higher probability to the gesture when the stopping state S is nearer to the final state N. Let $a'_{ij} = a_{ij} e^{(j-i/N)}$, and (9) can be rewritten as

$$\tilde{P}'(O \,|\, M, S) = \max_X \left\{ \prod_{t=1}^{S} b_{x_t}(o_t) a'_{x_t x_{t+1}} \right\}. \tag{10}$$

Thus, the Viterbi algorithm can still be used for the calculation of (10). The probability that O is an incomplete gesture modeled by M is

$$\begin{aligned} \tilde{P}(O \,|\, M) &= \max_S \tilde{P}'(O \,|\, M, S) \\ &= \max_{X,S} \left\{ \prod_{t=1}^{S} b_{x_t}(o_t) a'_{x_t x_{t+1}} \right\}. \end{aligned} \tag{11}$$

Equation (1) can be solved by Bayes' Rule and assuming that

$$P(O \,|\, g_i) = \tilde{P}(O \,|\, M_i). \tag{12}$$

C. Gesture Verification by Recognition

Given an observation sequence O (incomplete or complete gesture path), for the three defined gestures, three confidence values are calculated based on the modified HMMs that indicate how likely O will be a gesture g_i

$$C_i = P(g_i, O). \tag{13}$$

C_i values are used to verify whether O is an intentional gesture or not. A confidence value C_{visual} on the presence of a gesture is calculated by visual cue as

$$C_{\text{visual}} = \frac{C_{\max}}{\sum_i C_i} \cdot C_{\max} \tag{14}$$

where $C_{\max} = \max_i C_i$. When an incomplete gesture is moving on, C_{visual} is calculated for each sample point. To determine the best point to predict, a gating parameter is required. Due to the tradeoff between responsiveness and accuracy, finding an optimal parameter is application tailored. We determine the gating parameter C_{gate} empirically, and start gesture verification whenever $C_{\text{visual}} > C_{\text{gate}}$. In general, the further O moves through the corresponding HMM model, the more confident that there is a gesture; however, the longer response time is required.

III. Gesture Detection by Fusing Visual, Speech, and Electronic Slides

The proposed gesture prediction from incomplete observation is mainly relying on visual cue. In lecture videos, speech also provides complementary hints for detecting the presence of gestures. In this section, we first describe the multimodal relations among visual, speech and slides. Then we present our

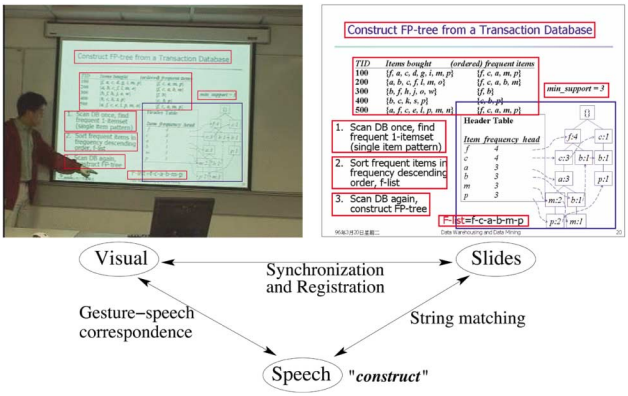

Fig. 5. Relations between multimodal cues.

approach to employ these relations to improve the accuracy and responsiveness of gesture detection by combining speech and semantic information extracted from electronic slides.

A. Relations Between Visual, Speech and Slides

Imagine the scenario where a presenter interacts with the projected slides while spelling out keywords. In a multimodal environment, ideally we can sample the gesture and speech, then verify the gesture with the words found in electronic slides to determine the region of focus. By jointly considering these cues, both the accuracy and response time of gesture prediction can be improved. The scenario is illustrated in Fig. 5. When a gesture is pointing to a paragraph in the slide, a keyword "*construct*" is found in speech, which lies in the paragraph interacting with the gesture. This case frequently happens during a lecture since the presenter tends to read out the keywords to be highlighted. We exploit such correspondence between gesture and speech to boost the performance of prediction. Fig. 5 models the relations between three multimodal cues. Gesture-speech correspondence plays an essential role in our task. It is worth noticing that the correspondence cannot be directly estimated, but rather relies on the semantic information from slides which indirectly bridges the gesture-speech relationship. Visual-slide correspondence recovers the geometric alignment between video and slide, while speech-slide correspondence matches keywords found in voice and text sources. By mapping both speech and gesture to slides, the correspondence between them are indirectly estimated. In visual-slide correspondence, the alignment is achieved through the synchronization of videos and slides with video text analysis based on our work in [25], [26]. Basically texts from both video and slides are extracted for matching, and the spatial correspondence is recovered by computing homography. In speech-slide correspondence, speech are online recognized and matched with words available in current slides with edit distance.

B. Gesture-Speech Correspondence for Gesture Detection

Two pieces of information are available: video text interacting with gesture through visual-slide alignment, and speech transcript generated by an automatic speech recognition (ASR) engine. By matching both information with slides respectively, the ROI where they coincide can be estimated. In our approach, the layout of each electronic slide is first structured into groups of semantic instance. Due to visual-slide correspondence, the semantic region that a recognized gesture highlights can be located. The words found in the region are then matched against the speech transcript. A confidence value is computed to hint the existence of gesture-speech correspondence. If such correspondence is found, there is more likely an intentional gesture present and thus boosts the accuracy and responsiveness of gesture detection.

With the PowerPoint slide in Fig. 5 as an example, the layout is semantically grouped into separate object instances. By constructing the one-to-one mapping between the instances in slides and videos through homography projection, we can easily organize and structure the layout of videos [29]. In Fig. 5, the projected slide region is partitioned into different regions-of-interest (ROIs). Each ROI is a unit that gesture may interact with at any instance of time. The video texts in ROIs are known since the mapping between ROIs and semantic instances in the electronic slide is aware of. Whenever a candidate gesture is interacting with an ROI, we search the matches between the words in ROI and speech transcript. Since the significance of matched words can vary, we weight the importance of words with traditional information retrieval techniques. Initially, the methods of stemming and stop word removal are applied to the texts in every slide. By treating each ROI as a document,term frequency (TF) and inverse document frequency (IDF) are computed for the remaining keywords to distinguish different ROIs. A confidence value C_{speech} is calculated based on the the TF and IDF values of the keywords as

$$C_{\text{speech}} = \sum_{w \in W} \frac{\log(1 + \text{TF}_w \cdot \text{IDF}_w)}{1 + \Delta t_w} \tag{15}$$

where W is the set of matched keywords, and Δt_w is the time interval between the presence of the gesture and of the keyword w in speech. Because the gesture and speech are not always temporally aligned, we also consider speech at time before and after the gesture is predicted. For robustness, Δt_w is used to degrade the importance of words which are not synchronized with gesture.

C. Gesture Detection by Fusing Speech and Visual

If the correspondence between visual and speech discussed in Section III-B can be detected, we are more confident that there is a gesture present, and thus speech can be combined with visual for gesture verification. When a candidate gesture is detected by visual cue, we keep tracking it and calculating the visual evidence C_{visual} in (14). Meanwhile the transcripts of speech are generated by ASR. We search the keywords in the transcripts that match with the texts in the ROI interacting with the candidate gesture. Once a correspondence between speech and visual is detected, C_{speech} is then calculated. The visual-speech confidence C is computed and fused as

$$C = \lambda_v C_{\text{visual}} + \lambda_s C_{\text{speech}} \tag{16}$$

where $\lambda_v + \lambda_s = 1$ and both parameters are weights to linearly rationale the significance of visual and speech. Since the quality

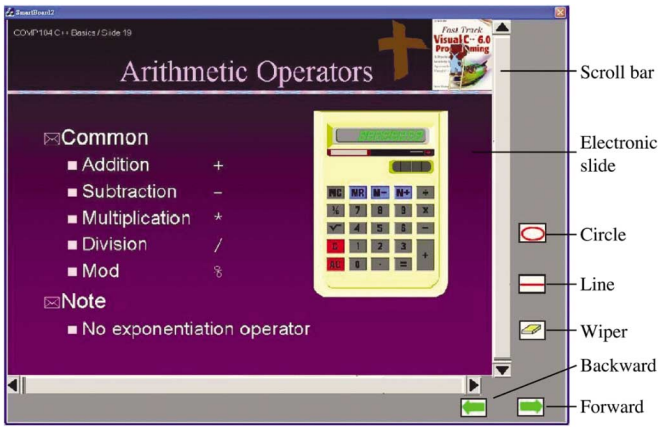

Fig. 6. Interface of the simulated smartboard.

of speech is not high and the output of ASR could be noisy, higher weight is expected for visual evidence.

IV. Simulated Smartboard

With the advance of HCI (Human-Computer Interaction) technologies, smartboard has been equipped and used in some conference rooms as a convenient tool for presentation. A presenter can write, make annotations and control the showing of the slides by using the touch screen and accessory tools such as markers and a wiper. In this section, we demonstrate that the smartboard can be software simulated with content-based solution as presented in Sections II and III.

A. System Setup and Interface

The system is set up in a classroom with a normal slide screen, an LCD projector, and a stationary video camera facing the screen. Fig. 6 shows the interface of the simulated smartboard. Prior to presentation, the electronic slides are uploaded to the computer. When the application is started, the slides are projected and displayed on the screen as shown in Fig. 6. Various tools, including the virtual buttons for *circling*, *lining*, *rubber*, *forward*, *backward*, and a scrolling bar are projected to the screen for the presenter to use. Notice that it is optional to use the buttons when making gestures. Our system can automatically predict and recognize the gesture of circling, lining and pointing. The gesture will be superimposed on top of the slide as soon as the gesture is detected. The buttons, if being pressed, can improve the response time of the system. To interact with the interface, a presenter can use either hand or laser pen, for instance, to press buttons and make gestures. The interactions are captured by the video camera and transferred to the computer through an IEEE 1394 cable for real-time processing. When an interaction is detected and recognized, the corresponding operation is performed and projected to the slide screen.

B. Functions

The system provides three levels of interactions: 1) automatic gesture recognition and camera motion simulation; 2) detection

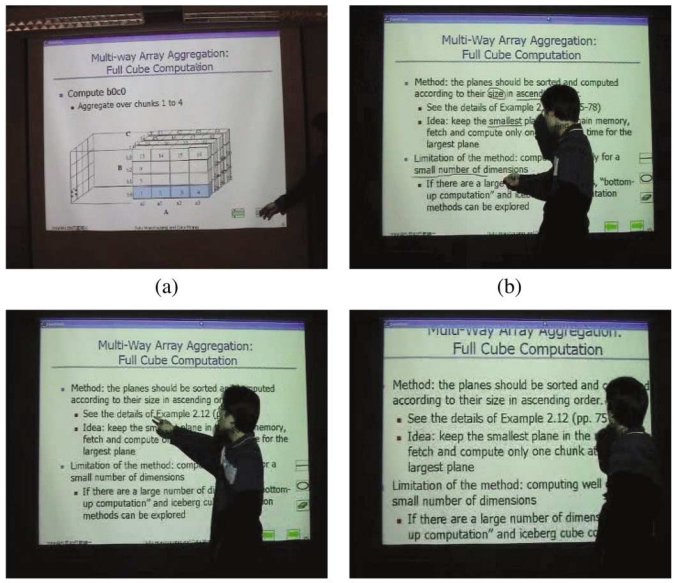

Fig. 7. Functions of the simulated smartboard. (a) Button gesture for slide control. (b) Annotation by using laser pen. (c) Dietic gesture for content highlighting: detected gesture (left); content highlighting by zoom-in (right).

of predefined gestures and actions; and 3) slide annotation with laser pen. Some example video clips can be found online [30]. In (1), a presenter can interact with the slide by gesturing near the ROIs. The system predicts and recognizes three types of gestures: *circling, lining* and *pointing* based on the algorithms proposed in Sections II and III. Naturally the response can be quicker if the presenter reads out some of the words in the ROI when gesturing. Once a gesture is detected, the ROI under interaction will be highlighted with simulated camera motion like zoom-in by employing the similar editing rules in our previous work [29]. For instance, in Fig. 7(c), a lining gesture pointing to a textline is detected. After a delay of several frames, the textline is shown in the center of the frame with a higher resolution. By highlighting the ROI, on one hand, the presenter can draw the audience's attentions to the content under discussion; on the other hand, the texts in higher resolution are easier to read as illustrated in Fig. 7(c).

In (2), a presenter can predefine the intended gestures by pressing buttons in the interface shown in Fig. 6. The buttons can be pressed with finger or laser pen. The functions of buttons are briefly summarized as follows.

- *Circle*: to draw a circle around an object (a word, equation or figure) on the slide.
- *Line*: to draw a line under an object (text or equation) on the slide.
- *Wiper*: to delete a marker (a circle, a line or an annotation) on the screen.
- *Forward/Backward*: to go to the next/previous slide.
- *Scroll bar*: to scroll up or down the slide.

To press a button, the hand needs to stay on top of it for at least 0.5 s so that the system can distinguish whether the hand is pressing the button or just sliding over it. Fig. 7(a) shows an example. When the button *forward* is pressed by the presenter, the next slide will be shown on the screen.

In (3), the presenter may make simple annotations by drawing or writing on the screen with a laser pen as illustrated in Fig. 7(b). The path of the laser pen is then tracked and displayed on the screen. Due to the precision and efficiency considerations, we do not support annotations or handwritings in small scales. Practically the laser pen should not be moved too fast. The presenter is expected to turn on and off the laser pen before and after making annotations. Therefore, the detected and tracked laser paths are assumed to be intentional when the pen is on.

C. Implementation Issues

To implement the three levels of functions described in the previous section, the system needs to be aware of three kinds of interactions: hand gesture highlighting, button-pressing, and laser pen annotation. The hand gestures are identified jointly by skin-color and frame difference cues [26], [29], while the trace of laser pen is mainly detected by tracking the color of the projected spot. The color cue from laser pen is not always reliable due to the reflection and varying background. Thus, if the hand holding a laser pen is not occluded, we also detect the hand around the laser spot to refine the detection of laser pen. If the laser pen is pointed in a distance from the screen and the hand cannot be detected along with the laser spot, laser light color becomes the only useful cue.

The detection of button pressing by hand or laser pen can be handled effectively. Since the buttons are fixed in position, only few regions from the incoming video streams need to be continuously monitored. To avoid false alarms such as the case when a hand slides over a button but without pressing, the trajectory of hand or laser pen needs to be aware of before any potential detection. In addition, we assume that the button pressing should last for at least 0.5 s to validate the action. Whenever pressed, a button will be displayed in the state of being pressed to indicate the activation of the command. After the gesture is finished and displayed on the slide, the button is then deactivated.

In response to the interactions, appropriate actions such as scrolling, zooming, writing, adding and removing annotations take place directly on the projected slides. To facilitate the easy manipulation of slides, the uploaded electronic slides are automatically converted to and saved as images. The semantic instances such as the textlines, figures and tables are extracted and indexed with their content and spatial positions. With this organization, we can efficiently operate the slides to project and display the editing decisions in real time.

The simulated smartboard can analyze 5–10 frames/s. From our experiment, the frame rate is enough for real-time tracking of hand gesture and laser pen movement. Currently we sample 8 frames/s for content analysis. The response time of the system is mainly dependent on the responsiveness of gesture detection and laser pen tracking, which can be considered as acceptable according to our experiments in Section V. Once a gesture for highlighting is recognized, a gradual camera zoom-in is simulated which takes about 2–4 s s to avoid abrupt visual changes. The speed of simulated camera motion can be adjusted according to the user's preference. The response to the button-pressing gesture and laser pen annotation is usually more efficient without recognition or simulating camera motion.

V. Experiments

We conduct two separate experiments for performance evaluation. The first experiment (Section V-A) is to verify the performance of gesture prediction with video-taped lectures. In this setup, the presenters are not aware of the smartboard or the algorithms for tracing their gestures. They present as usual in normal classroom environment. Our aim is to investigate the accuracy and responsiveness of our prediction algorithm. In the second experiment (Section V-B), the presenters can interact freely with the simulated smartboard, while they observe the editing effects projected on the slide screen during the presentation. We investigate, under the real scenario, the performance of our system with respect to different settings such as gesturing with hand instruction and laser pen.

A. Gesture Detection in Video-Taped Lectures

We conduct experiments on 5-h videos consisting of 15 presentations given by ten lecturers and tutors. The presenters include five males and five females. The presentations are given in one seminar room and three classrooms of different sizes, layouts, and lighting designs. A stationary camera is used to capture the projected slide and the presenter. There are five videos captured in the seminar room. Due to the reflection from the screen, performing text recognition and gesture detection in these videos are more difficult compared with those captured in the classrooms.

We compare two approaches: gesture prediction with visual-only cue, and with multimodal cues. In the dataset, there are 2060 gestures manually identified and marked. The average duration of a gesture is 4.62 s. For each gesture class, we use 200 samples to train the HMM. The training data is composed of gestures from another lecture video and some man-drawn figures such as ellipses and lines. Our previous experiments in [26] show that these training data works for recognizing real gestures from lecture videos.

We employ Microsoft Speech SDK 5.1 for speech recognition. The overall recognition rate is no more than 20% due to the environmental noise in audio track. Considering the fact that the performance of speech recognition is not high in general, visual cue is given a higher weight in multimodality fusion. As a result, the parameters in (16) are empirically set to $\lambda_v = 0.7$ and $\lambda_s = 0.3$. In addition, a parameter C_{gate} is also required to gate the confidence of detection in both (14) and (16). Practically, gestures with lower detection confidence will be regarded as noises. A larger value of C_{gate} usually means more competitive performance but with the expense of longer response time. Empirically, we set $C_{\text{gate}} = 0.45$.

Table I and Fig. 8 summarize and compare the performance by showing the recall and precision of gesture detection when varying the responsiveness constraints (delays). Obviously, allowing longer response time leads to better performance. With visual-only cue, 65% of all gestures can be correctly predicted $(\text{precision} = 0.75)$ with a delay of 2.5 s after the gestures start. In contrast, when multimodal cues are jointly considered, 85% of gestures can be recognized $(\text{precision} = 0.77)$ with only 1.8 s delay. From Fig. 8, to achieve the same accuracy, less response time is required when speech is combined with visual cue. The results indicate the significance of multimodal cues where the

TABLE I
RESULTS OF GESTURE DETECTION [N_g: TOTAL NUMBER OF GESTURES; N_d: NUMBER OF GESTURES DETECTED; N_f: NUMBER OF GESTURES FALSELY DETECTED; N_m: NUMBER OF GESTURES MISSED; N_c: NUMBER OF GESTURES CORRECTLY DETECTED. $\mathrm{Recall} = (N_c/N_g), \mathrm{Precision} = (N_c/N_d)$]

N_g			2060			
Gesture duration	Average		4.62 sec			
	Standard Deviation		1.83 sec			
Method	Delay (sec)	0.6	1.2	1.8	2.5	4.0
Visual	N_d	449	736	1255	1783	1966
	N_f	217	320	412	435	308
	N_m	1828	1644	1217	712	402
	N_c	232	416	843	1348	1658
	Recall *(%)*	11.3	20.2	40.9	**65.4**	**80.5**
	Precision *(%)*	51.7	56.5	67.2	**75.6**	**84.3**
Visual + Speech + Slides	N_d	980	1391	2275	2252	2170
	N_f	376	501	533	447	319
	N_m	1456	1170	318	255	209
	N_c	604	890	1742	1805	1851
	Recall *(%)*	29.3	43.2	**84.6**	**87.6**	**89.9**
	Precision *(%)*	61.6	64.0	**76.6**	**80.2**	**85.3**

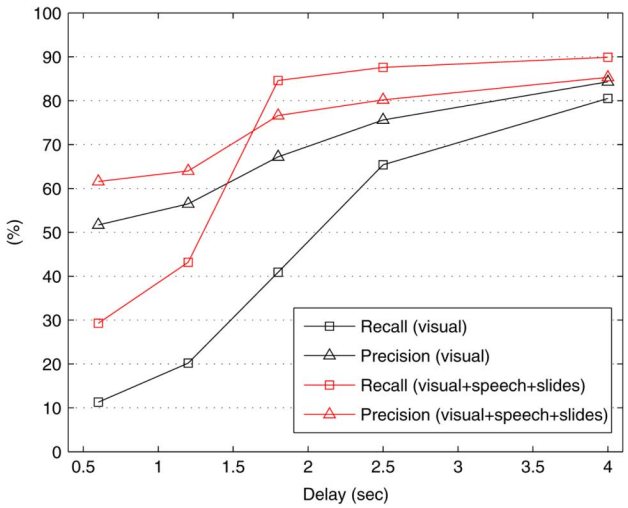

Fig. 8. Recall and precision of gesture detection. (The results are the same as in Table I, but presented to visualize the relationships between the performance of sing- and multimodality approaches).

majority of gestures can be correctly recognized half way before the complete gesture paths are observed. An interesting note found in the experiments is that most presenters, in addition to gesturing, tend to highlight keywords when explaining important concepts under lecturing. This, indeed, not only improves the recognition rate of speech, but also greatly signifies the benefit of gesture-speech correspondence, which boosts the recognition performance while shortening the response time.

The parameter C_{gate} can impact the performance if the value is not selected within a proper range. Fig. 9 shows the sensitivity of the parameter C_{gate} towards the performance and the response time. When $C_{\mathrm{gate}} > 0.4$, the algorithm performs satisfactorily in terms of both precision and recall values. A small gating confidence usually introduces many false alarms, while a very high value results small decline of the *recall*, which means some gestures are missed. When the gating confidence becomes larger, higher *precision* is achieved with an expense of longer response time. As seen in Fig. 9, setting the value of C_{gate} in the range of 0.4–0.6 can compromise the accuracy and speed.

Currently, the proposed gesture prediction algorithm can process approximately 13 frames/s. The frames are evenly sampled along the timeline, and the sampling rate is enough to depict the trajectories of gestures. Given the processing speed, our algorithm is efficient enough for real-time applications.

B. Performance of Simulated Smartboard

We evaluate the performance of simulated smartboard in providing the three levels of interactions: hand-based gesture highlighting (*dietic gesture*), gesturing with button pressing (*button gesture*), and laser pen annotation. A total of 3-h videos of presentation using the smartboard are captured for performance analysis in terms of gesture detection and laser pen tracking.

1) Gesture Detection: Table II shows the calculated *Recall* and *Precision* values of gesture detection. For *button gesture*, few gestures are missed due to the occlusion caused by the presenter and other objects. Some false alarms are inserted when the hand stays near or is projected onto the buttons. For *dietic gesture*, since a rather high $C_{\mathrm{gate}}(= 0.5)$ value is used to avoid false alarms in real scenarios, about 85% of all gestures can be correctly detected.

Fig. 10 summarizes the response time for gesture detection. The video rate is 24 frames/s, and the response time is expressed as the number of frames being delayed after an action is taken. For button gesture, we calculate the response time as the time interval from the hand interacting with the button to the activation of the command. To detect dietic gesture for content highlighting, we set $C_{\mathrm{gate}} = 0.5$. A gesture is verified only after the calculated confidence value is larger than this gating evidence. The responsiveness is evaluated by the time interval from the beginning of a gesture to the gesture being detected. With the fixed C_{gate}, the response time is out of direct control and can be up to the lifetime of the gesture. As shown in Fig. 10, the response time for more than 90% of all gestures is less than 3 s, which is considered as acceptable, and most gestures can be detected in the halfway of the gesture paths.

2) Laser Pen Tracking: In tracking laser spot, the projected annotation is usually deviated slightly from the real trajectory of the spot. To investigate this effect, we manually mark the laser spot trajectories and calculate the distance between the real and projected annotations. The deviation is generally within 0–5 pixels with the mean and standard deviation of 3.19 and 1.85, respectively. The deviation is considered acceptable and does not distort the intended annotations when assessing the effects of captured videos subjectively.

We sample eight frames every second for laser spot detection and tracking. The sample rate is enough to depict the annotation given that the laser pen is not moving too fast. To evaluate the responsiveness, for each point on the path of the annotation, we calculate the average time being delayed to display the laser point on the slide after it is projected by the pen. From experiments, the response time is usually less than one second with the mean and standard deviation of 17.43 and 4.44 frames, respectively.

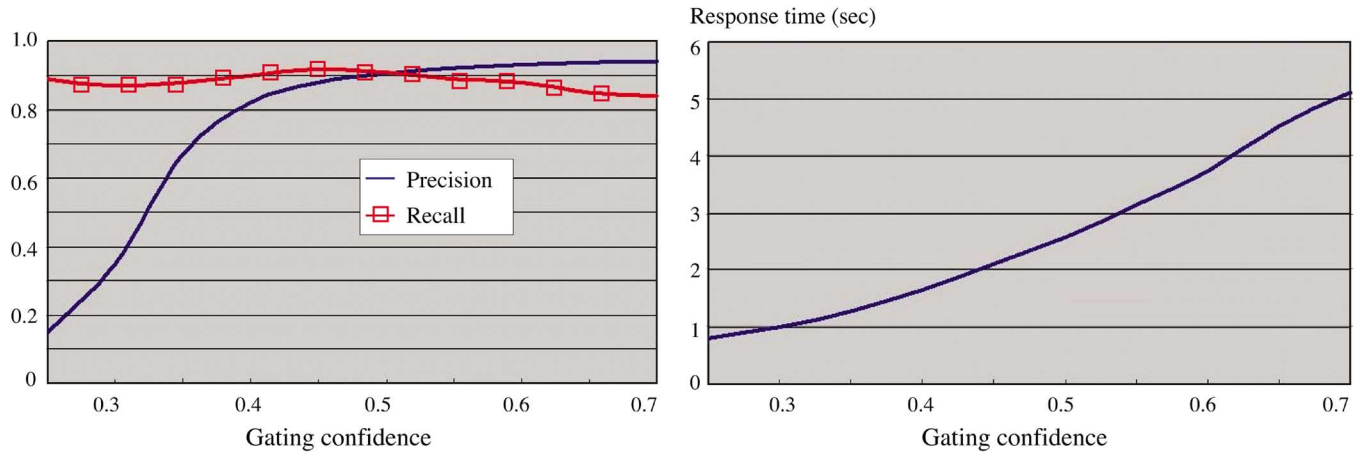

Fig. 9. Performance and response time of gesture detection with different gating confidence values.

TABLE II
ACCURACY OF GESTURE DETECTION [N_g: THE TOTAL NUMBER OF GESTURES; N_d: THE NUMBER OF GESTURES DETECTED; N_c: THE NUMBER OF GESTURES CORRECTLY DETECTED; N_m: THE NUMBER OF GESTURES MISSED; N_i: THE NUMBER OF GESTURES FALSELY INSERTED; $\text{Recall} = (N_c/N_g); \text{Precision} = (N_c/N_d)$]

	N_g	N_d	N_c	N_m	N_i	***Recall (%)***	***Precision (%)***
Button gesture	108	113	101	7	12	**93.5**	**89.4**
Hand gesture	104	97	88	12	9	**84.6**	**90.7**

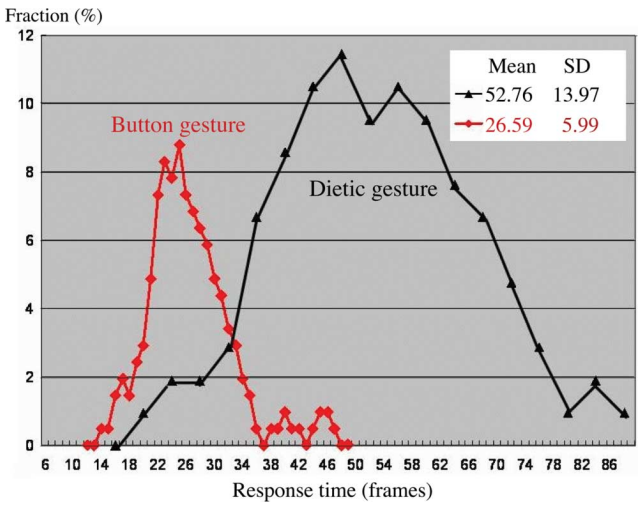

Fig. 10. Response time of gesture detection.

VI. CONCLUSION

We have presented a real-time gesture detection algorithm powered with prediction ability. Using the proposed modified HMM models, an intentional gesture can be efficiently predicted and verified. By further considering the gesture-speech multimodal correspondence, most gestures can be effectively recognized half way before the whole gesture path is completed. With this algorithm, we demonstrate the flexibility of building a simulated smartboard which supports on-the-fly presentation editing through three levels of interaction particularly with hand gesturing and laser pen annotation. With simulated smartboard as example, the experimental results indicate that our algorithm can cope with the requirement of real-time applications with high enough recall and precision performance.

For future work, the performance of speech recognition should be improved, and the fusion of different cues needs to be further studied. Other applications such as automatic camera management can be developed by employing the algorithm proposed in this work. Besides gesture, head posture proves to be another useful hint for lecture video content analysis [29]. In [27], we have proposed an efficient algorithm for head pose estimation which could be combined with gesture for real-time lecture video editing in the future.

REFERENCES

[1] G. D. Abowd, C. G. Atkeson, A. Feinstein, and C. Hmelo, "Teaching and learning as multimedia authoring: The classroom 2000 project," in *ACM Multimedia Conf.*, Los Angeles, CA, 2000, pp. 187–198.
[2] M. Chen, "Visualizing the pulse of a classroom," in *ACM Multimedia Conf.*, Berkeley, CA, 2003.
[3] M. Chen, "Achieving effective floor control with a low-bandwidth gesture-sensitive videoconferencing system," in *ACM Multimedia Conf.*, Juan les Pins, France, 2002, pp. 476–483.
[4] B. Erol, J. J. Hull, and D. S. Lee, "Linking multimedia presentations with their symbolic source documents: Algorithm and applications," in *ACM Multimedia*, Berkeley, CA, 2003, pp. 498–507.
[5] M. Gleicher and J. Masanz, "Towards virtual videography," in *ACM Multimedia Conf.*, Los Angeles, CA, 2000, pp. 375–378.
[6] M. Gleicher, R. M. Heck, and M. N. Wallick, "A framework for virtual videography," in *Proc. Int. Symp. on Smart Graphics*, Hawthorne, NY, 2002, pp. 9–16.
[7] L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *ACM Multimedia Conf.*, Orlando, FL, 1999, pp. 489–498.
[8] L. He and Z. Zhang, "Real-time whiteboard capture and processing using a video camera for remote collaboration," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 198–206, Jan. 2007.
[9] S. X. Ju, M. J. Black, S. Minneman, and D. Kimber, "Summarization of videotaped presentations: Automatic analysis of motion and gesture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 686–696, Sep. 1998.
[10] T. Liu, R. Hjelsvold, and J. R. Kender, "Analysis and enhancement of videos of electronic slide presentations," in *Proc. Int. Conf. Multimedia & Expo*, Lusanne, Switzerland, 2002, vol. 1, pp. 77–80.
[11] T. Liu and J. R. Kender, "Spatio-temporal semantic grouping of instructional video content," in *Proc. Int. Conf. Image and Video Retrieval*, Urbana-Champaign, IL, 2003, pp. 362–372.
[12] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz, "Automatic camera management for lecture room environment," in *Proc. Int. Conf. Human Factors in Computing Systems*, Seattle, WA, 2001, pp. 442–449.
[13] T. F. S. Mahmood and S. Srinivasan, "Detecting topical events in digital video," in *ACM Multimedia Conf.*, Los Angeles, CA, 2000, pp. 85–94.

[14] E. Machnicki and L. Rowe, "Virtual director: Automating a webcast," in *Proc. Multimedia Computing and Networking*, San Jose, CA, 2002.
[15] J. Martin and J. B. Durand, "Automatic gesture recognition using hidden Markov models," in *Proc. Int. Conf. Automatic Face and Gesture Recognition*, Grenoble, France, 2000, pp. 403–409.
[16] S. Mukhopadhyay and B. Smith, "Passive capture and structuring of lectures," in *ACM Multimedia Conf.*, Orlando, FL, 1999, pp. 477–487.
[17] C. W. Ngo, T. C. Pong, and T. S. Huang, "Detection of slide transition for topic indexing," in *Proc. Int. Conf. Multimedia & Expo*, Lausanne, Switzerland, 2002, vol. 2, pp. 26–29.
[18] M. Onishi and K. Fukunaga, "Shooting the lecture scene using computer-controlled cameras based on situation understanding and evaluation of video images," in *Int. Conf. on Pattern Recognition*, Cambridge, U.K., 2004, vol. 1, pp. 23–26.
[19] D. Q. Phung, S. Venkatesh, and C. Dorai, "Hierarchical topical segmentation in instructional films based on cinematic expressive functions," in *ACM Multimedia Conf.*, Berkeley, CA, 2003, pp. 287–290.
[20] L. A. Rowe and J. M. Gonzlez, BMRC Lecture Browser [Online]. Available: http://bmrc.berkeley.edu/frame/projects/lb/index.html
[21] Y. Rui, A. Gupta, and J. Grudin, "Videography for telepresentations," in *Proc. Int. Conf. Human Factors in Computing Systems*, Ft. Lauderdale, FL, 2003, pp. 457–464.
[22] T. K. Shih, T. H. Wang, C. Y. Chang, T. C. Kao, and D. Hamilton, "Ubiquitous e-learning with multimodal multimedia devices," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 487–499, Apr. 2007.
[23] T. Starner *et al.*, "Real-time American sign language recognition using desk and wearable computer-based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.
[24] M. N. Wallick and M. L. Gleicher, "Magic boards," in *SIGGRAPH*, Los Angeles, CA, 2005, no. 51.
[25] F. Wang, C. W. Ngo, and T. C. Pong, "Synchronization of lecture videos and electronic slides by video text analysis," in *ACM Multimedia Conf.*, Berkeley, CA, 2003, pp. 315–318.
[26] F. Wang, C. W. Ngo, and T. C. Pong, "Gesture tracking and recognition for lecture video editing," in *Proc. Int. Conf. Pattern Recognition*, 2004, pp. 934–937.
[27] F. Wang, C. W. Ngo, and T. C. Pong, "Exploiting self-adaptive posture-based focus estimation for lecture video editing," in *ACM Multimedia Conf.*, Hilton, Singapore, 2005, pp. 327–330.
[28] F. Wang, C. W. Ngo, and T. C. Pong, "Prediction-based gesture detection in lecture videos by combining visual, speech and electronic slides," in *Proc. Int. Conf. on Multimedia & Expo.*, Toronto, ON, Canada, 2006, pp. 653–656.
[29] F. Wang, C. W. Ngo, and T. C. Pong, "Lecture video enhancement and editing by integrating posture, gesture, and text," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 397–409, Feb. 2007.
[30] *Video Content Analysis for Multimedia Authoring of Presentations*, [Online]. Available: http://webproject.cse.ust.hk:8006/Smartboard.htm
[31] S. Young, HTK: Hidden Markov Model Toolkit Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington, DC, 1993.
[32] *IBM Auto Auditorium System*, [Online]. Available: www.autoauditorium.com

Feng Wang received the B.Sc. degree in computer science from Fudan University, Shanghai, China, in 2001 and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, in 2006.

He is currently a Research Fellow in the Department of Computer Science, City University of Hong Kong. His research interests include multimedia content analysis, pattern recognition, and IT in education.

Chong-Wah Ngo (M'02) received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University of Singapore in 1994 and 1996, respectively, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, in 2000.

Before joining City University of Hong Kong as Assistant Professor in the Computer Science Department in 2002, he was a Postdoctoral Scholar with the Beckman Institute, University of Illinois at Urbana-Champaign (UIUC). He was also a Visiting Researcher with Microsoft Research Asia in 2002. His research interests include video computing, multimedia information retrieval, data mining, and pattern recognition.

Ting-Chuen Pong received the Ph.D. degree in computer science from Virginia Polytechnic Institute and State University, Blacksburg, in 1984.

He joined the University of Minnesota-Minneapolis 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 and Technology (HKUST), Kowloon, 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. His research interests include computer vision, image processing, pattern recognition, multimedia computer, and IT in education.

Dr. Pong is a recipient of the HKUST Excellence in Teaching Innovation Award in 2001, the Annual Pattern Recognition Society Award in 1990, and the 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.