Hybrid Speaker Tracking in An Automated Lecture Room

Cha Zhang, Yong Rui, Li-wei He

Communication and Collatoration Systems Group
Microsoft Research
One Microsoft Way, Redmond, WA 98052
{chazhang, yongrui, lhe}@microsoft.com

Michael Wallick [*]

Computer Sciences Department
University of Wisconsin - Madison
1210 West Dayton Street, Madison, WI 53706
michaelw@cs.wisc.edu

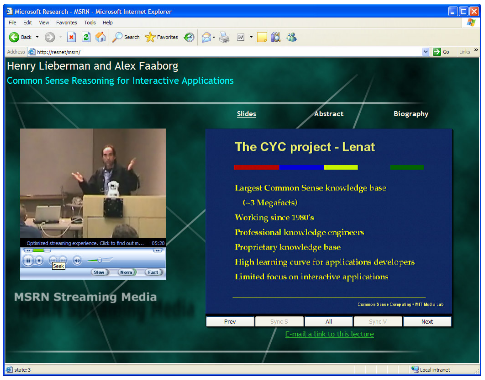
Abstract

We present a hybrid speaker tracking scheme based on a single pan/tilt/zoom (PTZ) camera in an automated lecture capturing system. Given that the camera's video resolution is higher than the required output resolution, we frame the output video as a subregion of the camera's input video. This allows us to track the speaker both digitally and mechanically. Digital tracking has the advantage of being smooth, and mechanical tracking can cover a wide area. The hybrid tracking achieves the benefits of both worlds. In addition to hybrid tracking, we present an intelligent pan/zoom selection scheme to improve the aestheticity of the lecture scene.

1 Introduction

Online broadcasting of lectures and presentations, live or on-demand, is increasingly popular in universities and corporations as a way of overcoming temporal and spatial constraints on live attendance. For instance, at Stanford University, lectures from over 50 courses are made available online every quarter [1]. University of California at Berkeley has developed online learning programs [2] with "Internet classrooms" for a variety of courses. Columbia University provides various degrees and certificate programs through its e-learning systems [3]. In our organization, an automated lecture capturing system (iCam) [4] has been developed and used on a daily basis for 3 years, during which more than 500 lectures have been captured and broadcast live online.

Figure 1 gives a snapshot of our web interface for watching seminars online. On the left hand side, there is a video stream generated by the iCam system. It is an edited video switching among the speaker view, the audience view, the screen view and the overview of the lecture room. The video resolution is 320×240. The slides of the talk are displayed on the right, which are captured as static images by a dedicated screen capture device before they are sent to the projector. The slides are updated automatically whenever a frame difference is detected.


Microsoft Research - MSRN - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://resnet/msrn/
Go
Links
Slides
Abstract
Biography
Optimized streaming experience. Click to find out m...
05:20
Seek
Slow
Norm
Fast
MSRN Streaming Media
Prev
Sync S
All
Sync V
Next
E-mail a link to this lecture
state:3
Local intranet

Henry Lieberman and Alex Faaborg
Common Sense Reasoning for Interactive Applications

The CYC project - Lenat

Largest Common Sense knowledge base
(~3 Megafacts)
Working since 1980's
Professional knowledge engineers
Proprietary knowledge base
High learning curve for applications developers
Limited focus on interactive applications

Common Sense Computing • MIT Media Lab

Figure 1: The web interface for watching seminars online.

The original iCam system [4] is composed of several analog cameras. For example, two cameras are mounted in the back of the lecture room for tracking the speaker. A microphone array/camera combo is placed on the podium for finding and capturing the speaking audience. Each camera is considered a virtual cameraman (VC). These VCs send their videos to a central virtual director (VD), which controls an analog video mixer to select one of the streams as output. Despite its success, the system has certain limitations. It is difficult to port the system to another lecture room. Analog cameras not only require a lot of wiring work, but also need multiple computers to digitize and process the captured videos.

We have been working on a new iCam system addressing the above issues. Instead of using analog cameras, we choose to use network cameras, which take the advantage of existing Ethernet wirings. For instance, in the back of the room, we now mount a single Sony SNC-RZ30 pan/tilt/zoom (PTZ) network camera for tracking the speaker. The audience is captured by another network camera. The captured videos are sent to a remote central computer through the network to process and compose the final output. The video mixer is no longer needed. These simplification creates some new challenges for the iCam system. In this paper, we focus on one of these issues, namely, how can we smoothly track the speaker given a single PTZ camera in the back of the lecture room.

In the old iCam system, two cameras are used to track the speaker. One of them is a static camera for tracking the lecturer's movement. It has a wide horizontal field of view (FOV) of 74 degrees and can cover the whole frontal area of the lecture room. The other camera is a PTZ camera for capturing the lecturer. Tracking results generated from the first camera will guide the movement of the second camera and keep the speaker at the center of its output video. Although working well, dual cameras not only increase the cost and the wiring/hardware complexity, but also require manual calibration during setup. In the new system, a single PTZ camera is used. In order to give a high resolution view of the speaker, the

[*] Work performed while at Microsoft Research.

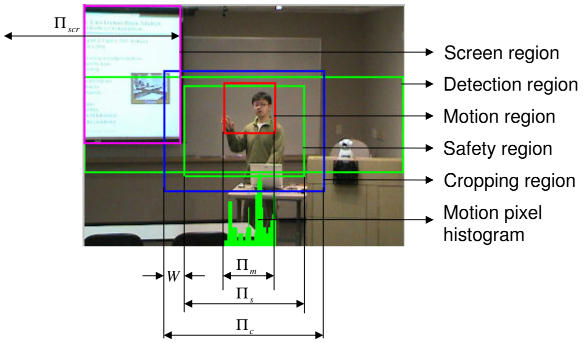


Figure 2: Regions used in speaker tracking. The horizontal segments are defined as follows: Π_m: motion region; Π_s: safety region; Π_c: cropping region; Π_{scr}: screen region; W: safety boundary. In this figure, the detection region and the screen region are manually specified during setup; the rest regions are calculated during the speaker detection/tracking.

PTZ camera can only cover a portion of the frontal area, making tracking errors hard to recover.

We address the single camera speaker tracking problem with a digital/mechanical hybrid scheme. The network camera is operated at resolution 640×480. We crop a 320×240 subregion as the output video based on where the speaker is. When the speaker moves to the boundary of the FOV, we mechanically pan/tilt the camera to keep the speaker inside the FOV. In addition, to improve the aestheticity of the lecture scene, we present an intelligent pan/zoom selection scheme according to the activity of the speaker.

The paper is organized as follows. Section 2 briefly describes our motion histogram based speaker detection algorithm. The tracking algorithm is presented in Section 3. Section 4 discribes our intelligent pan/zoom selection algorithm. Experimental results and conclusions are given in Section 5 and 6, respectively.

2 Speaker Detection

Automatic face detection and tracking have attracted a lot of interest in literature [5, 6]. However, few algorithms are deployed in practice due to robustness concerns. In the automated lecture room, the lighting can change dramatically due to slide changes. The speaker is small at low zooms and blurry because of the compression artifacts from the network camera. On the other hand, the speaker generally appears in a constrained area, which can limit the search range. In our system, we employ a motion histogram based speaker detection algorithm that is simple, sensitive and robust to lighting variations.

As shown in Figure 2, during the system setup stage, a detection region and a screen region are manually specified. In the detection region, if there is any moving object, it is most likely the speaker who is moving (motions in the screen region are ingored). Notice that this only needs to be done once for a given room. When the camera pans/tilts/zooms, we can recalculate the regions from the camera PTZ positions. Consider an incoming video frame at time instance $t_n, n = 0, 1, \ldots$. We perform a frame difference with the previous frame in the detection region. If a pixel has an intensity difference above a certain threshold, it is called a motion pixel. A horizontal motion pixel histogram is obtained, as shown in Figure 2. The height of the histogram represents the number of motion pixels in that column. Denote the histogram as $h_k^{t_n}, k = 1, \ldots, 640$. We locate the speaker by finding a motion segment $\Pi_m^{t_n} = (a_m^{t_n}, b_m^{t_n})$ on the horizontal axis (as shown in Figure 2) which satisfies:

$$\sum_{k\in\Pi_m^{t_n}} h_k^{t_n} = \sum_{k\in\mathcal{E}(\Pi_m^{t_n},\delta)} h_k^{t_n} > .70\sum_{k=1}^{640} h_k^{t_n},$$

where $\mathcal{E}(\Pi, \delta)$ is an expansion operator which expands region Π to both left and right by δ. The above equation says that the speaker motion segment is the one that contains 70% of the motion pixels. In addition, no motion pixel will be added if we expand the segment by δ. In the current implementation, we use $\delta = 5$ pixels. If the above detection fails, we will keep the motion segment unchanged, i.e., $\Pi_m^{t_n} = \Pi_m^{t_{n-1}}$.

3 Speaker Tracking

Given the speaker detection results, our goal is to generate a smooth output video following the speaker. Consider at time instance t_n, the speaker detection algorithm generates a motion segment $\Pi_m^{t_n}$. Given the previous cropping segment $\Pi_c^{t_{n-1}} = (a_c^{t_{n-1}}, b_c^{t_{n-1}})$ at t_{n-1}, the output of the algorithm is $\Pi_c^{t_n} = (a_c^{t_n}, b_c^{t_n})$, which is used to crop the subregion for the output video at t_n. The vertical position of the cropping region is fixed. In the current implementation, the cropping segment has a constant width 320, i.e., $b_c^{t_n} - a_c^{t_n} = 320$. Mechanical panning command to the camera will also be generated if the speaker moves outside the FOV, as will be discussed in Section 3.2.

3.1 Digital Tracking

Rules collected from professional videographers suggested that the speaker tracking camera should not move too often – only move when the speaker moves outside a specified zone [4]. When we perform digital cropping for the output video, the same rule applies. Given the previous cropping segment $\Pi_c^{t_{n-1}}$, we define a safety region $\Pi_s^{t_{n-1}} = (a_s^{t_{n-1}}, b_s^{t_{n-1}})$, where $a_s^{t_{n-1}} - a_c^{t_{n-1}} = b_c^{t_{n-1}} - b_s^{t_{n-1}} = W$, where W is the safety boundary, currently having a constant value $W = 40$ pixels, as shown in Figure 2. If the motion segment is empty or it falls completely inside this safety region ($\Pi_m^{t_n} \subset \Pi_s^{t_{n-1}}$), the virtual camera will not move.

Rule 1 If $\Pi_m^{t_n} \subset \Pi_s^{t_{n-1}}$, $\Pi_c^{t_n} = \Pi_c^{t_{n-1}}$.

In the following discussions, we assume $\Pi_m^{t_n} \neq \emptyset$. When $\Pi_m^{t_n}$ is not a subset of $\Pi_s^{t_{n-1}}$, we need to consider two scenarios. First, if $\Pi_m^{t_n}$ and $\Pi_s^{t_{n-1}}$ do not overlap at all ($\Pi_m^{t_n} \cap \Pi_s^{t_{n-1}} = \emptyset$), it is very likely that the speaker has completely moved outside the safety region, hence a digital panning operation should be issued. On the other hand, if $\Pi_m^{t_n}$ and $\Pi_s^{t_{n-1}}$ partially overlap ($\Pi_m^{t_n} \cap \Pi_s^{t_{n-1}} \neq \Pi_m^{t_n}$), which means the speaker is on one side of the cropping region but not out yet, we issue a panning operation to bring the speaker back to the safety region only if such status lasts for a certain period T_0, say 3 seconds. This is necessary to reduce the motion of the virtual camera.

Without loss of generality, let us assume there is a need to digitally pan the virtual camera to the right. We at least have

$b_m^{t_n} > b_s^{t_{n-1}}$. Let $d^{t_n} = b_m^{t_n} - b_s^{t_{n-1}}$. If we move the cropping region to the right by d^{t_n} at t_n, the speaker will be inside the safe region again. Unfortunately, such a scheme will cause the virtual camera hopping instead of moving smoothly toward the right.

By observing professional videographers, we found that they can pan the camera very smoothly, even though the speaker may make a sudden motion. They do not pan the camera at a very fast speed, which implies that the panning speed should be limited. In addition, human operators cannot change their panning speed instantaneously. Therefore, we apply a constant acceleration, limited speed (CALS) model for the motion of our virtual camera. Let the moving speed of the virtual camera or cropping region at time instance t_n be v^{t_n} ($v^{t_n} \geq 0$). We have:

Rule 2 If $\Pi_m^{t_n} \cap \Pi_s^{t_{n-1}} = \emptyset$ or for a period greater than T_0 we have $\Pi_m^{t_n} \cap \Pi_s^{t_{n-1}} \neq \Pi_m^{t_n}$, digital panning is performed which follows a CALS model:

$$\begin{aligned} s^{t_n} &= \operatorname{sign}(d^{t_n}), \\ v^{t_n} &= \min(v^{t_{n-1}} + \alpha s^{t_n}(t_n - t_{n-1}), v_{\max}), \\ \Pi_c^{t_n} &= \mathcal{S}(\Pi_c^{t_{n-1}}, v^{t_n}(t_n - t_{n-1})). \end{aligned}$$

where s^{t_n} is the sign of d^{t_n}; α is the constant acceleration; $v_{\max}$ is the maximum panning speed; $\mathcal{S}(\Pi, x)$ is a shift operator that shifts region Π by x.

3.2 Mechanical Tracking

The digital tracking algorithm described above can track the speaker very well inside the FOV. However, speakers may step out of the FOV and become invisible. In such cases, we need to mechanically pan the camera to follow the speaker. Notice that before the speaker steps out of the FOV, the speaker detector will always report a motion segment around the input video boundary. Therefore, the rule for mechanical tracking is very simple:

Rule 3 Mechanical panning of the camera should be issued if $\Pi_m^{t_n} \cap \Big[(0, \epsilon) \cup (640 - \epsilon, 640)\Big] \neq \emptyset$.
Here ϵ is a small value such as 2 pixels.

During the mechanical panning, the motion based speaker detection algorithm cannot detect the speaker reliably. Therefore we keep the digital cropping region fixed until the mechanical panning has stopped. We also constrain that two sequential mechanical panning motions have to be separated by a certain time interval (3 seconds). The amount of mechanical panning relies on the camera zoom level. In the current implementation it causes roughly 100 pixels of translation. This is obtained as follows. Assume the width of the speaker is 120 pixels. When mechanical panning is issued, the center of the speaker is about 60 pixels towards the boundary, and the cropping region is either (0,320) or (320,640). By panning 100 pixels, the speaker will come back to the center of the *cropped* view if he/she stays static.

4 Intelligent Pan/Zoom Selection

Mixing digital and mechanical tracking by applying Rules 1-3 together can provide us very satisfactory results – we almost never lose the speaker, and the camera motion is very smooth. However, there are aesthetic requirements beyond following the speaker in an automated lecture capturing system, as discussed below.

4.1 Panning for Mimicking a Screen Camera

Professional videographers suggested that if the speaker walks in front of the screen, or if there are animations displayed on the screen, we should turn our camera to the screen [4]. Since we do not have a dedicated screen camera, we need to use the PTZ camera to mimick a screen camera while tracking the speaker. This requires that the screen region must be kept inside the FOV for as much as possible, so that we can switch to and crop the screen region at any time. Such additional requirement brings new challenges to hybrid tracking.

To fulfill the above requirement, we give mechanical tracking higher priority than digital tracking whenever the following three conditions are satisfied. First, the screen is not fully inside the FOV. Let the screen region at time instance t_n be $\Pi_{scr}^{t_n}$. We have $\Pi_{scr}^{t_n} \cap (0, 640) \neq \Pi_{scr}^{t_n}$. Second, there is a need to perform panning towards where the screen is due to speaker motion. Third, it is safe to perform a mechanical panning. That is, after the mechanical panning, the speaker will still be inside the FOV. In such scenarios, we will override the digital panning operation with a mechanical panning operation. As a rule:

Rule 4 A mechanical panning of the camera is issued if:
1. $\Pi_{scr}^{t_n} \cap (0, 640) \neq \Pi_{scr}^{t_n}$;
2. A panning *towards the screen region* is needed according to Rule 2 or 3;
3. $\Pi_m^{t_n} \subset (\eta, 640)$ if panning to the right or $\Pi_m^{t_n} \subset (0, 640 - \eta)$ if panning to the left.
The third condition is needed so that after the mechanical panning, the speaker is very likely within the FOV. In the current implementation, we use $\eta = 160$.

4.2 Automatic Zoom Level Control

Speakers behave very differently when giving the lectures. Some speakers stand in front of their laptops and hardly move; others actively move around, pointing to the slides, writing on the whiteboard, switching their slides in front of their laptop, etc. For the former type of speakers, we want to zoom in more, so that we can see clearly the speakers' gestures and expressions. In contrast, for the latter type of speakers, we do not want to zoom in too much because that will cause the virtual camera moving around too much during the speaker tracking. In our system, we develop an automatic zoom level control scheme based on the speakers' activity.

Unlike speaker tracking, it would be distracting if we change the zoom level of the camera each time a new frame comes in. We therefore do it once every time period T_1. During each period, we accumulate the amount of pixels the cropping region has been moved. Recall at time instance t_n, the movement is $v^{t_n}(t_n - t_{n-1})$ for digital panning. Let:

$$u = \textstyle\sum_{t_n \in T_1} \; v^{t_n}(t_n - t_{n-1}) \; + M \times u_0.$$

where M is the number of mechanical pannings in period T_1; u_0 is the number of pixels moved during each mechanical panning. Currently $u_0 \approx 100$. At the end of each time period T_1, we adjust the zoom level by the following rule:

Rule 5 The zoom level of the camera is changed at the end of each time period T_1, following:

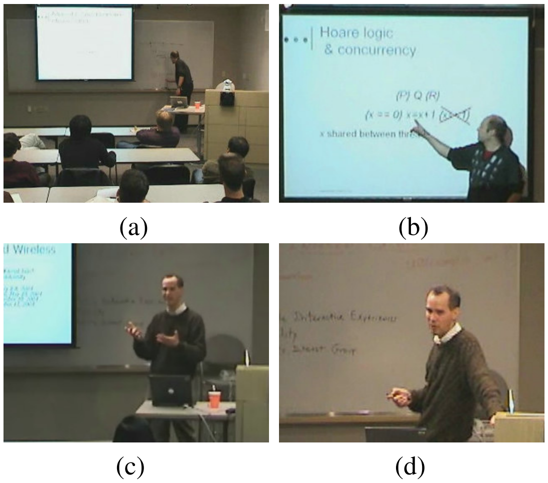


Figure 3: Output images from real lectures. They are all generated from a single PTZ camera. (a) A global view of the lecture room; (b) a screen view; (c) a speaker view that is zoomed out; (d) a speaker view that is zoomed in.

$$z_{\text{new}} = \begin{cases} \max(z_{\text{old}} - \Delta z, z_{\min}) & \text{if } u > U_1 \\ \min(z_{\text{old}} + \Delta z, z_{\max}) & \text{if } u < U_2 \\ z_{\text{old}} & \text{otherwise} \end{cases}$$

Here z_{new} is the new zoom level and z_{old} is the old zoom level. Δz is the stepsize of zoom level change. $z_{\max}$ and $z_{\min}$ are the maximum and minimum zoom levels. A greater value of zoom level means larger size speaker in the output video. $U_1 > U_2$ are two activity thresholds.

The time period T_1 is normally set to 2 minutes. The zoom action of the camera is usually hidden by switching to an audience view from the audience camera on the podium. We always start with the smallest zoom level $z_{\min}$ at the beginning of the lecture. Experimental results show that the zoom level of the speaker camera can often be stabilized within 5-10 minutes.

5 Experimental results

The new iCam system has been deployed and used on a daily basis in our lecture room for 4 months, during which 40+ lectures have been captured and broadcasted. No constraint is imposed on the moving speed of the speaker. As long as the speaker does not move out of view during the mechanical panning (this happens very rare in the 40+ captured videos), the hybrid tracker will successfully track him/her. When the speaker does get lost, the motion detector reports that there is no motion in the past 5 seconds, and the system will switch to an audience view, zoom out the speaker camera and re-detect the speaker in a larger field of view.

Figure 3 shows a few snapshots of the output videos in different lectures. Figure 3 (a) is a global view of the lecture room; (b) is an output view that is pointing to the screen region. Figure 3 (c) and (d) are from the same lecture. Figure 3 (c) was captured at the beginning of the lecture. Because the speaker was relatively inactive during his talk, after a few minutes, the camera automatically zoomed in, and we got a high resolution view as Figure 3 (d).

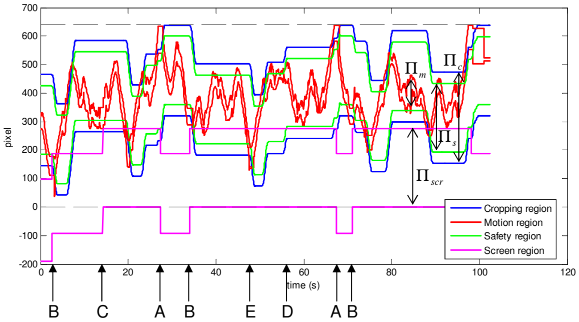


Figure 4: Various regions in a short period of a real lecture.

Figure 4 shows the speaker tracking results in a real lecture for about 100 seconds. The horizontal axis is time, and the vertical axis is pixel unit. The black dash lines are the boundary of the FOV of the capturing camera, which is $(0, 640)$. The region between the two red curves are the motion segment (Π_m) obtained by the speaker detector. It appears that the speaker is very active because the motion segment moves around a lot. The region between the two blue curves are the views cropped digitally (Π_c). Compared with the motion segment, it is much more stable. The region between the two magenta curves are the screen region (Π_{scr}).

Let us focus on a few time instances when panning are performed. In Figure 4, at the time instances marked as A, B or C, mechanical pannings are used (as only mechanical panning can cause the screen region to shift). At the instances marked as D or E, digital panning is used. At those marked as A, mechanical panning is issued because the speaker moves out of the FOV. At those marked as B and C, we perform mechanical panning because the screen is not fully inside the FOV. At those marked as B and E, we perform panning because the motion segment is completely outside the safety region. At those marked as C and D, we do panning because the motion segment is not fully inside the safety region for a while (T_0 in Rule 2).

6 Conclusions

We presented a speaker tracking algorithm in an automated lecture capturing system that can track the speaker both digitally and mechanically. The new iCam system has been successfully deployed, which is among only a few such systems that are being used on daily basis (to our best knowledge). We envision similar systems will become common practice in the near future, and greatly enhance people's collaboration and education experience.

References

[1] Stanford Online, http://scpd.stanford.edu/scpd/students/onlineclass.htm.

[2] UC Berkeley Online Learning, http://learn.berkeley.edu.

[3] Columbia Video Network, http://www.cvn.columbia.edu.

[4] Y. Rui, A. Gupta and J. Grudin, "Videography for telepresentations", *Proc. of ACM CHI*, pp.457–464, 2003.

[5] D. Comaniciu and V. Ramesh, "Robust detection and tracking of human faces with an active camera", *IEEE Visual Surveillance*, 2000.

[6] J. Yang and A. Waibel, "A real-time face tracker", *WACV 1996*.