Gesture Tracking and Recognition for Lecture Video Editing

Feng Wang
Department of CS
HongKong University of
Science &Technology
wfeng@cs.ust.hk

Chong-Wah, Ngo
Department of CS
City University of
Hong Kong
cwngo@cs.cityu.edu.hk

Ting-Chuen, Pong
Department of CS
HongKong University of
Science & Technology
tcpong@cs.ust.hk

Abstract

This paper presents a gesture based driven approach for video editing. Given a lecture video, we adopt novel approaches to automatically detect and synchronize its content with electronic slides. The gestures in each synchronized topic (or shot) are then tracked and recognized continuously. By registering shots and slides and recovering their transformation, the regions where the gestures take place can be known. Based on the recognized gestures and their registered positions, the information in slides can be seamlessly extracted, not only to assist video editing, but also to enhance the quality of original lecture video.

1 Introduction

In the past few years, issues in multimedia authoring of live presentation have attracted numerous research attentions. Representative demonstrated systems include Classroom 2000 [1], BMRC lecture browser [13], and interactive virtual classroom [2]. Major research efforts include topical event detection [3, 7, 10], gesture analysis [5], multi-streams synchronization [7, 12], presentation summarization [4] and editing [9, 6, 14].

This paper focuses on the analysis of human gestures in lectures for video editing purpose. Online captured lecture videos, especially the lectures that are captured by a single static camera, are normally unedited and lack of "momentum". Automatic editing of lecture videos have been investigated in [9, 6, 14] by integrating shots from multiple cameras. One straightforward way is to create the cutting effects between different camera angles and these will normally give the viewers the sense of presence. In [9], an overview camera and a tracking camera are used for capturing. Editing is done by switching shots between cameras based on a set of predefined editing rules. Since there is no content analysis prior to editing, the edited videos may look "odd" sometimes.

Instead of rule-based driven editing, we propose a gesture driven editing which is more appropriate and natural for lecture videos. One example is when a presenter points or circles a particular region in screen, a close-up view of the region is shown in the edited video. The technical challenges involved in this problem are twofold: 1) how to effectively track and recognize gestures; 2) how to perform content editing particularly for videos that are suffered from low resolution and lighting fluctuation. For the first challenge, skin color detection, tracking and HMM (hidden Markov model) recognition are adopted. For the second challenge, the registration of video shots and electronic slides is carried out by matching textual content. Once the mapping from shots to slides is known, the information available in slides can be extracted directly to edit as well as to repair the original video for better visual effects.

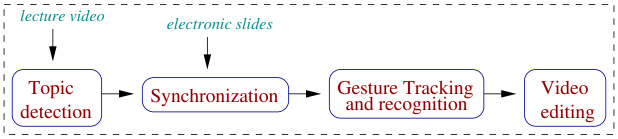


Figure 1. Content analysis of lecture video

Figure 1 illustrates the overview of our framework. Initially the possible topics (or shots) in a video are detected. The shots are synchronized with electronic slides based on video text analysis. For each synchronized shot, candidate gestures are tracked and recognized throughout the sequence. The recognized gestures are then utilized to guide video editing.

2. Synchronization of Shots and Slides

We use a static video camera to capture lecture videos. A presenter can move freely in front of regions that are viewable by the camera (see Figure 2 for examples). For topic detection, we adopt a rapid topic indexing approach in [10] based on the analysis of video texts in DC sequences without decompression. For synchronization, we adopt the recent works in [12] by automatic video text detection, super-resolution reconstruction, binarization and OCR recognition. The novelty of this approach lies on the utilization of high-resolution video text reconstruction prior to the OCR recognition. The matching of shots and slides are achieved by title and content similarity measures as in [12].

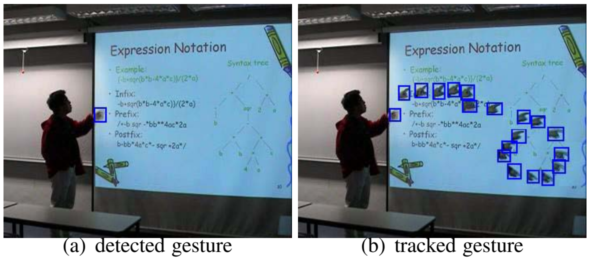
3. Gesture Tracking

We focus gesture tracking on the video region where a slide is projected. After synchronization, gestures are detected and located by integrating the inter-frame difference and skin-color detection. Once a gesture is detected, it is tracked until out of the defined slide region.

3.1. Skin Color Detection

Initially one frame is selected from every 10 frames, and the difference between the current and previous frames are computed. Candidate gestures are obtained if there are enough changes happened in the slide region. Skin color detection is then carried out to verify and locate the possible gestures.

Skin color has been shown to be a useful and robust cue for face and gesture detection. Different color spaces such as RGB, normalized RGB, HSV, YUV and YCrCb are used for skin color modeling. Various models including single Gaussian, multiple Gaussian and Bayes have been proposed to model the skin color distribution. These models, in general, require a large amount of training data for classifiers. A relatively straightforward approach to build a skin classifier is by explicitly defining (through a number of rules) the boundaries of skin cluster in a color space. Here we adopt the rules in [11] for our application. A pixel $\{R, G, B\}$ is classified as skin color if $\{R > 95, G > 40, B > 20, |R - G| > 15, R > G, R > B\}$ or $\{R > 220, G > 210, B > 170, |R - G| \leq 15, R > B, G > B\}$. One major advantage of this approach is that a rapid classifier can be easily constructed. The classified skin-color pixels are further clustered and those clusters with too small or too large sizes are excluded. Figure 2(a) shows one example where a hand gesture is successfully detected by our approach.



(a) detected gesture (b) tracked gesture

Figure 2. Gesture detection and tracking

3.2. Skin Color Tracking

Once a gesture is detected, it is tracked after every three frames. The tracking is based on skin-color detection and clustering. The detected skin-color pixels are grouped into several clusters. The tracked gesture is the cluster that satisfies the following criterion: 1) near the old location; 2) approximately equal dimensions; 3) similar color distribution. Figure 2(b) depicts the moving path of a tracked gesture.

4. Gesture Recognition

We adopt discrete HMM models for gesture recognition [8]. We defined three gestures in our application: *circling* (draw a circle around something), *lining* (draw a line along something) and *pointing* (point to somewhere for emphasis).

4.1. Gesture Segmentation

The moving path of a hand may consist of several gestures and some non-gesture movement. Before gesture recognition, a path need to be segmented into several parts, each of which only includes at most one possible gesture. Our segmentation is based on break point detection. A break point is identified based on the the following conditions:

1. The speed of movement changes rapidly before and after the point.

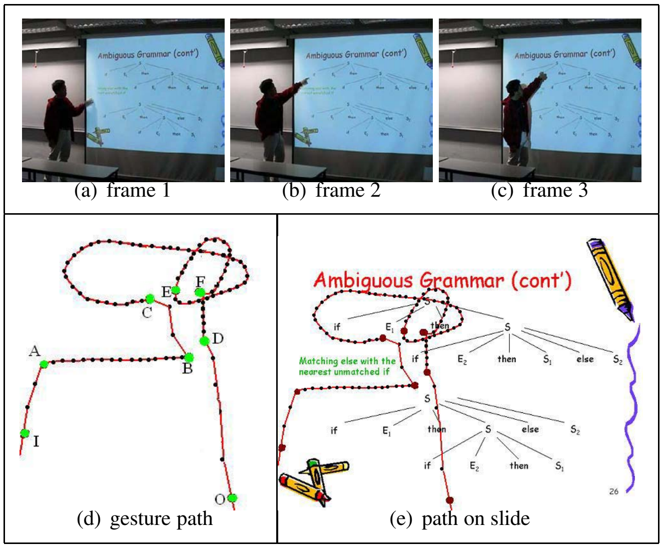
2. The point is very close to the starting point or a point before it on the path.

3. The direction of movement changed suddenly before and after the point.

4. The point that is the farthest away from starting point.

For each moving path, these conditions are checked one after another. Condition 1 has the highest priority. We assume the hand moves smoothly in a gesture. This condition will find most of the links between different gestures and the hand entering and leaving slide region. Then we look for the points that satisfy Condition 2 along each path segmented by 1. The purpose is to look for the possible closed circles. The path between the two end points should have large enough height and width to avoid pointing gestures, which also need to be satisfied when checking conditions 3 and 4.

The observed path is separated into several parts at these breakpoints. Each resulted part may be one of the following shapes or gestures: *circle, line, pointing*, or some non-gesture movement. The portions that last too short duration (less than 10 frames) are excluded as non-gesture. Also, the first and last parts are regarded as entering and leaving slide region. All the segmented path portions will fall into 4 classes: *circle, line, pointing* and *non-gesture*. Figure 3 illustrates the procedure. Fig 3(a)-(c) are 3 frames on the gesture path. Four points, A, B, C, D are found by Condition 1, and E, F by 2. IA, BC, ED and DO are excluded after segmentation.


Ambiguous Grammar (cont')
A
B
C
D
E
F
I
O
Matching else with the nearest unmatched if
if
E1
then
S
E2
S1
else
S2
26

(a) frame 1 (b) frame 2 (c) frame 3

(d) gesture path (e) path on slide

Figure 3. Gesture Segmentation

4.2. Feature Extraction

We select 20 points along each gesture path as feature points. Let v_m be the point that is farthest away from the starting point v_0, and $D = |v_0 v_m|$. For each point v_i, two kinds of features are calculated: the relative distance from v_0, denoted as $d_i = |v_0 v_i|/D$, and the angle ϕ_i between vector $v_{i-1}v_i$ and horizontal axis. The parameters d_i and ϕ_i are both quantized with 10 levels.

4.3. HMM Recognition

HMM has been successfully employed in speech recognition. Recently, it has also been applied in sign language interpretation and gesture recognition. The major advantage of HMM over other techniques such as neural-net (NN) and dynamic time warping (DTW) is its ability in selectively, knowledgeably and scalably tailoring the model to the task at hand. In our approach, Viterbi and Baum-Welch algorithms are employed to solve the evaluation and estimation problems for HMM. The number of states used is: 6 states for *lining* and 8 states for both *circling* and *pointing*.

5. Video Editing

Once a gesture is recognized, the *focus* of teaching is known. We can thus effectively simulate appropriate camera motion (*e.g.*, close-up of a particular slide region) and *cutting* effects to automatically edit the original videos. To enhance the quality of video, we also incorporate the information available in electronic slides during editing.

5.1 Registration of Video and Slide

During video capture, the slide regions are projected to another plane that usually does not parallel to the projected screen. To unwrap and register it from video to the real electronic slide, we solve for homography H given $p = (x_i, y_i)$ in a video and $\hat{p} = (\hat{x}_i, \hat{y}_i)$ in slides.

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

Hence,

$$\hat{x}_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02} \tag{2}$$

$$\hat{y}_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12} \tag{3}$$

For n points, by further manipulation, we have

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -\hat{x}_1 x_1 & -\hat{x}_1 y_1 & -\hat{x}_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -\hat{y}_1 x_1 & -\hat{y}_1 y_1 & -\hat{y}_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -\hat{x}_n x_n & -\hat{x}_n y_n & -\hat{x}_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -\hat{y}_n x_n & -\hat{y}_n y_n & -\hat{y}_n \end{bmatrix}$$

$$\times \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which is denoted as

$$Ah = 0 \tag{4}$$

This is a problem of linear least squares. We minimize

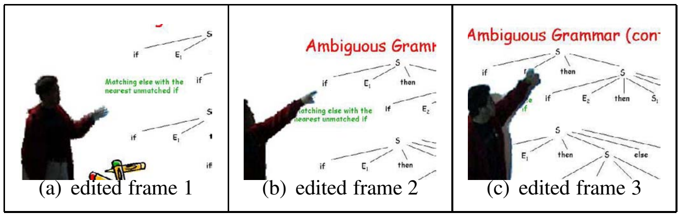
$$||Ah||^2 = (Ah)^T Ah = h^T A^T Ah \tag{5}$$

where h is the eigenvector of $A^T A$ with the smallest eigenvalue. Since h is only defined up to a scale, we just solve for the unit vector h. Eqn (5) requires four or more points.

The corresponding points between shots and slides are found by utilizing the positions of recognized texts. Since the positions of recognized video texts are known during synchronization [12], they are matched with the content and positions of texts extracted from electronic slides. Because recognition of titles is usually precise, we only utilize titles in slides for registration, which are enough for solving h. Figure 3(e) shows the moving path of a tracked gesture that is superimposed on the slide after registration.

5.2 Editing

After registration, the information in slides is extracted to aid the editing of low-resolution video shots for better visual effects. Figure 4 shows the edited results of the 3 frames in Figure 3. When a gesture is recognized, by registration, we know its position in the corresponding slide. The region where the lecturer focuses on can then be extracted to replace the slide region in shots.



(a) edited frame 1 (b) edited frame 2 (c) edited frame 3

Figure 4. Edited video frames of Fig 3(a)-(c)

6. Experiment

We conduct experiments on two videos. The duration of each video is about 40 minutes, and each one displays approximately 30 slides. Table 1 shows the results of synchronizing shots and slides. Encouraging results are obtained. Few mis-matching happens when two slides have the same title but not enough texts recognized in contents.

Table 1. Results of synchronization

Lecture Video	Total number of slides	# of correctly matched slides	Accuracy
1	34	31	91.2%
2	26	25	96.2%

There are totally $59,523$ frames in these two videos. All gestures are detected correctly. The tracking algorithm also works well in most time, except with gesture missing in 11 frames and error in 4 frames. The main reasons for missing and error are occlusion and under or over-illumination of the projected screen. For gesture recognition, we use 200 samples for each gesture to train HMM. Since there are not enough data for both training and testing, we use some man-drawn figures such as ellipses and lines, instead of gesture paths extracted from videos as training data. Our experiments indicate these training data works for recognizing real gestures from videos. Table 2 shows the results of gesture recognition.

7. Conclusion

We have presented a method to edit lecture videos based on gesture detection, tracking and recognition. Currently, only hand gestures are exploited to estimate the activities of the lecturer in a classroom. In future, advanced approaches such as head pose estimation and its joint relationship with hand gestures will be incorporated for video editing.

Acknowledgement

The work described in this paper was supported in part by the grants DAG01/02.EG16 ,VPAAO01/02.EG01, and a grant from City University of Hong Kong (Project No. 7001546).

Table 2. Results of gesture recognition

Video	Gesture	Circling	Lining	Pointing
1	Number of gestures	45	51	105
	Correctly recognized	45	49	98
	Recognition rate	*1.00*	*0.96*	*0.94*
	Overall Recognition Rate	*0.955*		
2	Number of gestures	35	67	92
	Correctly recognized	34	65	88
	Recognition rate	*0.97*	*0.97*	*0.96*
	Overall Recognition Rate	*0.964*		

References

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

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

[3] D. Phung, S. Venkatesh & C. Dorai, “High Level Segmentation of Instructional Videos Based on Content Density,” *ACM Multimedia*, 2002.

[4] L. He, E. Sanocki, A. Gupta & J. Grudin, “Auto-Summarization of Audio-Video Presentations,” *ACM Multimedia*, pp. 489-498, 1999.

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

[6] Q. Liu, Y. Rui, A. Gupta & J. J. Cadiz, “Automatic Camera Management for Lecture Room Environment,” *Int. Conf. on Human Fectirs in Computing Systems*, 2001.

[7] T. F. S. -Mahmood, “Indexing for topics in videos using foils,” *Int. Conf. CVPR*, pp. 312-319, 2000.

[8] J. Martin & J. B. Durand, “Automatic Gestures Recognition Using Hidden Markov Models,” *Int. Conf. Automatic Face and Gesture Recognition*, 2000.

[9] S. Mukhopadhyay & B. Smith, “Passive Capture and Structuring of Lectures,” *ACM Multimedia*, 1999.

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

[11] P. Peer, J. Kovac & F. SOLINA, “Human skin colour clustering for face detection,” *Int. Conf. on Computer as a Tool*, 2003.

[12] F. Wang, C. W. Ngo & T. C. Pong, “Synchronization of Lecture Videos and Electronic Slides by Video Text Analysis,” *ACM Multimedia Conference*, 2003.

[13] L. A. Rowe & J. M. Gonzlez, “BMRC Lecture Browser,” http://bmrc.berkeley.edu/frame/projects/lb/index.html.

[14] *IBM AutoAuditorium System*, www.autoauditorium.com.