Automated Lecture Recording System

Han-Ping Chou
Dept. of Information Management
Chung-Hua University
Hsin-Chu, Taiwan

Jung-Ming Wang, Chiou-Shann Fuh
Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

Shih-Chi Lin, Sei-Wang Chen
Dept. of Computer Science and Information Engineering
National Taiwan Normal University
Taipei, Taiwan

***Abstract*—Lecture recording plays an important role in online learning and distance education. Most of they are recorded by a cameraman or a static camera. In this paper, we propose an automatic lecture recording system. A Pan-Tilt-Zoom (PTZ) camera is shooting as it operated by a cameraman. Three parts are developed in this system. The first one is preprocessing for detecting the position of the lecturer and the screen. The second part is designed to track their motion to define the lecture information. According to the tracking result, we can control the PTZ camera in the third part based on the camera action table designed beforehand.**

***Keywords*—camera man, OpenCV, mean shift, action table.**

I. INTRODUCTION

Some academic or commercial institutions will conduct the lecture with the aim of carrying on the education, the propaganda or the communication. To assist these audiences in receiving information without the limitation of time and places, we will record the lectures and broadcasts in the network. Such recording will have the following works: First, we need to explore the lecture room to decide the camera setting. Second, a cameraman operates a camera to make a recording. Third, a post-production is designed.

As the discussions in [1] and [2], however, recording a lecture is really expensive. It would include of the fixed and labor cost. In fixed costs, we may require computer server, microphone, camera, and etc. Fortunately, fixed cost is only paid in construction. Labor costs, such as the payments for equipment, operating cameras, and post-production, may need to pay for each lecture recording. Since labor cost is required every time, many researches focus on automatic recording systems in order to reduce the cost. Here are some examples: Berkeley Internet Broadcasting System (BIBS) [2], AutoAuditorium system [3], iCam system[1] [4] [5] [6], University of Toronto ePresence system[7] and others [8] [9].

A lecture would have the following information to be recorded: the slides on the screen, the lecturer's talk and gesture, and the audience's response. Some researches, such as [1] and [16], show the slides and lecturer in different windows. This will have a clear slide, but they need to have the slide file from the lecturer or have another camera to shoot the slide. In [5], they focus on recording the voice more clearly. Several microphones are set in the room, and sound source is located to select the microphones to receive sound.

Image information, such as the lecturer's gesture and the audience's response, is more interested than voice and the slide. In [17], they use a wide-angle and high-resolution camera to cover the whole scene, and cut the sub-window of the scene for output. Similarly, a PTZ camera also can be applied to extend the viewing field [18]. Those methods help to record the lecture livelier than those video captured using only the static camera.

In this paper, we propose an automatic lecture recording system. In this system, we use a PTZ (pan, tilt, zoom) camera shooting in a lecture. Only image information is considered right here. The lecturer and the screen are the main information captured in our system. Based on the camera action table suggested by [5] and [1], we control the PTZ camera to make the recording video as that shot by a real cameramen. In the next section, we will talk about our system configuration. Processing and lecture information acquistion are discussed in Section 3 and Section 4. Finally we give some experiments and conclusions in Section 5 and Section 6, respectively.

II. SYSTEM CONFIGURATION

The professional cameramen suggest that the camera should be mounted on the central back hall and at the eye level of the audience. Under this setup, the lecturer and screen can be captured in our recording. Fig. 1 shows the whole system environment on the top-view.

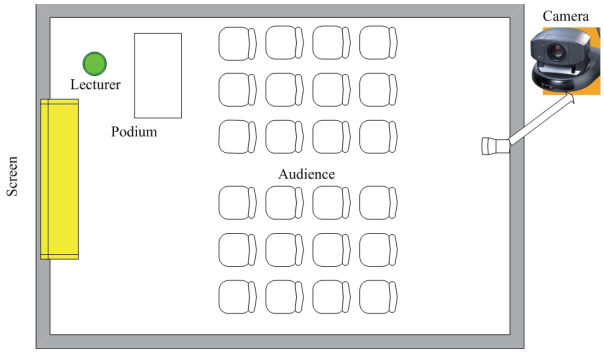


Figure1. The top-view of the system environment.

The processing in our system can be divided into three parts: preprocessing, acquisition of lecture information, and camera action. The system flow chart are shown in Fig. 2. In preprocessing, the screen and the lecturer are detected to get their positions before our system running. Acquisition of

lecture information is designed to track the screen and lecturer constantly for getting their following positions. Camera action is to control the camera viewing according to the camera action tale designed beforehand. The details will be discussed in the following sections.

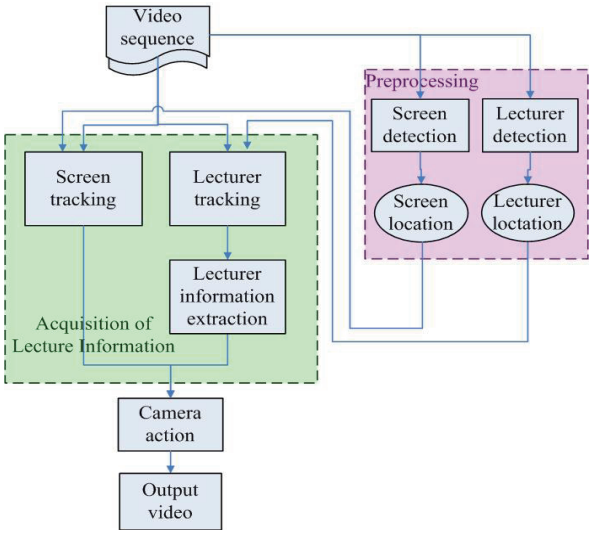


Figure 2. System flow chart

III. PREPROCESSING

Our processing includes lecturer detection and screen detection, which are applied to locate the position of the lecturer and the screen. The detection result helps us to initialize the state of the camera.

A. Lecturer Detection

Assume that there is only one person standing in the front of the lecture room, and that person is our lecturer. We also assume that the lecturer would face to the audience, then his or her face will be shown in the video because our camera is mounted on the back wall. Under the above assumptions, we can apply face detection method to locate the lecturer's face.

We use face detection method based on Adaboost algorithm proposed by Viola and Jones [10]. This algorithm trains a strong classifier to detect a face in the whole image. OpenCV 's face detection is such classifier, and the program can be downloaded from the OpenCV web site http://www.opencv.org. Fig.3 shows the result of the face detection in which the human face is marked using a square.

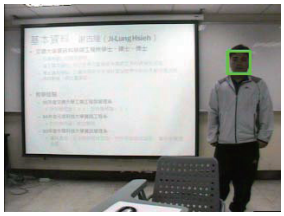


Figure3. Human face is located by OpenCV's face detection method.

B. Screen Detection

In the lecture scenes, the brightness of the screen is higher than the others. We can use this features to detect the screen location. At first, we calculate the gray histogram of the input image, and then use Otsu's methods to calculate the threshold to filter the brighter blobs in the image. Each blob is calculated out the aspect ratio and size to decide the screen region. Let a be the blob size and A be size of the minimum square cover this blob. Since a screen must be 4:3 aspect ratio in square shape, we will filter those blobs with large aspect ratio, small area ration a/A, and small size. Fig. 4 shows the example of the detection result.

After getting the screen region, four corners are extracted to represent the screen's coordinates of the location. Coordinates of the corners marked as (x_t^i, y_t^i) at time t, i=1, 2, 3, 4 as shown in Fig.5). Besides, we also calculate the viewing angles $(\theta_t^{x^i}, \theta_t^{y^i})$ between the viewing directions to each corner and the image center at time t, where the angle θ is calculated by the formula $\theta = \tan^{-1}\frac{d}{f}$, and d is the distance between the corner to the image center, and f is the focal length. Fig. 6 shows the concept of the calculation.

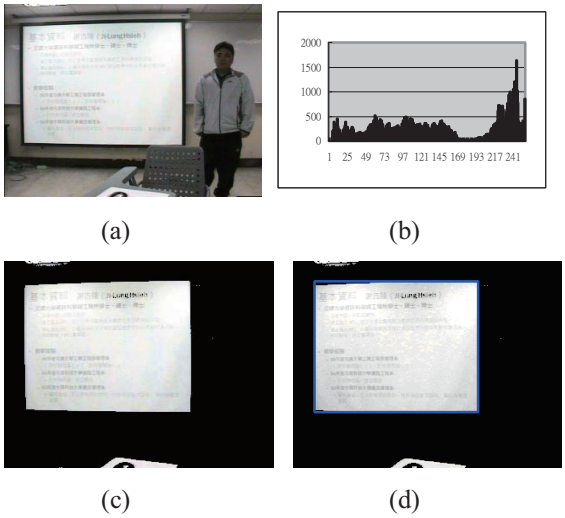


Figure 4. Screen detection: (a) Input image (b) Gray histogram of the input image (c) Brighter blobs extraction (d) Screen region detection result.

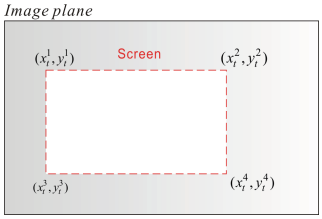


Figure 5. The coordinates of the corners in the image plane.

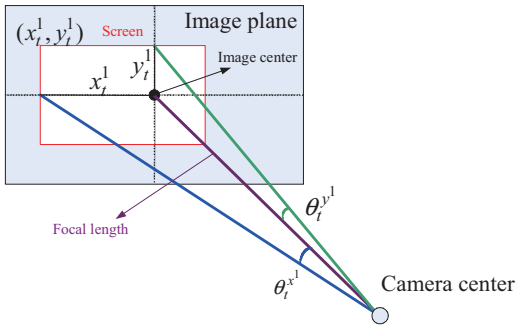


Figure 6. Viewing angle $(\theta_t^{x^i}, \theta_t^{y^i})$ between the coordinates (x_t^i, y_t^i) and the camera center.

IV. ACQUIRING LECTURE INFORMATION

After have the locations of the lecturer and the screen, we trace their following locations using tracking method. The tracking result helps us to define their relationship, and then we can operate the camera according to the action table (Table 1).

A. Screen Tracking

After detecting the screen, we can track the screen's following coordinates based on the motion of the PTZ camera. Our method not only can get the new location of the screen in the image, but also can get screen position relative to the location of speakers even if the camera did not capture the screen.

Fig.7 shows the relationship between the screen coordinates in the world and the image plane. After the camera motion, we use the motion parameter values to re-calculate the location *P'* projected from *P*. The re-calculation of the screen coordinates is mainly divided into two parts: The first one is to calculate the changing of the camera optical axis, which is caused by the camera rotation (pan angle θ^P and tilt angle θ^T), and the second one is to calculate the new focal length changed by zoom operation.

Suppose the point P project to the image plane at coordinates $P_t(x_t^i, y_t^i)$, the viewing angles between point P and the camera center at time *t* with the focal length f_t is $(\theta_t^{x^i}, \theta_t^{y^i})$. After we give a pan action with the angle θ^P (Fig.8) at time *t*+1, the angles between point P and the camera center will change to $(\theta_{t+1}^{x^i}, \theta_{t+1}^{y^i})$, where $\theta_{t+1}^{x^i} = \theta_t^{x^i} + \theta^P$ and $\theta_{t+1}^{y^i} = \theta_t^{y^i}$. The focal length will not change, $f_{t+1} = f_t$. The new projection result $P_{t+1}(x_{t+1}^i, y_{t+1}^i)$ can be calculated using the following equations:

$$x_{t+1}^i = f_{t+1} * \tan(\theta^P + \theta_t^{x^i})$$
$$y_{t+1}^i = f_{t+1} * \sec(\theta^P + \theta_t^{x^i}) * \tan(\theta_t^{y^i}) \quad (1)$$

Similarly, if we have the change in tilt angle, we will have the following equations:

$$x_{t+1}^i = f_{t+1} * \sec(\theta^T + \theta_t^{y^i}) * \tan(\theta_t^{x^i})$$
$$y_{t+1}^i = f_{t+1} * \tan(\theta^T + \theta_t^{y^i}) \quad (2)$$

After combining (1) with (2), we will have

$$x_{t+1}^i = f_{t+1} * \sec(\theta^T + \theta_t^{y^i}) * \tan(\theta^P + \theta_t^{x^i})$$
$$y_{t+1}^i = f_{t+1} * \sec(\theta^P + \theta_t^{x^i}) * \tan(\theta^T + \theta_t^{y^i}) \quad (3)$$

After the zoom action from f_t to f_{t+1}, we can calculate the screen coordinates according to the focal length and the geometric relationship between the triangles. Fig.9 shows an example to predict the coordinates on the *x*-axis. The point P projects to the image plane at coordinates x_t^i at the time *t*, and the focal length is f_t. The new coordinate $P_{t+1}(x_{t+1}^i, y_{t+1}^i)$ will be changed to

$$x_{t+1}^i = (x_t^i) * \frac{f_{t+1}}{f_t}$$
$$y_{t+1}^i = (y_t^i) * \frac{f_{t+1}}{f_t} \quad (4)$$

Equations (3) and (4) help us to track the location of the screen after the camera actions. Fig. 10 shows the result of the screen tracking.

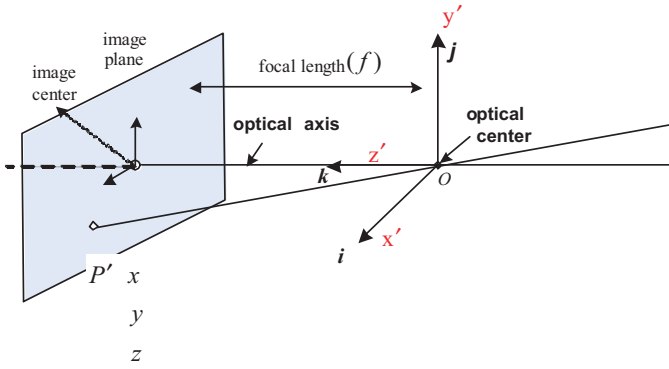


Figure 7. Diagram of world coordinate and image plane.

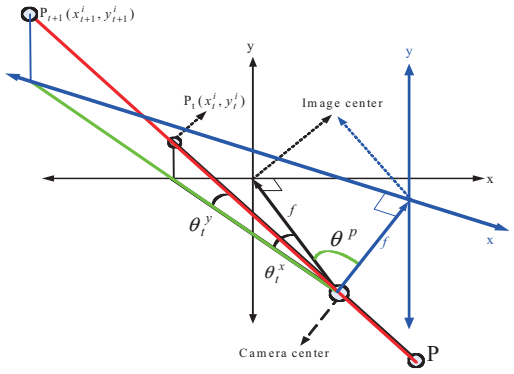


Figure 8. Camera coordinate after panning with the angle θ^P.

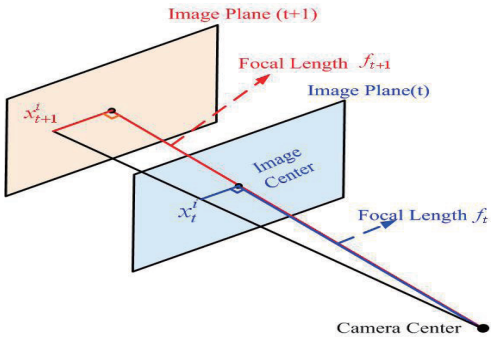
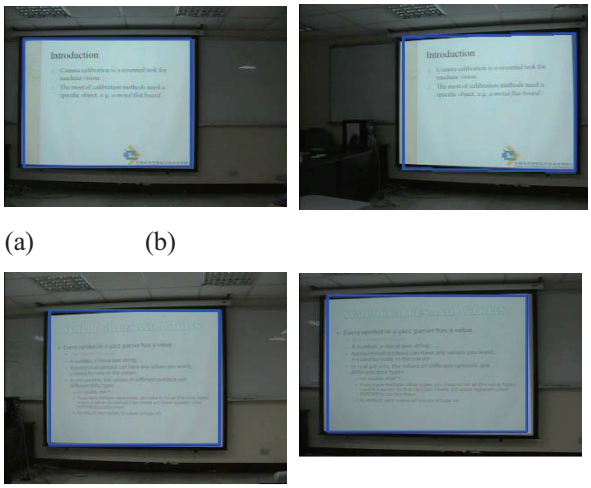


Figure 9. The change of the coordinates after zooming.



(a) (b)

(c) (d)

Figure 10. Screen tracking: (a) Before rotating (b) After rotating (c) Before zooming (d) After zooming.

B. *Lecturer Tracking*

We use mean shift method [13] to track the lecture movement. The target module is provided by the lecturer detection. In this paper, we use two features, color and edge orientation histogram (EOH) [11], to represent the module and to match the candidate module in current frame.

- *1) Color Representation*

Let $\{\mathbf{x}_i^*\}_{i=1...n}$ be the pixel location of the target module, centered at 0, the function $b: R^2 \to \{1.....m\}$ that associates to the pixel at location $\mathbf{x}_i^*$ and corresponds to the color of that pixel. The target module of the object using color histogram is formulated as

$$q^c = \{q_u^c\}_{u=1,.....m} \qquad (5)$$

where

$$q_u^c = \frac{1}{C} \sum_{i=1}^{n} k(\|\mathbf{x}_i^*\|^2)\delta[b(\mathbf{x}_i^*) - u],$$

C is a normalization constant, k is an Epanechnikov kernel function, and δ is the Kronecker delta function. The candidate module for the color histogram at the current frame is defined at location $\mathbf{y}$ as

$$p^c(\mathbf{y}) = \{p_u^c(\mathbf{y})\}_{u=1,.....m} \qquad (6)$$

where

$$p_u^c(\mathbf{y}) = \frac{1}{C_h} \sum_{i=1}^{n} k(\left\|\frac{\mathbf{x}_i - \mathbf{y}}{h}\right\|^2)\delta[b(\mathbf{x}_i) - u].$$

- 2) EOH *Representation*

Similarly, let $\{\mathbf{x}_i^*\}_{i=1...n}$ be the pixel location of the target module, centered at 0, the function $o: R^2 \to \{1.....v\}$ that associates to the pixel at location $\mathbf{x}_i^*$ and corresponds to the edge orientation of that pixel. The target module of the object using EOH is formulated as

$$p^e(\mathbf{y}) = \{p_r^e(\mathbf{y})\}_{r=1,.....v} \qquad (7)$$

where

$$p_r^e(\mathbf{y}) = C_h \sum_{i=1}^{n} k(\left\|\frac{\mathbf{x}_i - \mathbf{y}}{h}\right\|^2)\delta[o(\mathbf{x}_i) - r].$$

The candidate module for the EOH at the current frame is defined at location $\mathbf{y}$ as

$$p^c(\mathbf{y}) = \{p_u^c(\mathbf{y})\}_{u=1,.....m} \qquad (8)$$

- *4) Object Tracking under the Mean Shift Algorithm*

We estimate the discrete density q of the target module, while p is estimated at a given location $\mathbf{y}$ of the candidate module. The estimate of the Bhattacharry coefficient is given by:

$$\rho(\mathbf{y}) \equiv \rho[p(\mathbf{y}), q] = \sqrt{p(\mathbf{y})q} \qquad (9)$$

The coefficient is between 0 and 1. The more similar q and p are, the coefficient is closer to 1.

Our tracking problem is to find the best match of location by maximizing the Bhattacharryya Coefficient in current frames. The search for the new target location in the current frame starts at the estimated location $\mathbf{y}_0$ of the target in the previous frame by using formula:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{n} \mathbf{x}_i k_u \left\|\frac{\mathbf{x}_i - y_0}{h}\right\|^2 w_i}{\sum_{i=1}^{n} k_u \left\|\frac{\mathbf{x}_i - y_0}{h}\right\|^2 w_i} \qquad (10)$$

where

$$w_i = \sum_{u=1}^{m} \delta[b(\mathbf{x}_i) - u]\sqrt{\frac{q_u^c}{p_u^c(\mathbf{y}_0)}} + \sum_{r=1}^{v} \delta[o(\mathbf{x}_i) - w]\sqrt{\frac{q_r^e}{p_r^e(\mathbf{y}_0)}}.$$

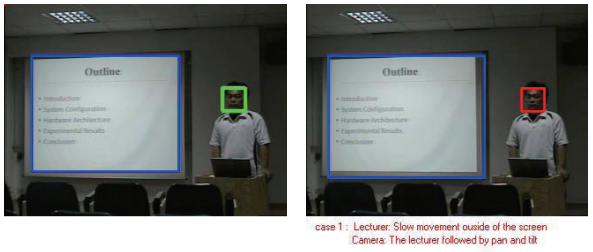
Equation (10) is derived from the Bhattacharryya coefficient, and gradually move to the new location which maximizing the $\rho(\mathbf{y})$ by using mean shift algorithm [12] [13] [14] [15].

- *5) Face direction*

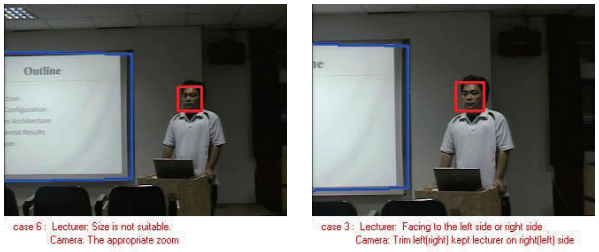
After lecturer tracking, a lecturer's facial module can be identified in image sequence. To complete a camera's movement, it is important not only to get a lecturer's positions, but also to analyze other states. The face direction of a lecturer is the state acquired here. The obtained facial module is divided into the right and the left parts that are used to identify the face direction based on the number of pixel of skin color. As in fig.11 (a), the pixel number of skin color in the left part is

larger than the one in the right part, so it can be determined that the lecturer is facing to the left.

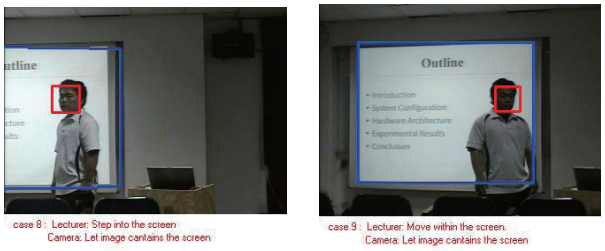
Figure 11. Face direction (a) Left (b) Front (c) Right



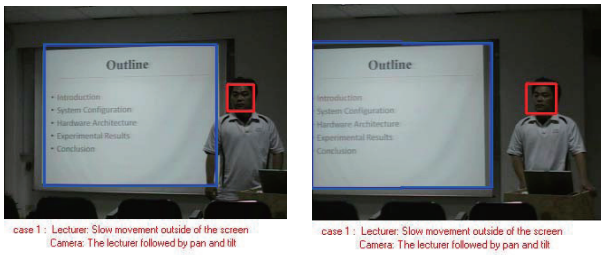
(a) Time 129 (b) Time 133



(c) Time170 (d) Time259



(e) Time 313 (f) Time 424



(g) Time 431 (h) Time 454

Figure 12. Automated lecture recoding

V. EXPERIMENTAL RESULTS

Location of lecturer, face direction and location of screen can be obtained by Acquisition of Lecture Information procedure and then this system automatically controls the PTZ camera to do appropriate actions by using the information to judge the lecture events according to the camera action table (Table 1) that we established beforehand.

This system uses the EVI-D30 PTZ camera to shoot the lecture scene, and uses video capture card to get the video sequences (320*240), the hardware is Core2 1.86Ghz, 1GB ram, the operation system is in Microsoft Windows XP, the IDE is Borland C++ 6.0.

Fig.12 shows the result of our automated camera shooting the lecturer. First of all (Fig.12a), the square with light color marks the lecturer's face by lecturer detection, if the location of green square continues appearing in the nearby area itself for a while, our system changes to lecturer tracking (Fig.12b), and starts the automatic operation. The square with dark color marks the screen location and red square marks the location of lecturer's face, and there is description of lecture's events and camera action below the image. While the lecturer is facing the audience, the camera follows the lecturer's movement and close-up shot of lecturer face, showed as Fig.12 (b) (c) (d). Fig.12 (e) (f) shows the lecturer step out of the screen, and camera follows the lecturer movement.

VI. CONCLUSIONS AND FUTURE WORK

This automated lecture recording system is designed for the purpose to record precious lectures and speeches, and it is refined to reduce the cost of recording. In order to reduce the cost, a camera action table, under which the system controls a camera automatically, is set up.

In the future, more and more detailed analyses of lecturer's gestures and movements will be added into the system to operate a camera precisely. Furthermore, the visual aids and voices will be captured to provide the remote audience a complete content.

REFERENCES

[1] Q. Liu, Y. Rui, A. Gupta, J. J. Cadiz ,"Automating camera management for lecture room environments," *Proc. of the SIGCHI conf. on Human Factors in Computing Systems,* pp. 442-449, Seattle, 2001.

[2] L. A. Rowe, D. Harley, P. Pletcher, and S. Lawrence, "BIBS:Alecture webcasting system," *Center for Studies in Higher Education, UC Berkeley*, 2001.

[3] M. Bianchi, "Automatic video production of lectures using an intelligent and aware environment," *Proc. of the 3rd Int'l Conf. on Mobile and Ubiquitous Multimedia*, pp. 117-123, College Park, Maryland, 2004

[4] M. N. Wallick, Y. Rui; L. He, "A portable solution for automatic lecture room camera management", *IEEE Int'l Conf. on Multimedia and Eposition*, vol 2, pp. 987 – 990, Taipei, 2004

[5] C. Zhang, Y. Rui, J. Crawford, and L. W. He, " An automated end-to-end lecture capture and broadcasting system," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol 4, no. 1, pp. 1-23, Brazil, 2008.

[6] Y. Rui, L. He, A. Gupta, and Q. Liu, , "Building an intelligent camera management system," *Proc. of the ACM Multimedia*, vol.9, pp. 2-11, Ottawa, 2001

[7] R. Baecker, ”A principled design for scalable internet visual communications with rich media, interactivity, and structured archives”, *Proc. of the 2003 Conf. of the Centre for Advanced Studies on Collaborative research*, pp. 16-19, Toronto, 2003

[8] M. Onishi, and K. Fukunaga, ”Shooting the lecture scene using computer-controlled cameras based on situation understanding and evaluation of video images,” *Proc. of the 17th Int'l Conf. on Pattern Recognition*, vo1. 1, pp. 781 – 784, Cambridge, 2004.

[9] F. Wang, C. W. Ngo, and T. C. Pong, “Synchronization of lecture videos and electronic slides by video text analysis,” *Proc. of the ACM Int'l Multimedia*, pp. 315-318, Berkeley, 2003

[10] P. Viola and M. J. Jones, "Robust real-time face detection," *Int'l Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.

[11] W. Liu, and Y. J. Zhang, ”Real time object tracking using fused color and edge cues,” *Proc. of Int'l Symposium on Signal Processing and Its Application*, pp.1-4 , Sharjah, United Arab Emirate, 2007

[12] K. Fukunaga, and L. Hostetler,"The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975

[13] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, 1995.

[14] D. Comaniciu, V. Ramesh, and P. Meer, P, ”Real-time tracking of non-rigid objects using mean shift,” *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 142-149, Kauai, Hawaii, 2000

[15] D. Comaniciu, V. Ramesh, and P. Meer, ”Kernel-based object tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[16] S. Mukhopadhyay, and B. Smith,”Passive capture and structuring of lectures,” *ACM international conference on Multimedia*, pp. 477-487, Orlando, 1999

[17] T. Yokoi, and H. Fujiyoshi,”Virtual camerawork for generating lecture video from high resolution images,” *IEEE Int'l Conf. on Multimedia and Expo*, pp. 751-754, Amsterdam, 2005

[18] C. Zhang, Y. Rui, L. He, and M. Wallick, “Hybrid speaker tracking in an automated lecture room,” IEEE International Conference on *Multimedia and Expo*, pp. 81-84, Amsterdam, 2005.

TABLE I. CAMERA ACTION TABLE

Cases	***Status of Lecturer***	***Face***	***PTZ Camera Actions***
1	Slow movement outside of the screen	*	The lecturer followed by pan and tilt
2	Fast movement outside of the screen	*	Zoom out
3	No movement outside of the screen	Left or Right	Trim left (right) for lecture who is kept on right(left) side of the image
4	Toward to the screen.	Face to the screen	The zoomed-out image contains part of the screen
5	Face location is not suitable.	*	Keep the lecturer centered by pan or tilt
6	Size is not suitable.	*	The appropriate zoom
7	A suitable location.	*	None
8	Step into the screen.	*	Image contains the screen, that acted by zoom, pan and tilt respectively
9	Move within the screen.	*	None
10	Move within the screen and approach its boundary.	Face to the opposite direction of the screen	Image contains part of the sconce outside of the screen.
11	Disappear.	*	Zoom out