Building an Intelligent Camera Management System

Yong Rui, Liwei He, Anoop Gupta and Qiong Liu[1]

Collaboration and Multimedia Systems Group, Microsoft Research

One Microsoft Way

Redmond, WA 98052-6399

{yongrui, lhe, anoop}@microsoft.com and q-liu2@ifp.uiuc.edu

ABSTRACT

Given rapid improvements in storage devices, network infrastructure and streaming-media technologies, a large number of corporations and universities are recording lectures and making them available online for anytime, anywhere access. However, producing high-quality lecture videos is still labor intensive and expensive. Fortunately, recent technology advances are making it feasible to build automated camera management systems to capture lectures. In this paper we report our design of such a system, including system configuration, audio-visual tracking techniques, software architecture, and user study. Motivated by different roles in a professional video production team, we have developed a multi-cinematographer single-director camera management system. The system performs lecturer tracking, audience tracking, and video editing all fully automatically, and offers quality close to that of human-operated systems.

Keywords

Automated camera management, Virtual cinematographer, Virtual director, Lecturer tracking, Sound source localization.

1 INTRODUCTION

In recent years, with rapid pace of technological advances and accompanying emphasis on life-long learning, both universities and corporations are offering more lectures, seminars, and classes to teach and train students and employees. To accommodate audiences' time and/or space conflicts, many of these lectures are made available online, allowing people to attend remotely, either live or on-demand. For instance, at Stanford University, lectures from over 50 courses are made available online every quarter [19]. The Microsoft Technical Education Group (MSTE) has supported 367 on-line training lectures with more than 9000 viewers from 1998 to 1999 [12].

While online broadcasting and publishing of lectures is gaining momentum, a large number of lectures are still not recorded or published today. A key barrier is the cost to produce those lectures. While the cost of recording equipment and disk storage is becoming lower every day, the cost of hiring camera operators is actually getting more expensive.

To produce a high-quality lecture video, human operators need to perform many tasks, including tracking a moving lecturer, locating a talking audience member, showing presentation slides, and selecting the most suitable video from multiple cameras. Consequently, high-quality videos are usually produced by a video production team that includes a director and multiple cinematographers. To decrease the recording cost, a single human operator may take on the roles of the director and multiple cinematographers simultaneously. However, it takes years of training and experience for a human operator to perform all these tasks and hiring such an operator is therefore still quite expensive.

As computer technology advances, an alternative to hiring a human operator is to construct a fully automated lecture-room camera management system. While this was almost impossible to do a decade ago, recent computer vision and signal processing techniques are making it feasible to start this process. In order to build such a system, we will need the following important modules, as suggested by the human video production team paradigm:

- The lecturer-tracking virtual cinematographer (VC) module that tracks and films the lecturer.
- The audience-tracking VC module that locates and films audience members.
- The slide-tracking VC module that monitors and films lecture slides (e.g., PowerPoint projection).
- The overview VC module that looks at the whole podium area. This is a safe back-up module, in case the other VC modules are not ready.
- The virtual director (VD) module that selects the final video stream from the multiple VCs' video cameras.

While there have been previous attempts at building an automated camera management system, few of them are addressing it at a complete-system level. For example, there exist various computer-vision and microphone-array tracking techniques, but how to integrate them in the context of a lecture room environment has not been deeply studied. Furthermore, there is almost no attempt on the explicit modeling of the VC/VD modules to accomplish similar goals that a professional video production team can achieve. Finally, little is done on a systematic study of professional video production rules and usability experiments. To address the above issues, we have started

[1] This author was a summer intern with Microsoft Research.

our research effort towards building a fully automated camera management system. We have reported how we collect professional video production rules, realize them in the system, and conduct user studies in our CHI'01 paper [13]. In this paper, we will focus on the system and technology side of our effort, which includes the development of the system architecture and various important VC and VD modules.

The rest of the paper is organized as follows. In Section 2, we provide a brief review of related research on lecture room automation. In Section 3, we present an overview of our system. In Sections 4 to 6, we describe the lecturer-tracking VC, audience-tracking VC and VD modules in great detail. We report experimental results in Section 7 and present concluding remarks in Section 8.

2 RELATED WORK

In this section, we provide a brief review of related work from two aspects: individual tracking techniques and existing automated lecture capture systems.

2.1 Tracking techniques

Tracking technology is required both to keep the camera focused on the lecturer and to display audience members when they talk. Depending on if the tracked people need to wear extra sensors, there are obtrusive tracking techniques and unobtrusive tracking techniques. The former includes infrared sensors, magnetic sensors and ultra-sound sensors. The latter includes various computer vision and microphone array techniques.

For obtrusive tracking, the human to be tracked is required to wear an IR or magnetic devices that emits electric or magnetic signals. A nearby receiver unit then uses the signal to locate the lecturer. This technique has been used in both commercial products (e.g., ParkerVision [15]) and research prototypes [14]. Even though tracking is usually reliable using this technique, we consider wearing an extra device during the lecture to be inconvenient and obtrusive.

For un-obtrusive tracking, there exists rich literature in computer-vision-based techniques. Typical ones include skin-color-based tracking [20], motion-based tracking [8], and shape-based tracking [1]. Another un-obtrusive technique is based on microphone array sound source localization (SSL), and it is most suited for locating talking audience members in a lecture room. There exist various SSL techniques in both research prototypes (see Section 5) and commercial products (e.g., PictureTel [16] and PolyCom [17]).

To summarize, different techniques exist for tracking objects. Obtrusive solutions are more reliable but less convenient. Vision and microphone array based techniques are unobtrusive and their quality is quickly approaching that of the obtrusive ones, especially when we put them in the context of lecture room camera management.

2.2 Related systems

Several projects exist for lecture room automation [3,14,21]. In [21], Wang and Brandstein report a real-time talking head tracker that targets automated video conferencing. However, such a system is only a component in our system, e.g., the lecturer-tracking module. No attempt is made in their work to construct a complete lecture-room camera management system.

In [14], Mukhopadhyay and Smith present an interesting lecture-capturing system. They use a moving camera to track the lecturer and a static camera to capture the entire podium area. Though there are overlaps between this system and ours, the focus is quite different. Because their system records multiple multimedia streams, e.g., audio, video, slides and HTML text, independently, synchronization of those streams is a key focus in their system. In our system, the various VC/VD modules cooperatively film the lecture in a seamless way, and synchronization is not a concern. Furthermore, our main focus is on sophisticated camera management strategies.

Bellcore's AutoAuditorium [3,7] is one of the pioneers in lecture room automation. Among existing systems, it is the closest to ours and has influenced our system design. The AutoAuditorium system uses multiple cameras to capture the lecturer, the stage, the screen, and the podium area from the side. An AutoAuditorium director selects which video to show to the remote audience based on heuristics. Though there are overlaps between the AutoAuditorium system and ours, there are substantial differences in the richness of VD's video editing rules, the types of tracking modules used, and the overall system architecture.

Additional research projects exist for exploring other aspects of lecture automation, such as Classroom2000's effort on notes-capturing [5], STREAM's effort on cross-media indexing [7], and Gleicher and Masanz's work on off-line lecture video editing [10]. Furthermore, several researchers have examined video mediated communication (e.g. Hydra, LiveWire, Montage, Poleholes, Brandy Bunch, and FlyCam) in the field of teleconferencing [6,9]. However, given its loose relation to this work, we do not elaborate on it here.

To summarize, progress has been made in the field of individual tracking techniques and to a lesser extent at the complete system level during the past few years. This paper focuses on how we integrate individual tracking techniques in the context of a lecture room environment, and design an effective camera management framework that accomplishes similar goals that a professional video production team achieves.

3 CAMERA MANAGEMENT SYSTEM OVERVIEW

As discussed in Section 1, in order to build an automated camera management system, we will need one or more of the lecture-tracking, audience-tracking, overview, slide-tracking, and director modules.

Considering different roles taken by the VCs and the VD, we develop a two-level structure in our system. At the lower level, VCs are responsible for basic video shooting tasks, such as tracking a lecturer or locating a talking audience. Each VC periodically reports its status S_T, camera zoom level Z_L, and tracking confidence level C_L to the VD. At the upper level,

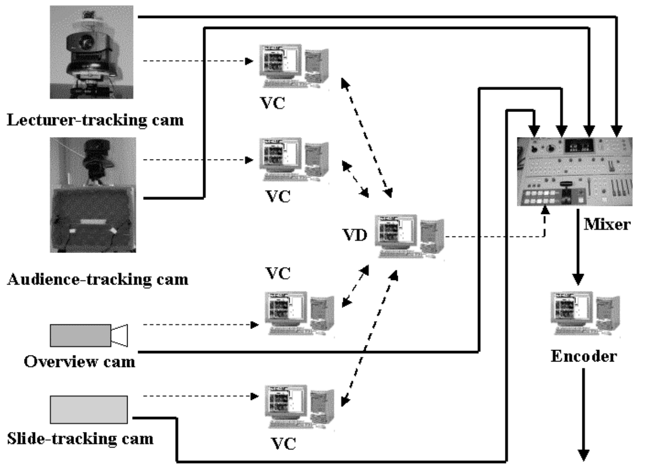


Figure 1 System block diagram. Dashed lines indicate control and status signals. Solid lines indicate video data. VC stands for virtual cinematographers and VD stands virtual director.

the VD collects all the S_T, Z_L, and C_L from the VCs. Based on the collected information and history data, the VD then decides which VC's camera is chosen as the final video output and switches the video mixer to that camera. The VD also sends its decision D_S back to the VCs to coordinate further cooperation. The edited lecture video is then encoded for both live broadcasting and on-demand viewing. The system block diagram is shown in Figure 1. One thing worth pointing out is that even though we represent various VC/VDs with different computers, they can actually reside in a single computer running different threads.

All the VCs have the following four components:

1. Sensors that sense the world, just like human cinematographers have eyes and ears.
2. Cameras that capture the scenes, just like human cinematographers have their video cameras.
3. Framing rules that control the camera operation, just like human cinematographers have their framing knowledge.
4. VC-VD communication rules, just like human cinematographers need to communicate with their director.

As our first attempt to the automated camera management system, we have chosen a minimum set of VC/VDs. Specifically, we have one lecturer-tracking VC, one audience-tracking VC, one slide-tracking VC, one overview VC, and one VD in our current system. Figure 2 shows a top view of one of our organization's lecture rooms, where our system is installed. The lecturer normally moves behind the podium and in front of the screen. The audience area is in the right-hand side in the figure and includes about 60 seats. There are four cameras in the room: a lecturer-tracking camera, an audience-tracking camera, a static overview camera, and a slide-tracking camera (i.e., a scan-converter) that captures whatever is being displayed on the screen from the projector (typically PowerPoint slides).

The user interface for the remote audience is shown in Figure 3. The left portion of the interface is a standard Microsoft MediaPlayer window. The outputs of lecture-tracking camera, audience-tracking camera, and overview camera are first edited by the VD and then displayed in this window. The output of the slide-tracking camera is displayed directly on the right-hand side window. An obvious alternative to this interface is to eliminate the right-hand side window and integrate the output of slide-tracking camera directly into the left-hand side window. However, the interface shown in Figure 3 is the interface already in use by our organization's lecture-capture team. In order to conduct a controlled comparison study with the human operator, we use the same interface for our system. Note that a similar user interface has also been used in [14].

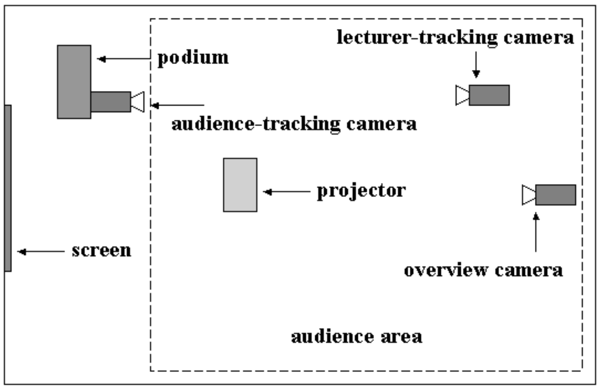


Figure 2. Top view of the lecture room layout

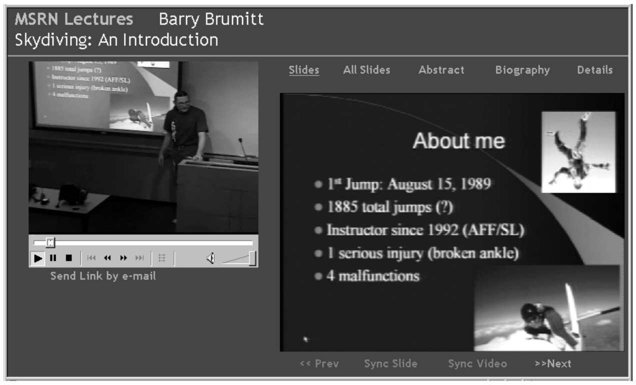
For the slide-tracking VC and overview VC, because they constantly and statically look at the screen and the podium area, no tracking is needed. Therefore no sensor or framing rule is needed for these two modules. For the lecturer-tracking VC and audience-tracking VC modules, it is much more complex of how we should select and set up the sensors and implement the framing rules. We will present detailed description for lecturer-tracking VC and audience-tracking VC modules in Sections 4 and 5.

4 LECTURER-TACKING VIRTUAL CINEMATOGRAPHER

The lecturer is a key object in the lecture. Therefore, accurately tracking and correctly framing the lecturer is of great importance. The responsibility of the lecturer-tracking VC is to follow the lecturer's movement and gestures for a variety of shots: close-up to focus on expression, median shots for gestures, and long shots for context. As discussed in Section 3, there are four components of a VC module: camera, sensor, framing rules and communication rules. We next discuss them in the context of the lecturer-tracking VC module.

4.1 Camera

For VCs, the most flexible cameras they can use are the pan/tilt/pan cameras (active cameras). Currently there are two major active cameras on the market, i.e., Sony's EVI D30/31 and Canon's VC-C3. These two cameras have similar quality and functionality. They both have step motors to drive pan/tilt and zoom simultaneously. The EVI camera pans between [-100, +100] degrees, tilts between [–25, +25] degrees, and has a highest zoom level of 12x. The



MSRN Lectures Barry Brumitt
Skydiving: An Introduction

About me

- 1st Jump: August 15, 1989
- 1885 total jumps (?)
- Instructor since 1992 (AFF/SL)
- 1 serious injury (broken ankle)
- 4 malfunctions

Figure 3. The user interface for remote audience

VC-C3 camera pans between [-90, +90] degrees, tilts between [-25, +30] degrees, and has a highest zoom level of 10x. Because of EVI camera's slightly better performance, we choose it as the active camera in our system.

4.2 Sensor

As detailed in Section 2, there are various tracking techniques available. We exclude the obtrusive tracking techniques from further discussion because of their unnecessary inconvenience. Between the computer-vision and microphone-array techniques, the former is better suited for tracking the lecturer.

In unconstrained environment, reliably tracking a target using computer vision techniques is still an open research problem. For example, some techniques can only track for a limited duration before the target begins to drift away; others require manual initialization of color, snakes, or blob [1]. While perfectly valid in their targeted applications, these approaches don't satisfy our goal of building a fully automated system.

Tracking a lecturer in the lecture room environment imposes both challenges and opportunities. On one hand, the lecture room is usually dark and the lighting condition changes drastically when the lecturer switches from one slide to another. The poor and changing lighting condition thus makes most color-based and edge-based tracking fail. On the other hand, we can take advantages of the following domain knowledge to make our tracking task manageable:

1. The lecturer is usually moving or gesturing during the lecture so that motion information becomes an important tracking cue.
2. The lecturer's moving space is usually confined to the podium area, which allows a tracking algorithm to predefine a tracking region to help distinguish lecturer's movement from that of the audience.

Domain knowledge 1 allows us to use simple frame-to-frame difference to conduct tracking so that we can develop a real-time system. Domain knowledge 2 allow us to specify a podium area in the video frame so that the motion-based tracking algorithm is not distracted by audience' movement. We next discuss some possible sensor-camera setups that we have experimented with that may accomplish this motion-based tracking requirement.

4.3 Sensor-Camera setup

4.3.1 Single active camera

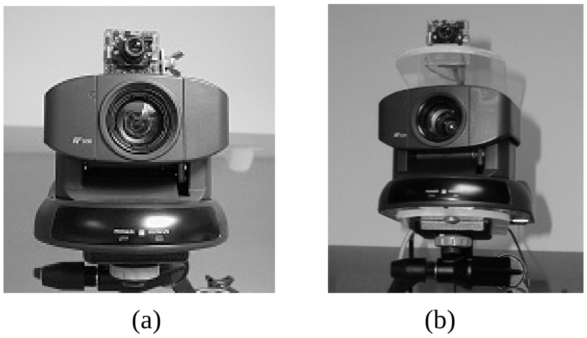
The simplest setup for lecturer tracking is to use a single active camera as both the sensor and the camera. Even though simple, there are two potential problems:

1. Because we use frame-to-frame differencing motion tracking, to avoid extra motion compensation steps, it is necessary to stop the camera first before capturing the frames. The stop-and-move mode results in unpleasant video capturing.
2. Because we want tight shots of the lecturer, the camera normally operates in the high zoom mode. The very narrow field of view (FOV) in the high zoom mode causes the camera to lose track quite easily.

4.3.2 A wide-angle camera attached to the active camera

To increase the active camera's FOV, we can use a wide-angle camera as the sensor and attach it directly to the active camera, as shown in Figure 4(a). An inexpensive wide-angle camera is Super Circuit's PC60XSA, which has a FOV of 74 degree at a cost of $60. This camera can comfortably cover the whole podium area when placed at the back of the lecture room. When the two cameras are attached and optical axes aligned, the lecturer-tracking VC uses the wide-angle camera to locate the target and then uses the target location to control the active camera.

Because of its wide FOV, the wide-angle camera introduces big radial distortion that is proportional to the distance from its frame center. Normally, we need to correct this distortion via camera intrinsic parameter estimation before we can calculate the target location. In this setup, however, because the two cameras move together, the target mostly will appear close to the center of the frame and it is not necessary to conduct the correction. To summarize, this particular setup addresses Problem 2, but Problem 1 remains.



(a) (b)

Figure 4. Sensor-camera setup for the lecturer-tracking VC. The top portion is a wide-angle camera and the lower portion is an active camera. (a) A wide-angle camera is directly attached to the active camera; (b) A wide-angle camera is placed on a static base right above the active camera.

4.3.3 Static wide-angle camera with moving active camera

In order to address Problem 1, the wide-angle camera should not move with the active camera. We can mount the wide-angle camera on a static base right above the active camera (see Figure 4 (b)) and align the wide-angle camera's optical axis to that of the active camera's home position.

This camera setup successfully solves both Problems 1 and 2. But this comes with a cost. Compared with the setup in Section 4.3.2, this setup needs an extra camera calibration step because of the wide-angle camera's distortion. In contrast to the setup in Section 4.3.2, now the lecturer can appear anywhere in the wide-angle camera frame, including its boundary's highly distorted regions. Among various camera intrinsic parameter calibration algorithms, Zhang's approach is one of the best and easy to use [23]. We therefore adopt it in our system to un-distort the wide-angle video frames. While this is an extra step, it is only needed once after we assemble the two-camera device. Because of its superior performance than those in Sections 4.3.1 and 4.3.2, this sensor-camera setup is chosen in our system.

4.4 Framing strategy and VC-VD communication

It is worth emphasizing here that tracking in a camera management environment is very different from that in a conventional computer vision application. In the latter situation, the goal is to detect and follow the target as closely as possible. In contrast, in camera management applications, not only do we want to follow the target, what is more important is that we want to instrument the camera intelligently such that viewers will enjoy watching the captured video. In Section 4.3, we have discussed how we set up a two-camera system and obtain the lecturer location from the wide-angle camera. In this section we will focus on how to properly frame the lecturer in order to produce close-to-professional-quality video.

Two of the most important video production rules we have gathered from professionals are [13]

1. Avoid unnecessary camera motion as much as possible.
2. Shots should be as tight as possible.

These two rules seem to result in conflicting camera control strategies. In order to maintain tight shots, we need to follow the lecturer as closely as possible. But following the lecturer constantly will produce unpleasant videos as per rule 1. After discussions with professionals and watching many professional-produced videos, we have developed a history-based reduced-motion camera control strategy.

To comply with rule 1, once the camera locks and focuses on the lecturer, it will maintain static until the lecturer moves out of the frame or the VD switches to a different camera. Let (x_t,y_t) be the location of the lecturer estimated from the wide-angle camera. According to the above control strategy, before the VD cuts to the lecturer-tracking camera at time t, the lecturer-tracking VC will pan/tilt the camera such that it locks and focuses on location (x_t,y_t). To determine the zoom level of the camera, lecturer-tracking VC maintains the trajectory of lecturer location in the past T seconds, $(X,Y) = \{(x_1,y_1), \ldots, (x_t,y_t), \ldots, (x_T,y_T)\}$. Currently, T is set to 10 seconds. The bounding box of the activity area in the past T seconds is then given by a rectangle (X_L, Y_T, X_R, Y_B), where they are the left-most, top-most, right-most, and bottom-most points in the set (X,Y). If we assume the lecturer's movement is piece-wise stationary, we can use (X_L, Y_T, X_R, Y_B) as a good estimate of where the lecturer will be in the next T' seconds. The zoom level Z_L is calculated as follows:

$$Z_L = \min\left(\frac{HFOV}{\angle(X_R, X_L)}, \frac{VFOV}{\angle(Y_B, Y_T)}\right) \qquad (1)$$

where *HFOV* and *VFOV* are the horizontal and vertical field of views of the active camera, and $\angle(\bullet,\bullet)$ represents the angle spanned by the two arguments in the active camera's coordinate system.

The above framing strategies achieve good balance between the two rules and the lecturer-tracking VC controls the active camera in such a way that not only it has the fewest movements, but also it maintains tightest shots as possible.

As for VC-VD communication, S_T takes on values of {READY, NOTREADY}. S_T is READY when the camera locks and focuses on the target. S_T is NOTREADY when the camera is still in motion or the target is out of the view of the camera. Z_L is computed by using Equation (1) and normalized to the range of [0,1]. C_L is 1 if the target is inside the active camera's FOV and is 0 otherwise.

5 AUDIENCE-TRACKING VIRTUAL CINEMATOGRAPHER

Showing the audience members who are asking questions is important to make useful and interesting lecture videos. Because the audience area is usually quite dark and audience members may sit quite close to each other, tracking audience using computer-vision-based technique will hardly work. A better sensing modality is based on microphone arrays, where audience-tracking VC first estimates the sound source direction using the microphones and then uses the estimation to control the active camera.

5.1 Sensor

In general, there are two types of microphones in terms of their response direction ranges. They are omni-directional microphones and uni-directional microphones (also called cardioid microphones). Within cardioid microphones, there are different levels of directionality available, including sub-cardioid, regular cardioid, hyper-cardioid and super-cardioid microphones, decreasing in direction ranges in that order [18].

Because super-cardioid microphones are highly directional, it is possible that we can locate sound source directly without using any signal processing techniques. An obvious solution is to put multiple super-cardioid microphones in a fan-shaped configuration, with each microphone facing outwards. The audio outputs of these microphones are then connected to a PC where each microphone's output energy is computed. The microphone that has the largest energy is the active microphone and the direction it is facing is the sound source direction.

While it is easy to implement, there are several problems with this configuration. Firstly, the directionality resolution is not high enough. For example, even for high-end cardioid microphones, their direction range is seldom less than 50 degrees. With such a coarse resolution, the active camera can hardly get any close-up view. Secondly, which is more severe, if a target is in between two microphones, there will be ambiguity on where the target is. We can solve this problem by having overlapping microphones, but that will significantly increase the system's cost and array's form factor. Finally, the super-cardioid microphones are quite expensive. They can range from a few hundred dollars up to tens of thousands of dollars, which defeats our goal of developing a cheap and easy-to-use system.

Fortunately, we can develop sound source localization (SSL) techniques that use much cheaper omni-directional or regular cardioid microphones, yet achieve better performance than those expensive super-cardioid microphones. As we mentioned in Section 2, there are commercial products available that implement SSL steered tracking cameras (e.g., PictureTel [16] and PolyCom [17]). However, they do not expose their APIs and do not satisfy our framing strategies. For example, their response time is not quick enough and they do not accept commands such as a slow pan from left to right. To have full control of the audience-tracking VC module, we decided to develop our own SSL techniques.

There are three types of SSL techniques exist in the literature, i.e. steered-beamformer-based, high-resolution spectral-estimation-based, and time-delay-of-arrival (TDOA) based [4]. The first two types of techniques are computationally expensive and not suitable for real-time applications. So far, the winning technique is the TDOA-based techniques, where the measure in question is not the acoustic data received by the sensors, but rather the time delays between each sensor.

Within various TDOA approaches, the generalized cross-correlation (GCC) approach receives the most research attention and is the most successful one [4,22]. Let *s(n)* be the source signal, and *x₁(n)* and *x₂(n)* be the signals received by the two microphones:

$$\begin{aligned} x_1(n) &= as(n-D) + h_1(n) * s(n) + n_1(n) \\ x_2(n) &= bs(n) + h_2(n) * s(n) + n_2(n) \end{aligned} \tag{2}$$

where D is the TDOA, a and b are signal attenuations, $n_1(n)$ and $n_2(n)$ are the additive noise, and $h_1(n)$ and $h_2(n)$ represent the reverberations. Assuming the signal and noise are uncorrelated, D can be estimated by finding the maximum GCC between $x_1(n)$ and $x_2(n)$:

$$\begin{aligned} D &= \arg\max_{\tau} \hat{R}_{x_1x_2}(\tau) \\ \hat{R}_{x_1x_2}(\tau) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega) G_{x_1x_2}(\omega) e^{j\omega\tau} d\omega \end{aligned} \tag{3}$$

where $\hat{R}_{x_1x_2}(\tau)$ is the cross-correlation of $x_1(n)$ and $x_2(n)$, $G_{x_1x_2}(\omega)$ is the Fourier transform of $\hat{R}_{x_1x_2}(\tau)$, i.e., the cross power spectrum, and $W(\omega)$ is the weighting function.

In practice, choosing the right weighting function is of great significance for achieving accurate and robust time delay estimation. As can be seen from equation (2), there are two types of noise in the system, i.e., the background noise $n_1(n)$ and $n_2(n)$ and reverberations $h_1(n)$ and $h_2(n)$. Previous research suggests that the maximum likelihood (ML) weighting function is robust to background noise and phase transformation (PHAT) weighting function is better dealing with reverberations [22]:

$$\begin{aligned} W_{ML}(\omega) &= \frac{1}{|N(\omega)|^2} \\ W_{PHAT}(\omega) &= \frac{1}{|G_{x_1x_2}(\omega)|} \end{aligned}$$

where $|N(\omega)|^2$ is the noise power spectrum.

It is easy to see that the above two weighting functions are at two extremes. That is, $W_{ML}(\omega)$ puts too much emphasis on "noiseless" frequencies, while $W_{PHAT}(\omega)$ completely treats all the frequencies equally. To simultaneously deal with background noise and reverberations, we have developed a technique similar to that in [22]. We start with $W_{ML}(\omega)$, which is the optimum solution in non-reverberation conditions. To incorporate reverberations, we define a generalized noise as follows:

$$|N'(\omega)|^2 = |H(\omega)|^2 |S(\omega)|^2 + |N(\omega)|^2$$

Assuming the reverberation energy is proportional to the signal energy, we have the following weighting functions:

$$W(\omega) = \frac{1}{\gamma |G_{x_1x_2}(\omega)| + (1-\gamma)|N(\omega)|^2}$$

where $\gamma \in [0,1]$ is the proportion factor.

Once the time delay D is estimated from the above procedure, the sound source direction can be estimated given the microphone array's geometry. As shown in Figure 5, let the two microphones be at locations A and B, where AB is called the baseline of the microphone array. Let the active camera be at location O, whose optical axis is perpendicular to AB. The goal of SSL is to estimate the angle $\angle COX$ such that the active camera can point at the right direction. When the distance of the target, i.e., |OC|, is much larger than the length of the baseline |AB|, the angle $\angle COX$ can be estimated as follows [4]:

$$\angle COX \approx \angle BAD = \arcsin\frac{|BD|}{|AB|} = \arcsin\frac{D \times v}{|AB|} \tag{4}$$

where D is the time delay and v = 342 m/s is the speed of sound traveling in air.

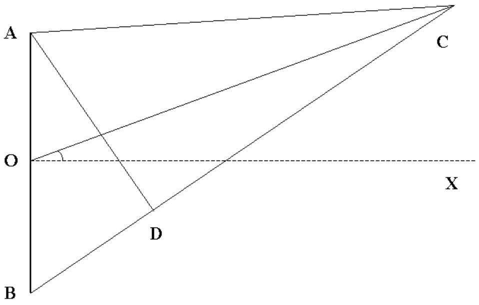


Figure 5. Sound source localization

5.2 Sensor-Camera setup

It is clear that to estimate the panning angles of the active camera, we need at least two microphones in a configuration similar to that in Figure 5. If we want to estimate the tilting angle as well, we will need a third microphone. By having four microphones in a planar grid, we can also estimate the distance of the sound source in addition to the pan/tilt angles [4]. Of course, adding more microphones will increase the system complexity as well. In our particular application, however, we can have simpler solutions. Because audience members are typically sitting on their seats, if the active camera is mounted slighted above the eye level, tilting is not necessary. Furthermore, because estimating sound source distance is still less robust than estimating sound source directions, in our current system we focus our attention only on how to accurately control the panning angles of the active camera. We next discuss several potential sensor-camera setups.

5.2.1 Microphones attached to the active camera

Just like a human head has two ears on it, we can attach two microphones to the left and right sides of the active camera (see Figure 6(a)). The good part of this configuration is that we only need to estimate if the sound source is from left or from right, and no need to know the exact direction. This configuration, however, has two major problems in the lecture room context.

One is that the camera's width is not wide enough to be a good baseline for the microphone array. As can be seen from Equation (4), the SSL resolution is inversely proportional to the length of the baseline. A small baseline will result in poor resolution. A solution to this problem is to attach an extension structure to the camera and then attach the microphones to that structure to extend the microphone array's baseline, as shown in Figure 6(a). But this solution leads to the second problem of this configuration, *distraction*. Local audience members do not want to see moving objects that may distract their attention. That is why in most of lecture rooms the tracking cameras are hidden inside a dark dome. In this configuration, however, since the microphones are attached to the active camera, the whole tracking unit has to be outside the dome in order for the microphones to hear. By extending the baseline of the microphone array, we will increase the distraction factor as well. The distraction factor of such a setup makes it unusable in real lecture rooms.

5.2.2 Static microphones and moving camera

An alternative solution is to detach the microphone array from the active camera, but still keep the microphone array's baseline to be perpendicular to the camera's optical axis to ensure easy coordinate system transformation (Figure 6(b)). By separating the microphones from the camera, we have a more flexible configuration. For example, the camera can be hidden inside a dark dome above the microphone array. In addition, because the microphone array is static, we can have a much larger baseline without causing any movement distraction. In our current system, the baseline is 22.5cm.

5.3 Framing strategy and VC-VD communication

Because our current audience-tracking VC only controls the panning angle of the active camera, while keeping a constant tilting angle and zoom level, the framing strategy is relatively simple. Once the microphone array detects a sound source and estimates the sound source direction with enough confidence, the audience-tracking VC will pan the active camera to that direction. Status S_T is READY when the active camera locks on the target and stops moving. S_T is NOTREADY when the active camera is still panning toward the sound source direction.

Because we have fixed the zoom level of the audience-tracking VC's camera, Z_L is a constant (e.g., 0.5) all the time. As for a good estimate of the confidence level C_L, there is a natural quantity associated with GCC that we can use: the correlation coefficient ρ. ρ represents how correlated two signals are, and thus represents how confident the TDOA estimate is. Its value always lies in the range of [-1,+1]. C_L can then be computed as follows:

$$C_L = (\rho + 1)/2$$

$$\rho = \frac{\hat{R}_{x_1 x_2}(\tau)}{\sqrt{\hat{R}_{x_1 x_1}(0)\hat{R}_{x_2 x_2}(0)}}$$

where $\hat{R}_{ij}(\tau)$ is defined in Equation (3).

In addition to promptly showing the talking audience members, an important framing rule for audience-tracking VC is to have the ability to show a general shot of the

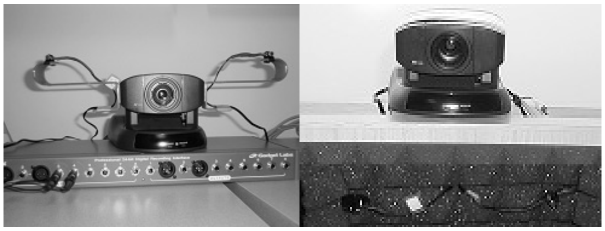

(a) (b)

Figure 6. (a) Microphones are attached to extension structures of the active camera; (b) Microphones (lower portion of the figure) are separated from the active camera.

audience members even though none of them is talking. This added ability makes the recorded lecture more interesting and useful to watch. This type of shots is normally composed of a slow pan from one side of the lecture room to the other. To support this framing rule, the audience-tracking VC's status S_T takes an extra value of {GENERAL} in addition to {READY, NOTREADY}, and the VD's decision D_S takes an extra value of {PAN} in addition to {LIVE, IDLE}. Status S_T equals {GENERAL} when the camera is not moving and the microphone array does not detect any sound source either. After the VD receives status S_T = {GENERAL} from the audience-tracking VC, it will decide if it wants to cut to the audience-tracking VC's camera. If so, it will send decision D_S = {PAN} to audience-tracking VC. Upon receiving this decision, audience-tracking VC will slowly pan its camera from one side of lecture room to the other.

6 VIRTUAL DIRECTOR

The responsibility of the VD module is to gather and analyze reports from different VCs, to make intelligent decisions on which camera to select, and to control the video mixer to generate the final video output. Just like video directors in real life, a good VD module observes the rules of the cinematography and video editing in order to make the recording more informative and entertaining. The specific rules for making a good lecture recording are reported in our CHI'01 paper [13]. Here we will focus on how we design a flexible VD module such that it can easily encode various editing rules. We propose to equip the VD with two tools: an event generator to trigger switching from one camera to another, and a finite state machine (FSM) to decide which camera to switch to.

6.1 Event generator – when to switch

The event generator has an internal timer to keep track of how long a particular camera has been on, and a report vector $R = \{S_T, Z_L, C_L\}$ to maintain each VC's status S_T, zoom level Z_L and confidence level C_L. The event generator generates two types of events that cause the VD to switch cameras. One event type is STATUS_CHANGE. It happens when a VC changes its status, e.g., from READY to NOTREADY. The other even type is TIME_EXPIRE. It triggers if a camera has been on for too long.

6.1.1 STATUS_CHANGE *events*

The event generator maintains a report vector $R = \{S_T, Z_L, C_L\}$ to keep track of all VC's status. Because the slide-tracking camera's video is shown in a separate window (see Figure 3), there are only three cameras need to be dispatched by the VD. The report vector therefore has three elements, representing the current information from lecturer-tracking VC, audience-tracking VC and overview VC, in that order. $S_T[1]$ takes two values {READY, NOTREADY}. $S_T[2]$ takes three values {READY, NOTREADY, GENERAL}. Because the overview camera is the safe back-up, $S_T[3]$ takes only one value {READY}. Together, they represent a combination of 2x3x1=6 overall statuses for the whole system.

The event generator constantly monitors the report vector $R = \{S_T, Z_L, C_L\}$. If any of the three VCs reports a status change, e.g., from READY to NOTREADY, a STATUS_CHANGE event is generated and sent to the VD. The VD will then take actions to handle this event (e.g., switch to a different camera).

6.1.2 TIME_EXPIRE *events*

In addition to the STATUS_CHANGE events described in Section 6.1.1, the event generator also generates TIME_EXPIRE events.

In video production, switching from one camera to another is called a *cut*. The period between two cuts is called a video *shot*. An important video editing rule is that a shot should not be too long or too short. To ensure this rule, each camera has its minimum shot duration D_{MIN} and its maximum allowable duration D_{MAX}. If a shot length is less than D_{MIN}, no camera-switching will be made. On the other hand, if a camera has been on longer than its D_{MAX}, a TIME_EXPIRE event will be generated and sent to the VD. Currently, D_{MIN} is set to 5 seconds for all cameras based on professionals' suggestions.

Two factors affect a shot's length D_{MAX}. One is the nature of the shot and the other is the quality of the shot. The nature of shot determines a base duration D_{BASE} for each camera. For example, lecturer-tracking shots are longer than overview shots, because they are in general more interesting. Based on the rules we have collected from the professionals [13], the current base duration D_{BASE} is set to 60 sec, 10 sec, 5 sec, and 40 sec for lecturer-tracking camera when S_T = READY, audience-tracking camera when S_T = READY and S_T = GENERAL, and overview camera when S_T = READY, respectively.

The quality of a shot is defined as a weighted combination of the camera zoom level Z_L and tracking confidence level C_L. Quality of the shot affects the value of D_{MAX} in that high quality shots should last longer than low quality shots. The final D_{MAX} is therefore a product of the base length D_{BASE} and the shot quality:

$$D_{MAX} = D_{BASE} \times (\alpha Z_L + (1-\alpha) C_L)$$

where α is chosen experimentally. We use $\alpha = 0.4$ in our current system.

6.2 FSM – where to switch

In Section 6.1, we have discussed how the event generator generates events to trigger the VD to switch cameras. In this section we will discuss which camera the VD switches to upon receiving the triggering events.

In [11], He *et. al.* proposed a hierarchical FSM structure to simulate a virtual cinematographer in a virtual graphics environment. This work influenced our design of our VC and VD modules. Compared with their system, our system works in the real world instead of a virtual world, which imposes many physical constrains on the way we can manipulate cameras and people. For example, it was not possible in our system to obtain a shot from an arbitrary angle. Furthermore, while their system

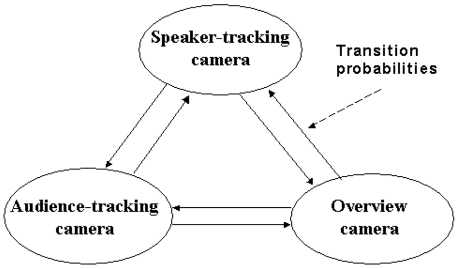


Figure 7. A three-state FSM

can assume all the cameras are available at all times in the virtual environment, our system cannot have that assumption because targets may not be in the FOV of some cameras. This imposes much greater complexities in our VD module.

Figure 7 shows a three-state FSM where the lecturer-tracking VC camera, audience-tracking VC camera, and overview VC camera are each represented by a state. The three states are fully connected to allow any transition from one state to another.

When to transit is triggered by the events described in Section 6.1. Where to transit is determined by the transition probabilities in the FSM. Professional video editing rules can easily be encoded into those probabilities. For example, a cut is more often made from the lecturer-tracking camera to the overview camera than to the audience-tracking camera. To encode this rule, we only need to make the transition probability of the former higher than that of the latter. At a microscopic level, each camera transition is random, resulting in interesting video editing effects. At a macroscopic level, some transitions are more likely to happen than others, obeying the video editing rules. Experimental results in Section 7 reveal that such an FSM strategy performs very well in simulating a human director's role.

7 EXPERIMENTS AND USER STUDY

In our CHI'01 paper, we have reported detailed user study results [13]. Here we will only concentrate on the highlights of the study. Our user study had two goals. First, we wanted to evaluate how much each individual video production rule affected the remote audience's viewing experience. Second, we wanted to compare the overall video quality of our automated system to that of a human operator. The human operator that we used in the study is our organization's regular camera operator, who has many years of experience in photo and video production.

7.1 Experiment Setup

Our system was deployed in one of our organization's lecture rooms. Originally, there were four cameras in the room, as shown in Figure 2. The camera operator used those four cameras to record regular lectures. The lectures are broadcast live to employees at their desktops and archived for on-demand viewing.

To make a fair comparison between our system and the human operator, we restructured the lecture room such that both the human operator and our system had four cameras: they shared the same static overview camera and slide projector camera, while both of them had separate lecturer-tracking cameras and separate audience-tracking cameras that were placed at close-by locations. They also used independent video mixers.

For user testing, two studies were conducted. The first study was a field study with our organization's employees while the second was a lab study with participants recruited from nearby colleges. For the field study, four lectures were used: three were regular technical lectures and the fourth was a general-topic lecture on skydiving held specifically for this study. This skydiving lecture was also used for the lab study.

For the field study, a total of 24 employees watched one of the four lectures live from their desktops in the same way they would have watched any other lectures. While providing a realistic test of the system, this study lacked a controlled environment: remote audience members might have watched the lecture while doing other tasks like reading e-mail or surfing the web. For a more controlled study, we conducted a lab study with eight college students who were not affiliated with our organization. College students were recruited because of their likelihood of watching lectures in their day-to-day life. The field study and the lab study are complementary to each other. By conducting both studies, we hope we can evaluate our system in a comprehensive way.

The interface used for both studies is shown in Figure 3. All four lectures for the study were captured simultaneously by the human operator and our system. When participants watched a lecture, the human operator captured version and our system captured version alternated in the MediaPlayer window (Figure 3). For the three 1.5-hour regular lectures, the two versions alternated every 15 minutes. For the half-hour skydiving lecture, the two versions alternated every 5 minutes. Which version was shown first was randomized. After watching the lecture, participants provided feedback using a survey. The highlights of the study are reported in the following section.

7.2 Study Results

We measured our system's performance by using questions based on individual video production rules collected from the professionals, as well as a few overall quality questions.

Table 1. Survey results for individual questions

(1 = strongly disagree, 5 = strongly agree)	Study session	Human operator			Our system		
		Mean	Median	St. dv.	Mean	Median	St. dv.
The operator followed the speaker smoothly	Field	3.19	3.00	0.83	2.65	2.50	0.88
	Lab	3.50	3.50	0.53	2.87	3.00	0.83
The operator zoomed and centered the camera appropriately	Field	3.11	3.00	0.88	2.67	3.00	1.02
	Lab	4.00	4.00	0.53	3.00	3.50	1.20
The operator did a good job of showing audience when they asked questions	Field	2.53	2.00	1.01	2.22	2.00	0.94
	Lab	3.25	3.50	0.89	2.87	3.00	0.83
The operator did a good job of showing audience reactions to the speaker	Field	2.83	3.00	0.71	2.55	3.00	0.69
	Lab	3.25	3.00	1.04	2.50	2.50	0.93

Table 2. Survey results for overall quality

(1 = strongly disagree, 5 = strongly agree)	Study session	Human operator			Our system		
		Mean	Median	St. dv.	Mean	Median	St. dv.
Overall, I liked the way this operator controlled the camera	Field	3.55	4.00	0.83	2.82	3.00	1.18
	Lab	4.00	4.00	0.53	3.00	2.50	1.31
The operator did a good job of showing me what I wanted to watch	Field	3.40	3.00	0.75	2.86	3.00	1.17
	Lab	4.00	4.00	0.53	2.88	2.50	1.13
I liked the frequency with which camera shots changed	Field	3.40	4.00	0.75	2.91	3.00	1.11
	Lab	3.50	3.50	1.20	2.75	2.00	1.39

The individual questions we asked and the survey results are summarized in Table 1. One issue with the individual questions is that it may be unreasonable to expect that audience members pay specific attention to individual video production rules. Thus, we also ask overall quality questions and the results are summarized in Table 2.

For both the individual rules and overall quality questions, we use the Wilcoxon test to compare the performance difference between our system and the human operator. Results from both tables show that there is a general trend that the human is rated slightly higher than the automated system. However, none of the differences are found to be statistically significant at the p=0.05 level, except for the question, “the operator did a good job of showing me what I wanted to watch” with the lab study subjects [13]. To push the comparison to an extreme, at the end of the survey we asked a simple Turing test: “do you think each camera operator is a human or computer?” The results are summarized in Table 3. The data clearly show that participants could not determine which system is the computer and which system is the human at any rate better than chance. For these particular lectures and participants, our system passed the Turing test.

8 CONCLUDING REMARKS

In this paper, we have reported the system design, audio-visual tracking techniques, camera management strategies, and highlights of user study results of a fully automated camera management system. We demonstrate that the quality of our system is getting close to that of a human-operated system.

As computer technology continues to advance and cost of equipment continues to drop, we envision in the near future automated camera management systems like this one will be deployed in many lecture halls and classrooms. Broadcasting and archiving a lecture will be as easy as turning on a light switch. To make all these happen, we are continuing to work on more robust tracking techniques, richer camera management strategies, and more modular design of the system so that it is easily customized to different users’ needs.

Table 3. Turing test results

Study session	Correct	Incorrect	No opinion
Field	17	16	15
Lab	7	7	2

9 ACKNOWLEDGEMENT

The authors would like to thank Jim Crawford and Dave Crosier for helping us deploy the system in the MSR lecture room; JJ Cadiz for helping the user study; Barry Brumitt for presenting the lab study lecture; Dinei Florencio for valuable discussions on microphone arrays; and John McGrath, John Conrad, Travis Petershagen, Greg Small and Jay Deffinbaugh for sharing their video production rules with us.

10 REFERENCES

1. Baumberg, A. & Hogg, D., An efficient method for contour tracking using active shape models, *TR 94.11*, Univ. of Leeds.
2. Benesty, J., Adaptive eigenvalue decomposition algorithm for passive acoustic source localization, *Journal of Acoustics of America*, vol. 107, January 2000, 384-391.
3. Bianchi, M., AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations, *Proc. of Joint DARPA/NIST Smart Spaces Technology Workshop*, July 1998.
4. Brandstein, M. and Silverman, H., "A Practical Methodology for Speech Source Localization with Microphone Arrays," *Computer, Speech, and Language*, 11(2):91-126, April 1997.
5. Brotherton, J. & Abowd, G., Rooms take note: room takes notes!, Proc. AAAI Symposim on Intelligent Environments, 1998, 23-30.
6. Buxton, W., Sellen, A., & Sheasby, M., Interfaces for multiparty videoconferences, *Video-mediated communication* (edited by Finn, K., Sellen, A., & Wilbur, S.), Lawrence Erlbaum Publishers.
7. Cruz, G. & Hill, R., Capturing and playing multimedia events with STREAMS, *Proc. ACM Multimedia’94*, 193-200.
8. Cutler, R. & Turk, M., View-based Interpretation of Real-time Optical Flow for Gesture Recognition, *IEEE Automatic Face and Gesture Recognition*, April 1998
9. Foote, J. and Kimber, D., FlyCam: Practical panoramic video, *Proc. of IEEE International Conference on Multimedia and Expo*, vol. III, pp. 1419-1422, 2000
10. Gleicher M., & Masanz, J., Towards virtual videography, *Proc. of ACM Multimedia’00*, LA, Nov. 2000
11. He, L., Cohen, M., & Salesin, D., The virtual cinematographer: a paradigm for automatic real-time camera control and directing, *Proc. of ACM SIGGRAPH’96*, New Orleans, LA. August 1996.
12. He, L., Grudin, J., & Gupta, A., Designing presentations for on-demand viewing, *Proc. of CSCW’00*, Dec. 2000
13. Liu, Q., Rui, Y., Gupta, A. and Cadiz, J.J., Automating camera management in lecture room environments, *Proc. of ACM CHI 2001*, Seattle, WA, March, 2001, http://www.research. microsoft.com /~yongrui/ps/chi01b.doc
14. Mukhopadhyay, S., & Smith, B., Passive Capture and Structuring of Lectures, *Proc. of ACM Multimedia’99*, Orlando.
15. ParkerVision, http://www.parkervision.com/
16. PictureTel, http://www.picturetel.com/
17. PolyCom, http://www.polycom.com/
18. Sound Professionals, http://www.soundprofessionals.com/ moreinfopages/cardioid/generalinfo.html
19. Stanford Online, http://stanford-onlines.stanford.edu/
20. Stiefelhagen, R., Yang, J., & Waibel, A., Modeling focus of attention for meeting indexing, *Proc. of ACM Multiemdia’99*.
21. Wang, C. & Brandstein, M., A hybrid real-time face tracking system, *Proc. of ICASSP98*, May 1998, Seattle, 3737-3740.
22. Wang, H. & Chu, P., Voice source localization for automatic camera pointing system in video conferencing, *ICASSP’97*
23. Zhang, Z., A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330-1334, 2000