A Portable Solution for Automatic Lecture Room Camera Management

Michael N. Wallick [†], Yong Rui[‡] and Liwei He [‡]

[†]University of Wisconsin-Madison
1210 West Dayton Street
Madison, WI 53706
michaelw@cs.wisc.edu

[‡] Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
{*yongrui, lhe*}*@microsoft.com*

Abstract

Rapid advances in technology and decreasing costs have made it possible to attach high resolution video cameras to just about any computer and record the interactions in a lecture room. Additionally, lecture rooms may be outfitted with several cameras for this purpose. However, recording the interactions alone does not create effective video. In this paper we present a method for not only recording, but also editing in real-time, lectures. Unlike previous work, the system is highly portable, allowing quick set-up in various types of lecture rooms. This portability is achieved by using the abstraction of virtual cameramen and physical cameras, and a scriptable interface to the editing rules.

Keywords
Automated camera management, Video production rules, Virtual video director, Speaker tracking, Portable video production.

1. Introduction

In previous work [4,5], we presented techniques for automating camera management in lecture settings. While the techniques produced reasonable results, there were still several drawbacks primarily in the portability of the system. For example, the original system was implemented and tested in a single lecture room, with tailored editing rules. Furthermore, the room is outfit with several analog video cameras, microphones and an analog audio-video mixer. This makes porting the system to other lecture rooms difficult and expensive, e.g., require re-writing and re-compilation of the system if moved to a room with different size and different number of cameras. The emergence of new technologies (both hardware and software) prompted us to revisit our Intelligent Camera Management (ICam) system to address its portability:

- Abstraction between virtual cameramen and physical cameras to allow portability in different lecture rooms settings (different room sizes and different audio-video equipment).
- A scripting language that can encode virtual director rules in a portable way.

In addition to the portability of the system, we have further improved speaker tracking virtual cameraman which now can handle multiple people to provide better user experience. The remainder of this paper is organized as follows. In Section 2 we will give a brief overview of some of the existing works involving automatic generation of lecture room video. Next, Section 3 describes the basic overview of how our new system works. In Section 4 we discuss the new virtual cameramen, with emphasis on our new speaker tracking. Section 5 lays out the scripting language we defined for describing camera behaviors. Section 6 shows how the system can be integrated into two different lecture room designs. We give concluding remarks in Section 7.

2. Related Work

In this section we present a brief overview of some of the systems that look at automatic camera management and video creation. This is by no means an exhaustive survey.

AutoAuditorium [1] is a commercial system, and one of the first examples in this domain. AutoAuditorium works by placing several hardware components in a lecture room. The speaker and PowerPoint slides are tracked during the lecture. The cameras will switch between the different views as the lecture progresses. This system is based entirely in hardware, expensive to install, and difficult to move to different locations once installed.

Gleicher et. al.'s [2,3] Virtual Videography system works by recording lectures with a small number of unmoving cameras. After the lecture is recorded, it is edited offline. While their system is portable, requiring only a few cameras in the classroom, the offline processing can be very time consuming, and does not allow for live broadcasting. Our system is online and real-time.

3. System Overview

In this section, we give an overview of the system and highlight the new design and components that make the system extensible and portable. Our automated system simulates a professional human filming crew, with one virtual director and multiple virtual cameramen. It is the responsibility of the director to pick which cameraman should be broadcasting at any given time. The director communicates with the cameramen by checking and setting states within the cameramen. If the director wants a cameraman to go on-air, it will first set the cameraman's state to "preview." This signals the cameraman to get ready to be broadcast. Once the cameraman is ready, it sets its state to "ready," and the director will see this and change the state to "on-air." When a camera is to stop

[†] Research performed while at Microsoft Research.

broadcasting, the director will change the state to "off-air" [5]. Figure 1 is a block diagram overview of the system

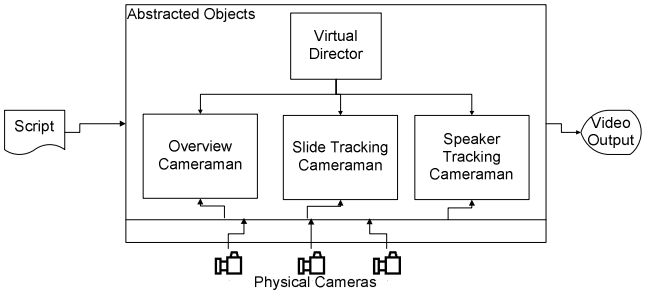

Figure 1: Overview of the system.

Note that there are two major parts in the system. One is inside the box and called the abstracted objects. It contains the virtual director, virtual cameramen, and communication mechanism between them. This abstracted object is common to all the different lecture room settings and is implemented using a flexible DMO, or DirectX Media Object. The other major part includes the specific directing scripts and specific physical cameras used in a specific lecture room. The script describes the states in the system state machine, and the transition rules that the director should follow. The separation between these two parts allows portability of the system.

The DMO may have any number of physical cameras connected to it. Further, DirectShow (the technology a DMO is based on) abstracts the physical camera type, meaning that any camera type, or combination of cameras, may be connected to system. For example, our system uses both Aplux 1.3 mega-pixel USB2.0 digital video cameras and Sony EVI-D30 Pan-Tilt-Zoom (PTZ) analog cameras (see Figure 2). It can easily handle other cameras as well.

Figure 2: (Left) Sony EVI-D30 PTZ Camera (the top port is a static monitoring camera). (Right) Aplux 1.3 mega-pixel USB2.0 digital video camera.

4. Improved Virtual Cameramen

The emergence of inexpensive and high resolution video cameras is a major factor in the portability of our system. Newer video cameras can be connected directly to a computer's existing ports, meaning that any computer may be outfit with high resolution video cameras.

Because the resolution of these video cameras is so high, it is possible for a single camera to mimic actions of several lower resolution cameras digitally. In fact, in a small room, a single physical camera suffices.

The cameras are processed internally by virtual cameramen. Currently we define 3 types of cameramen: overview, slide tracking camera, and speaker tracking camera. Each cameraman takes input from some physical camera and produces the correct output shot.

An overview camera captures the entire podium area. Slide tracking camera monitors a predefined area (the screen/slide area) for change and shows the slide image. Speaker tracking finds the speaker in the image. Each physical camera serves as input to one or more virtual cameramen. The cameraman process the video of the physical camera based on the cameraman type.

When queried by the director, each cameraman can return a confidence and a score. The confidence indicates how sure the cameraman can do the right action, and the score indicates how good that shot is. The overview camera is always confident; the slide tracking camera is confident when a new slide appears and the confidence decreases with time. The speaker tracking camera's confidence is based on how well the tracking algorithm localizes the speaker. A significant improvement over the previous system is the speaker tracking cameramen and we will next focus on it.

4.1 Speaker Tracking Cameraman

The speaker tracking cameraman attempts to locate the speaker in the video. It will output a subsection of the original video containing the speaker.

Tracking is handled in a separate thread of operation, and is updated every quarter to half a second. Whenever the DMO signals that there is a new frame, and enough time has elapsed, the tracker will update the location of the speaker using the following tracking algorithm. Let F_i and F_{i-1} be the current and last tracked frame, respectively. D is the difference between those two frames, and B is the binary difference. We have

1) $D = |F_i - F_{i-1}|$
2) $B = Threshold(D)$
3) Build a vertical projection P for each column in B.
   For every on pixel in a column of B, increment columns location in the projection by r.

where the vertical projection P represents the number of difference pixels in any given column, i.e. the horizontal location of the change or the speaker. The value for r is based on the height of any pixel. Pixels that are higher in the image have higher values for r. This gives the speakers head higher weights than lower parts of the body.

P is searched for the peak closest the last position of the speaker and the location is updated. The highest pixel in B, around the peak is assumed to be the speakers head location. Next, P is searched for additional peaks to indicate the presence of more than one person in the shot.

The found location of the speaker is further temporally filtered:

a) $x' = \alpha_x * x_t + (1- \alpha_x)*x_{t-1}$

b) $y' = \alpha_y * y_t + (1- \alpha_y)*y_{t-1}$

The camera keeps a small buffer of where the speaker has been in order to set an appropriate zoom level. If the speaker moves out of the frame, while on-air, the camera can slowly pan or make a transition cut to an overview shot then back to the speaker shot with the speaker reframed at the center. In the previous implementation, the tracker zooms out when there is more than one person in view. This problem is addressed using the histogram method described above. While there are more sophisticated tracking algorithms that may be employed, the method we describe works in real time and is accurate enough for our purposes.

In smaller rooms a single Aplux digital camera (1.3 mega-pixel) is appropriate for tracking. However, in larger rooms, a Sony PTZ camera is necessary. Because of the separation of virtual cameramen and physical cameras, our system easily handles both the Aplux digital camera and the Sony PTZ camera. In the latter case, the cameraman uses a static monitoring camera (top portion of Figure 2 (a)) that is collaborated to the Sony camera. The speaker is tracked using the static monitoring camera, and the Sony camera zooms in on the speaker (Figure 4).

Figure 3 shows an example of information from the tracking camera. There are several regions of interest in this camera view. The first is the tracking region, where the system looks in to locate the presenter. Second is the PowerPoint or presentation region, where slides are projected. Tracking is not performed inside the presentation region, since changing slides may confuse the tracker. Next is the location of the speaker. Finally is the estimated view of where the Sony camera is looking. The bottom of the image is the vertical projection of the speaker (P). A drawing of the setup for this room is shown in Figure 6.

Figure 3: Information provided by tracking cameraman.

5. Scripting Language

In our previous system the rules for transitioning between cameras are predefined and hard coded into the system. This greatly limits portability as the system has to be rewritten and recompiled for a new lecture room. In our new implementation we have defined a simple yet descriptive scripting language for specifying the cameras and the rules for the lecture room. Different rooms get different scripts.

Figure 4: Output from the Speaker Tracking cameraman.

In the script, the user specifies the virtual cameramen, states in the system, and the transitions. Each cameraman is given a name and a physical camera to associate with it. The states are associated with a virtual cameraman. The transitions refer to a state and a set of conditions. If a condition is met then the director will switch to another state specified by the transition.

Consider, for example a small conference room with a layout in Figure 5. In this room there is one physical camera connected to a PC. For visual effects, presenters use either the white board or the monitor in the front of the room. The following is an example of a script that may be used for this room:

```
1. CAMERAS
2. Overview OverviewCam 0
3. SlideCamera SlidesCam 0
4. SpeakerTracker TrackerCam 0
5. STATES
6. Overview OverviewCam
7. Slide SlideCam
8. Tracking TrackerCam
9. TRANSITIONS:
10. ALLSTATES
11. CONFIDENCE SlideCam 7 G
12. Slides 1.0
13. Slide
14. TIME 8 G
15. Tracking 2.0 Overview 10
16. Overview
17. TIME 8 G
18. Tracking 2.0 Slides 0.5
19. Tracking
20. CONFIDENCE TrackerCam 4 L
21. TIME 45 G
22. Overview 1.0 Slides 1.0
```

In the above example, lines 1, 5, 9 indicate that the cameras states and transitions are going to be specified respectively. Lines 2-4 describe the cameramen to be used (overview, slide tracking, and speaker tracking) all of which get their input from physical camera 0. Lines 6-8 define the states in the system; there is one state for each cameraman. Finally

lines 10-22 define the transitions in the system. The first transition (lines 10-12) specifies that if the slide tracking camera man has a confidence greater than 7 switch to the slide camera. Lines 13-15 and 16-18 specify that the slide camera and overview camera should be shown for 8 seconds, after which the system should transition to one of the other states, with a bias towards tracking. Lines 19-22 describe the rules to transition from the speaker tracker. This should happen whenever the speaker tracker has a confidence of less than 4, meaning the speaker has probably been lost, or the shot lasts longer than 45 seconds. From the tracker, it should go to either overview or slides with equal probability.

6. Usage Examples

The original implementation of the ICam system is only set up for a single lecture room. We have now built a system that is portable and can be easily implemented under several different circumstances. In this section we describe use of the system in two different rooms. The first is a small conference room and the second is a large auditorium.

Figure 5 is a simple layout of the small conference room. There is a white board, and a monitor in the front of the room. The center of the room has a conference table. In the back of the room is a workspace where a computer is set up with a single physical camera set up to record lectures in the room. The assumed use of this room is that participants will sit around the table, and the leader will stand in the front of the room using the white board and/or monitor to present information. The physical camera is an Alux 1.3 mega-pixel digital video camera operating at 10 frames per second. This camera represents three virtual cameramen: overview, speaker tracking, and slide tracking. An example script for this room was described in Section 5.

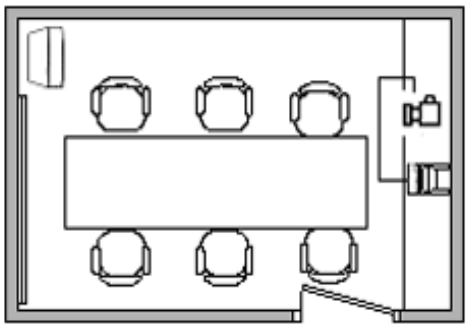

Figure 5: Drawing of small conference room

The second room that we consider is a large lecture room. Figure 6 is a simple layout of this room. The front of the room has a white board and screen for projected presentations. There are five physical cameras and 4 virtual cameramen. The cameras in this room are: two overview cameras (front of classroom and audience), one slide camera, one Sony PTZ camera, and one monitoring camera attached to the PTZ camera (see Figure 2 (a)). The monitoring camera is not used for broadcast, rather to drive the Sony PTZ (see Section 4). As with the smaller room, the script specifies that the slide camera should be used whenever a new slide appears. The speaker should be tracked as much as possible, and the two overview cameras and slide camera may be used to add variety to the video.

We conducted preliminary information study of the system by showing the output video to both professional videographiers and people who would normally watch such videos. Both groups were positive about results of the automated video.

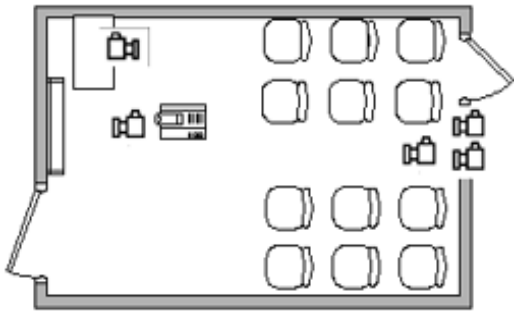

Figure 6: Drawing of large lecture room.

7. Conclusion

We have developed a portable automatic camera management system for lecture rooms. This system works in real time and any video camera can be used as input. The portability of the system comes from the separation between virtual cameramen and physical cameras and from the scripting language. By doing this, we have been able to run the same system in both a small conference room and a large lecture room under very different circumstances.

Successful lecture room automation systems will make a major impact on how people attend and learn from lectures. The cost of hardware for such systems is already reasonable and is continuously dropping. By eliminating the need to hire human videographers in some cases, a growing number of presentations can be made accessible online in universities and corporations.

8. References

1. Bianchi, M., AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations, *Proc. of Joint DARPA/NIST Smart Spaces Technology Workshop*, July 1998.
2. Gleicher, Michael; Masanz. Towards Virtual videography. ACM Multimedia 2000, Los Angeles, CA. November, 2000.
3. Gleicher, Michael; Heck, Rachel; Wallick, Michael. A Framework for Virtual Videograph. Smart Graphics. June 2002.
4. Liu, Q.; Rui, Y.; Gupta A.; and Cadiz, J., Automating Camera Management for Lecture Room Environments, *Proc. of SIGCHI '01*, Seattle, WA.
5. Rui, Yong; Gupta, Anoop; Grudin, Jonathan, Videography for Telepresentation. Proc. of ACM CHI 2003, Fort Lauderdale, FL. 2003.