An Automated End-to-End Lecture Capturing and Broadcasting System

Cha Zhang, Yong Rui, Jim Crawford and Li-wei He
Microsoft Research

September 2005

Technical Report
MSR-TR-2005-128

Contact author's address: Cha Zhang, Microsoft Research, One Microsoft Way, Redmond, WA 98052. Email: chazhang@microsoft.com.

Increasingly popular, lectures are given before a live audience, while simultaneously being viewed remotely and recorded for subsequent on-demand viewing over the Internet. Providing such services, however, is often prohibitive due to the cost of labor-intensive content capturing and pre-/post-production. This paper presents a complete end-to-end system that is fully automated and supports capturing, broadcasting, viewing, archiving and search. Specifically, we describe a system architecture that minimizes the pre- and post-production time, and a fully automated lecture capturing system called *iCam2*, which synchronously captures all the contents of the lecture, including audio, video and visual aids. As no staff is needed during the lecture capturing and broadcasting process, the operation cost of our system is negligible. The system has been used on a daily basis for more than 4 years, during which 522 lectures were captured with 20,000+ online viewers.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1. INTRODUCTION

Live/on-demand Internet broadcasting of lectures in workplace, conferences and educational settings has attracted more and more interest recently due to the improvements in network bandwidth, computer performance and compression technologies. Many corporations make seminars and training sessions available for employees who have temporal or spatial constraints on live attendance [He et al. 2001; Bianchi 1998]. Many conferences recorded their presentations, and made them available to conference participants [SGR ; NOSSDAV]. More and more universities also make lectures available online for both regular and distance education, such as University of California at Berkeley [BER], Stanford University [STA], and Columbia University [CVN]. The UK Open University was the first to webcast live and online degree ceremonies [Scott and Mason 2001], which received a lot of attention.

Although online viewing provides a convenient way for people to view lectures at more convenient time and location, the cost of providing such services can be prohibitive. Rowe et al. [Rowe et al. 2001] and Rui et al. [Rui et al. 2001] studied the cost issues in their respective papers, and summarized the cost into two major parts: 1) fixed cost, which includes computer servers, microphones and cameras; 2) recurring staffing cost, which includes pre-lecture activities (setting up the equipment), during-lecture activities (controlling cameras to track presenters and audience and switch between cameras), and post-lecture activities (post the lecture to a web site).

Fixed cost is mostly a one time investment, and it drops continuously during the past decade. Staffing cost, on the other hand, occurs every lecture and may not decrease. Recently there have been a lot of efforts in building automated/intelligent capturing systems to reduce the staffing cost [Liu and Kender 2004], e.g., the AutoAuditorium system by Bianchi [Bianchi 1998], the "Classroom 2000" project [Abowd 1999] at Georgia Tech, the Cornell Lecture Browser by Mukhopadhyay and Smith [Mukhopadhyay and Smith 1999], the Berkeley Internet Broadcasting System (BIBS) [Rowe et al. 2001] at UC Berkeley, the Microsoft iCam system [Rui et al. 2001; Rui et al. 2004] and the ePresence system [Baecker 2003] at University of Toronto. These systems have focused on different aspects of "automation". For example, some of them automated the camera control and tracking of presenters and audience [Bianchi 1998; Rui et al. 2001; Onishi and Fukunaga 2004; Gleicher and Masanz 2000]; others automated digital ink and electronic slides capture [Abowd 1999; Wang et al. 2003]; yet others focused on automated broadcasting [Rowe et al. 2001; Baecker 2003]. Nevertheless, they all met their own application goal by following a set of design principles derived from what is available and what is necessary in their applications.

This paper presents a complete end-to-end system that is fully automated and has been used on a daily basis in our organization for a few years. Compared with existing systems, our system distinguishes itself by meeting a few stringent requirements raised in practice. For instance, in our organization the users of the system are full time employees. High-quality live broadcasting becomes very important, because it may give the employees great benefit by allowing them to watch the lecture live online and multi-task at the same time. In addition, the employees expect no post-production delays after the talk, so that they can watch the lectures whenever they have spare time. On the speaker's side, many of our speakers are external. This imposes two restrictions on our system. First, it is often very difficult to get their slides before or after the talk, thus we need to develop a mechanism to grab the slides during the lecture. Second, since it is usually the first time for most external speakers to use our system, they have little time to learn and adapt to the system. The system needs to be truly passive, and poses no restrictions on the speakers' behavior (for example, asking the speaker to stay within the field of view of a fixed camera). As for the system administrators who run and maintain the system, we want to minimize their work by making the lecture capturing and broadcasting processes fully automated. This includes a carefully designed system architecture that requires no human interaction during the broadcasting, and techniques to track the speaker and the audience in the lecture room. Last but of equal importance, to popularize the system, we want the system to be generalizable so that it can be installed in conference rooms with different configurations.

With the above requirements in mind, at the architecture level, we have designed a system that has minimum pre- or post-production. As soon as the speaker walks into the lecture room and gets ready for his/her presentation, the system administrator may start the capturing and live broadcasting process with a single click of a button. Remote users can immediately see what is happening in the lecture room. The contents of the lecture, including audio, video and visual aids such as slides, are captured synchronously and at hight quality with our automated lecture capturing system called *iCam2*. *iCam2* is developed based on our previous *iCam* system [Rui et al. 2001], and focuses on generalizability while improving its overall performance. The system captures high quality audio and generates professional-looking edited videos by switching between multiple camera views according to cinematographic rules. Visual aids/slides are synchronously captured for easy browsing and re-synchronization during on-demand viewing. Similar to *iCam*, the speaker can walk around during the presentation, and the system will automatically track him/her and any audience member who asks questions. Shortly after the lecture ends (less than one minute), the recorded lecture will be available for on-demand viewing, where the remote users have the flexibility to flip through the slides and re-synchronize audio/video with slides or vise versa. The whole capturing and broadcasting system was deployed in June, 2001. By mid-August, 2005, it had captured 522 lectures with 20,383 online viewers.

In the rest of the paper, we first describe related work in Section 2. The design principles and system architecture are presented in Section 3 and 4. The *iCam2* system for automated content production is detailed in Section 5. Section 6 describes the implementation of our system. System usage statistics and conclusions are given in Section 7 and 8, respectively.

2. RELATED WORK

As the name implies, a lecture capturing and broadcasting system has two major aspects — capturing and broadcasting. An ideal system should be able to automate both. In this section, we will review existing approaches on automated content capturing and automated broadcasting separately, although many systems have made progress on both aspects.

2.1 Automated content capturing

The contents of a lecture usually refer to the audio and video of the speaker/audience, and any visual aids such as slides or transparencies that the speaker uses to facilitate the talk. It is also important to make sure that all these contents are well synchronized, so that they can be presented to the remote clients in an organized fashion.

2.1.1 *Audio.* It has been emphasized in the video conferencing literature that audio is the most important information contained in a talk [Finn et al. 1977]. A typical audio capturing system (such as the one used in the STREAMS system [Cruz and Hill 1994]) includes a wireless speaker microphone, and a set of audience microphones distributed in the lecture room. The signals from all the microphones are mixed using an analog mixer. Although some analog mixers have built-in echo cancellation and noise suppression, most of them require a human operator to adjust the gain for each channel during the lecture. For instance, when the speaker talks, the microphones capturing the audience should be muted to remove unnecessary noises. On the other hand, when a question is raised from the audience, the human operator needs to figure out the closest microphone and increase its volume. The AutoAuditorium system [Bianchi 1998] tackled this problem using a special "Mic Ducker" circuit that selectively muted the audience microphones, although that required additional hardware. Another constraint is that distributed microphones need to be wired to the mixer, which can be expensive to set up. This paper will describe a digital mixing solution which is more portable. It automates the gain control process with a speaker speech detection algorithm, and enhances the audience audio using beamforming with microphone array (Section 5.1).

2.1.2 *Visual aids.* Visual aids such as slides are the next important information during a talk. In the Classroom 2000 [Abowd 1999] project, speakers must load their presentations into the system before class and teach using electronic whiteboards. The Cornell Lecture Browser [Mukhopadhyay and Smith 1999] allowed the system administrator to synchronize the audio/video and slides after the presentation, given that the slides are available. Unfortunately, many speakers, in particular those visiting speakers, will not send their slides ahead of time; some work on their slides until the last minute; some are not willing to share their slides at all. The ePresence system [Baecker 2003] got around this problem by an operator-initiated trigger which grabbed a scan converted representation of the data projector's output. In our system, this process is automated by a slide change detection algorithm, as detailed later.

2.1.3 *Video.* While audio conveys the major contents of the talk, it is video that makes a talk engaging [Tang 1992]. A large amount of effort has been dedicated to the video content production in literature. To leverage the cost of hiring professional videographers, it has been a common practice to use multiple cameras to capture the scene, and (optionally) process the data afterward or on-the-fly. The STREAMS project [Cruz and Hill 1994], for instance, captured multiple streams of the lecture, and gave remote clients the flexibility to choose which stream to watch. A similar approach was being experimented in the BIBS system [Rowe et al. 2001].

While providing all the streams to the users gives them flexibility, it increases the burden on the server bandwidth, and sometimes it could be distracting to the remote client. Similar to what a professional crew would do, a few approaches have been proposed to process the streams and generate a single one. Generally such a process is guided by a set of rules, which is either suggested by professionals [Rui et al. 2001], or summarized by mining professionally shot videos [Matsuo et al. 2002]. The AutoAuditorium system [Bianchi 1998] was a commercial product that can automate the whole lecture capturing process and record all information to a tape. In [Machnicki and Rowe 2002], Machnicki and Lowe developed a system that automates tasks such as control of recording equipment, stream broadcasting, camera control, and content decisions such as which camera view to broadcast. These camera switching heuristics are mostly time-based. A question monitor service based on audio analysis was used to detect questions from the audience and switch the webcast to show the audience member asking the question. Our previous *iCam* system [Rui et al. 2004] used a finite state machine to model the switching process. It can easily incorporate camera positioning rules suggested by the professionals, and can be event related. The implementation, however, was not flexible as all the rules are predefined and hard coded in the system. In the *iCam2* system, as described in Section 5.3, we define a simple yet descriptive scripting language for specifying these rules. This allows the system to be portable across various room configurations.

When it comes down to each individual camera, many systems used either fixed cameras or cameras controlled by human operators [Cruz and Hill 1994; Rowe et al. 2001; Baecker 2003]. Fixed camera is inexpensive and convenient to set up, but it may have severe limitations. For instance, if we use a fixed camera to capture the speaker, depending on the field of view of the camera, we may either get a low resolution speaker shot, or lose the speaker from time to time when he/she walks out of the field of view. A human operator can certainly solve the problem, but at a much higher cost. Automated speaker tracking has been an active research topic recently, and was used in a few existing systems [Bianchi 1998; Mukhopadhyay and Smith 1999; Rui et al. 2001; Onishi and Fukunaga 2004; Gleicher and Masanz 2000]. In the *iCam2* system, we develop a hybrid speaker tracking algorithm which uses a single pan/tilt/zoom (PTZ) camera instead of the dual camera setup in our previous *iCam* system, without any compromise on the tracking performance (Section 5.3.1).

2.2 Automated live/on-demand broadcasting

Few systems support live and on-demand lecture broadcasting on a daily basis, among which the BIBS [Rowe et al. 2001] system at UC Berkeley and the ePresence [Baecker 2003] system at University of Toronto are the most representative.

The BIBS [Rowe et al. 2001] has recently been adopted as an integral part of the university's course delivery infrastructure, and webcasts over 15 classes each semester. BIBS was designed to run automatically with as few staff as possible and at low cost. In classroom recording scenario, it is reasonable to assume that the lecture slides are available before or after the lecture. BIBS used the same mechanism as the Cornell Lecture Browser [Mukhopadhyay and Smith 1999] to synchronize the slides with the audio/video content, which sometimes needs human input.

The ePresence system [Baecker 2003] has been adopted by a few institutes, and will be available as an open source download as of July 2005 [XPM]. ePresence was designed to fulfill a list of goals derived from literature and prototype implementations. It is scalable, interactive, and able to support presenters and engage remote audiences with rich media. Due to its ambitious goals, ePresence was not designed to be fully automated – a moderator was required to coordinate the interactivity between remote users and speaker.

The Anystream Apreso Classroom [AAP] is a commercial product that supports fully automated lecture capturing and web publishing. It has nice features such as scheduled start and stop of the capturing process, synchronized audio and visual aids, and automated web publishing. It does not provide live broadcasting, though.

In this paper, we present a live/on-demand broadcasting system that is fully automated. As detailed later, our system architecture was carefully designed, so that it can minimize the pre- and post- production time.

3. DESIGN PRINCIPLES

The general design principles of our system are derived from past experiences and grounded in results from the video communications literature such as [He et al. 2001]. The emphasis, however, is on automation. We want to minimize the amount of work performed by humans, such that the system can run by itself on a daily basis, with little or no operation cost. Not surprisingly, this design philosophy is shared by many other systems, such as the BIBS [Rowe et al. 2001] and Apreso Classroom [AAP].

While a few previous works have made long lists of design principles [He et al. 2001; Baecker 2003], in this section we highlight a few principles that we believe are critical to our system. These principles are closely related to the requirements we mentioned in Section 1, and we believe our system is the first that fulfills all these stringent requirements.

[P1] The system is passive. We would like the speaker to behave normally during the talk, thus we do not pose any restrictions to them. The only device the speaker needs to wear is a wireless clip-on microphone.

[P2] The system has no pre- and post-production. For instance, we do not require the speaker to give us their slides/transparencies for pre-processing. After the presentation, no post-production such as slide integration is needed. The lecture is immediately available for on-demand viewing.

[P3] The system captures synchronized high resolution visual aids. Such synchronization is done on-the-fly during the lecture. Both the live and the on-demand viewers can watch them synchronously with the audio/video stream of the lecture.

[P4] The system captures audio and video of the lecture automatically. The previous *iCam* system was presented in [Rui et al. 2004]. A new version, which we call *iCam2*, has been developed (Section 5) which greatly enhances its portability. In both generations, no videographer or moderator is needed during

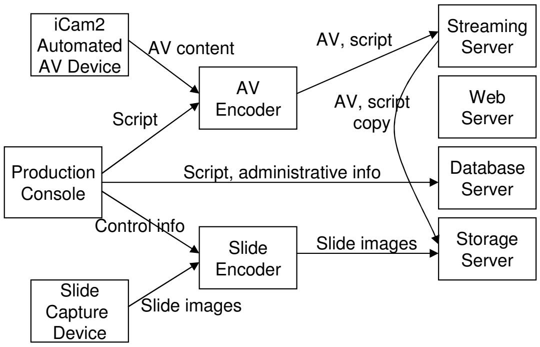


Fig. 1. System architecture of our system.

the lecture.

[P5] The system needs to be portable across various room configurations. We select network PTZ cameras to capture the video as they require minimum amount of wiring. A scripting language is developed to make the cinematographic rules adaptive to different number of cameras, different camera locations, and different room sizes.

[P6] The system allows the remote clients to view the lecture at their own pace. Metadata are sent along with the audio/video stream from the server. During on-demand viewing, the users can flip through the slides or jump around the audio/video stream. At any instance they can re-synchronize the audio/video with slides, or vise visa. No plug-ins for the Internet Explorer are needed to support such flexibility.

In the following text, we will annotate these design principles whenever applicable with the indexes [P1], [P2], etc.

4. SYSTEM ARCHITECTURE FOR AUTOMATED BROADCASTING

4.1 The capturing diagram

Figure 1 shows the architecture diagram of our system when capturing a lecture. The *iCam2* automated audio/video (AV) device and the slide capture device are the content providers, which will be discussed in detail in Section 5. The production console plays a central role, which coordinates the AV and slide capturing processes. When a slide change is detected, the production console sends commands to the slide encoder to trigger the capturing of slide. At the same time, it embeds a script command to the AV stream, which tells the remote client to update the slide image at that instance. Hence the AV stream and the slides are always synchronized on-the-fly [P3]. The AV encoder encodes both the AV stream and the script information into a single live stream, and sends it to the streaming server for live broadcasting. The slide encoder compresses slide images and sends them to a storage server, which is also accessed during live broadcasting.

To prepare for on-demand viewing, a copy of the AV/script stream is sent from the streaming server to the storage server during the capturing process [P2]. The production console also duplicates the script information to the database server.

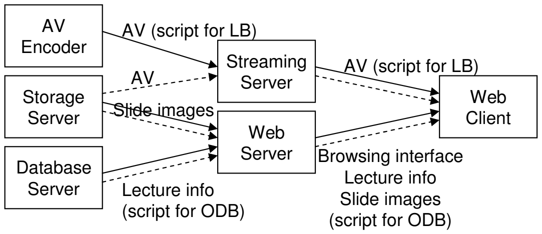


Fig. 2. Live and on-demand broadcasting diagram. The solid arrows are data flow for live broadcasting (LB), and the dash arrows are data flow for on-demand broadcasting(ODB).

This allows the remote client to watch the presentation at their own pace during on-demand viewing, because the scripts on the database server can be accessed at any order [P6]. In contrast, the scripts in the AV stream are sequentially embedded and un-indexed, which has limited random accessibility.

4.2 Live and on-demand broadcasting

Figure 2 shows the diagram for both live and on-demand broadcasting. Note the solid arrows are data flow for live broadcasting, and the dash arrows are data flow for on-demand broadcasting. In both cases, the web client only has direct access to the streaming server and the web server.

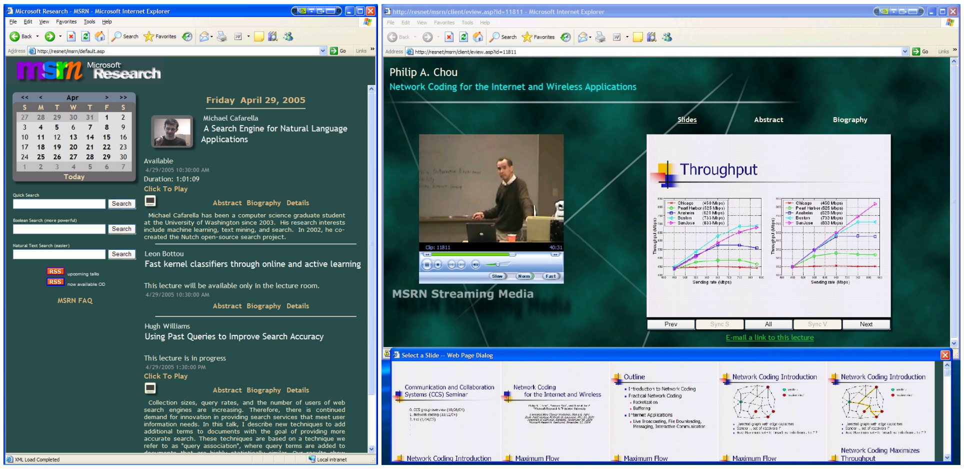
During live broadcasting, the streaming server provides the AV/script stream, and the web server provides the browsing interface, lecture information and slide images. Lecture information includes the speaker's biography and the abstract of the talk, which is entered into the database server before the lecture. The slide images, on the other hand, are captured during the lecture (Figure 1) and need to be fetched from the storage server.

The on-demand broadcasting data flow is slightly different from live broadcasting. First, the AV steam now originates from the storage server instead of the AV encoder. Second, script commands are retrieved from the database server. As mentioned in Section 4.1, this allows the remote clients to watch the talk at their own pace [P6].

4.3 The web interface

Figure 3 shows the web interfaces for lecture browsing on the remote client. Figure 3(a) is the index page. It has a calendar on the top left corner. If there are lectures available, the corresponding date appears in bold face. User can click on any date to retrieve detailed information of the talks available on that day, including the title, abstract, speaker biography, etc. For a recorded lecture, "Available" will appear under the title of the talk, and one can click the link to view the lecture on-demand. A live talk will show as "In progress". When the user clicks the link, it brings him/her to live viewing of the lecture. Text-based search capability is also provided in the index page (under the calendar). The user can use keywords to retrieve a talk he/she wants to listen to.

Figure 3(b) is the interface for watching the lecture. The video steam is displayed on the left, and the slides are on the right. If the lecture is live, the user can


Microsoft Research - MSRN - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://resnet/msrn/default.asp Go Links
msrn Microsoft Research
<< < Apr > >>
Today
Quick Search
Search
Boolean Search (more powerful)
Search
Natural Text Search (easier)
Search
RSS upcoming talks
RSS now available OD
MSRN FAQ
Friday April 29, 2005
Click To Play
Abstract Biography Details
XML Load Completed
Local intranet

Michael Cafarella
A Search Engine for Natural Language Applications
Available
4/29/2005 10:30:00 AM
Duration: 1:01:09

Michael Cafarella has been a computer science graduate student at the University of Washington since 2003. His research interests include machine learning, text mining, and search. In 2002, he co-created the Nutch open-source search project.

Leon Bottou
Fast kernel classifiers through online and active learning

This lecture will be available only in the lecture room.
4/29/2005 10:30:00 AM

Hugh Williams
Using Past Queries to Improve Search Accuracy

This lecture is in progress
4/29/2005 1:30:00 PM

Collection sizes, query rates, and the number of users of web search engines are increasing. Therefore, there is continued demand for innovation in providing search services that meet user information needs. In this talk, I describe new techniques to add additional terms to documents with the goal of providing more accurate search. These techniques are based on a technique we refer to as "query association", where query terms are added to

http://resnet/msrn/client/eview.asp?id=11811 - Microsoft Internet Explorer
Address http://resnet/msrn/client/eview.asp?id=11811
Philip A. Chou
Network Coding for the Internet and Wireless Applications
Slides Abstract Biography
Clip: 11811 40:31
Slow Norm Fast
MSRN Streaming Media
Throughput
Chicago (450 Mbps)
Pearl Harbor (525 Mbps)
Anaheim (625 Mbps)
Boston (733 Mbps)
SanJose (833 Mbps)
Throughput (Mbps)
Sending rate (Mbps)
Prev Sync S All Sync V Next
E-mail a link to this lecture
Select a Slide -- Web Page Dialog
Communication and Collaboration Systems (CCS) Seminar
Network Coding for the Internet and Wireless
Outline
Introduction to Network Coding
Practical Network Coding
Packetization
Buffering
Internet Applications
Live Broadcasting, File Downloading, Messaging, Interactive Communication
Network Coding Introduction
Network Coding Introduction
Network Coding Introduction
Maximum Flow
Maximum Flow
Maximum Flow
Network Coding Maximizes Throughput

(a) (b)

Fig. 3. Live viewing web interface. (a) The index page. (b) The lecture viewing interface.

watch the current video and slides synchronously, similar to watching a live event broadcast by a TV station. If the lecture was recorded before, the user will have more flexibility. He/she can browse all the slides used in the talk (bottom window of Figure 3(b)) by hitting the "All" button. Double clicking a thumbnail slide image will show the corresponding slide in the slide region. The user can also move the slider of the embedded Windows Media Player and jump to any place in the AV stream. The "Sync V" button allows audio/video to synchronize with slides, and the "Sync S" button synchronizes slides to audio/video [P6].

5. AUTOMATED LECTURE CAPTURING

In this section, we present our *iCam2* automated lecture capturing system [P4]. As shown in Figure 4, in a typical lecture room, we place a speaker camera at the back of the room for capturing the speaker, and a camera/microphone array combo on the podium for capturing the audience. Both cameras are Sony SNC-RZ30 pan/tilt/zoom (PTZ) network cameras. These cameras can be connected to any networked computer, which greatly enhances the system portability [P5]. The microphone array is an 8-element array. It serves two purposes — capturing the speech from the audience and guiding the audience camera to point to the right direction. As mentioned before, the *iCam2* system focuses on the portability issues while improving its overall system performance.

5.1 Audio

In the previous *iCam* system [Rui et al. 2004], the lecture audio is captured with two microphones. A wireless clip-on microphone captures the speaker [P1]. Another microphone sits on the podium and points to the audience in order to capture questions raised from the audience. The two signals as well as the audio signal

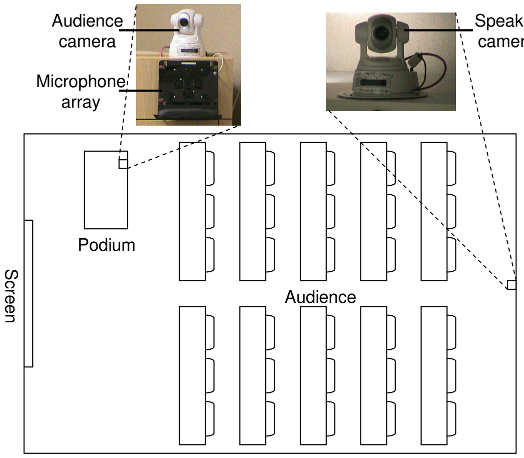


Fig. 4. A typical configuration of the *iCam2* system.

from the speaker's laptop are mixed through an analog audio mixer. A microphone array was installed on the podium, but its only purpose was to guide the audience camera. The system worked, but it could not capture the audiences' questions very well. In particular it had difficulty capturing questions raised by audiences in the back of the room, because the microphone on the podium had a limited range.

To address this problem, in *iCam2* we decide to use the microphone array to capture the audiences' speech. The diagram of the audio capturing process in the new system is shown in Figure 5. The microphone array has 8 microphones and its geometry is accurately measured. By using the fact that different microphones will perceive different phases in the signals from the same source position, we can identify the direction of the sound source [Rui and Florencio 2004]. Given that direction, the input signal from the 8 microphones are fused to form a beam pointing to the same direction, which gives much better sound quality [Tashev and Malvar 2005].

The processed audience speech and the speaker speech/laptop audio are then digitally mixed. Unfortunately, adding the two signals directly can cause problem: when the speaker talks or the laptop plays some audio file, not only the sound capture card captures the signal, the microphone array also captures a lot of room reverberation, which makes the mixed audio noisy. We therefore design a weighted mixing scheme to mute the signal from the microphone array whenever a speech or audio signal from the sound capture card is detected, as illustrated in Figure 5. The multiplication factor W in Figure 5 increases to 1.0 when no speech is detected at the sound capture card, and reduces to 0.0 otherwise. Note the microphone array signal needs to be muted fast (within 200 ms) so that the echo can be inaudible. On the other hand, the rate to increase W should be slow, because doing it fast may cause the weight to oscillate for speakers who like to pause between sentences. Such oscillation can cause unpleasant artifacts.

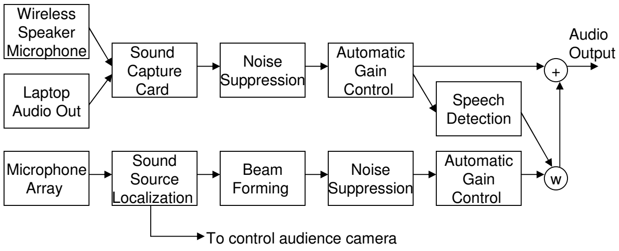


Fig. 5. The audio capturing process.

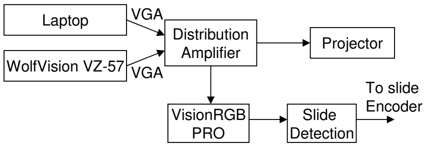


Fig. 6. Flow chart about the visual aids capturing. Please refer to Figure 1 for the slide encoder.

5.2 Visual aids

Since we do not acquire the original visual aids from the speaker before the lecture, we capture them on-the-fly [P2]. Figure 6 shows the diagram for such purpose. The design is modular, such that it can capture both slides and transparencies/papers. Transparencies/papers are scanned by a WolfVision VZ-57 Visualizer, which has VGA output. A distribution amplifier selects one of the inputs from the visualizer or a laptop, and sends it to the projector.

Meanwhile, a copy of the display is digitized by a video capture card [P3]. Currently images are grabbed at 1 fps, although 30 fps is possible. We then run a slide change detection algorithm over two subsequent frames. If the difference between two frames are greater than a threshold, we consider it a new slide, and send the production console a message to store this slide as well as informing the AV Encoder to embed a slide change script command. The slide detection and the production console resides on the same machine, thus there is no difficulty communicating between one another. Note that the slide detection algorithm does not have to have 100% accuracy. A slightly over sensitive detection algorithm will do the job well as it will not affect much of the viewing experience. For the same reason, even if the speaker goes back and forth in the slides, or adds some digital marks on the slides during the lecture, the stored images faithfully record what appears on the display in synchronic order, which is sufficient for archiving the lecture.

5.3 Video

Video makes lecture viewing more engaging. In *iCam2*, we replace the analog cameras used in the previous *iCam* system with two network cameras — a speaker camera and an audience camera [P5]. A diagram of the video capturing process is shown in Figure 7. The speaker camera will generate overview of the lecture room, speaker view, and screen view. The audience camera will produce overview

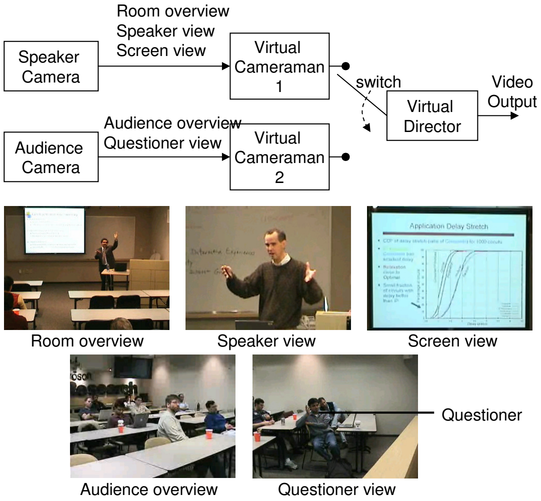


Fig. 7. The video capturing process.

of the audience, and questioner view pointing directly to the audience who asks a question. Each camera feeds its video to a virtual cameraman (VC). These VCs send their videos to a central virtual director (VD), which selects one of the inputs as the output video. The video processing components, including the VCs and the VD, are wrapped in a video source Direct Show filter [DSHOW], which can be used as a regular video device.

Other than the hardware upgrade, in *iCam2*, a hybrid tracking algorithm is developed to track the speaker with a single PTZ camera. The cinematography rules are now editable using a scripting language, which makes it possible to port our system to various room configurations.

5.3.1 *Speaker capturing.* In the previous *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. However, there is a research challenge here. In order to give a high resolution view of the speaker, the 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 [Zhang et al. 2005]. 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. This digital tracking approach is similar to Yokoi and Fujiyoshi's previous work in [Yokoi and Fujiyoshi 2004], however since our camera is a PTZ camera, we can also track the speaker mechanically. Digital tracking has the advantage of being smooth, and mechanical tracking can cover a wide area even when the camera is zoomed in (thus a high resolution shot for the speaker). The hybrid tracking achieves the benefits of both worlds.

To improve the aestheticity of the lecture scene, we further develop an intelligent pan/zoom selection scheme according to the activity of the speaker. The zoom level of the camera is controlled by the amount of motion detected throughout the tracking. If a speaker stands still most of the time, the camera will zoom in to obtain a close shot of the speaker. On the other hand, if the speaker moves around a lot, it is better to maintain at a low zoom level, such that the virtual camera will not be panning too often. The same speaker tracking camera is also used to mimic a screen capture camera, which is used when the speaker walks into the screen, or when the speaker shows some animation on the screen.

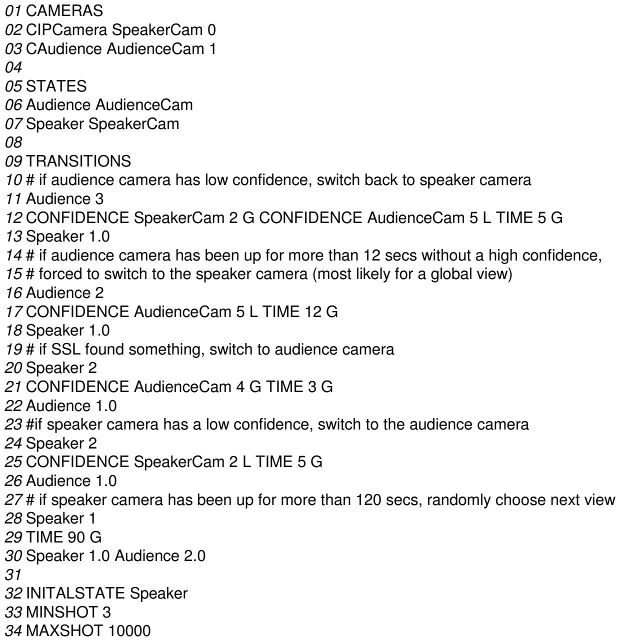
Overall, the quality of speaker tracking of *icam2* is similar to that of the previous *iCam* system, although only a single camera is used. The automatic zoom level control and the mimicked screen camera are newly introduced in *iCam2*, and the latter feature received much applause from the users.

5.3.2 *Audience capturing.* For a lecture capturing system, it is important to give shots of the local audience from time to time, because they are also part of the lecture. According to the cinematography rules [Rui et al. 2004], if the speaker has been on air for a while, the system should switch to the audience view with a certain probability. Five audience overview shot modes are currently implemented in the system: panning from left to right, panning from right to left, and three static views toward left, center and right. They are chosen randomly when an audience view is requested.

When there is a question raised from the audience, it is suggested by professional videographers that the system should give a shot to the person who asks the question [Rui et al. 2004]. This is through the help of a sound source localization (SSL) algorithm using the microphone array. In the previous *iCam* system, there are two microphones in the microphone array. In *iCam2*, we increase the number of microphones to 8, and use multi-microphone SSL to achieve more robust results [Rui and Florencio 2004].

5.3.3 *Scripting for cinematography rules.* The virtual director (VD) selects inputs from multiple virtual cameramen (VCs) according to cinematography rules. As mentioned before, in the previous *iCam* system, the cinematography rules are predefined and hard coded into the system. In *iCam2*, we develop a simple scripting language to define these rules, making it much flexible to tune and adjust to various room sizes, number of cameras and camera locations [P5].

Figure 8 shows a simplified script demonstrating the grammar. The line numbers in the front of each row are for demonstration purpose only. Line 1, 5, 9 indicate that the cameras states and transitions are going to be specified respectively. Lines 2–3 describe the cameramen to be used (speaker and audience), and they come from two different physical cameras indexed as 0 and 1. Lines 6–7 define the states in the system; there is one state for each cameraman. Lines 10–30 define the transitions in the system. Finally lines 32, 33 and 34 define the initial state of the system and



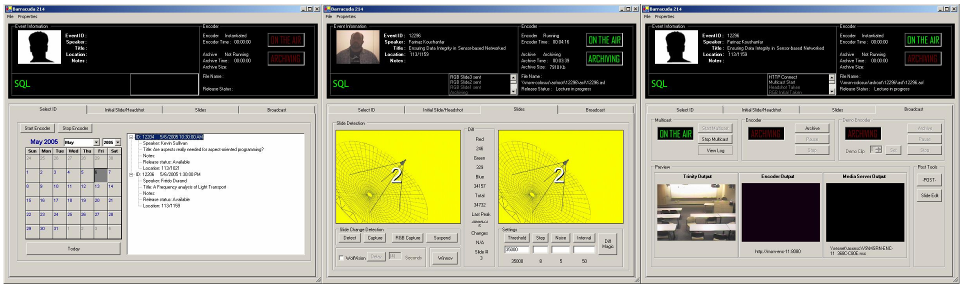
```
01 CAMERAS
02 CIPCamera SpeakerCam 0
03 CAudience AudienceCam 1
04
05 STATES
06 Audience AudienceCam
07 Speaker SpeakerCam
08
09 TRANSITIONS
10 # if audience camera has low confidence, switch back to speaker camera
11 Audience 3
12 CONFIDENCE SpeakerCam 2 G CONFIDENCE AudienceCam 5 L TIME 5 G
13 Speaker 1.0
14 # if audience camera has been up for more than 12 secs without a high confidence,
15 # forced to switch to the speaker camera (most likely for a global view)
16 Audience 2
17 CONFIDENCE AudienceCam 5 L TIME 12 G
18 Speaker 1.0
19 # if SSL found something, switch to audience camera
20 Speaker 2
21 CONFIDENCE AudienceCam 4 G TIME 3 G
22 Audience 1.0
23 #if speaker camera has a low confidence, switch to the audience camera
24 Speaker 2
25 CONFIDENCE SpeakerCam 2 L TIME 5 G
26 Audience 1.0
27 # if speaker camera has been up for more than 120 secs, randomly choose next view
28 Speaker 1
29 TIME 90 G
30 Speaker 1.0 Audience 2.0
31
32 INITALSTATE Speaker
33 MINSHOT 3
34 MAXSHOT 10000
```

Fig. 8. A simplified scripting for the virtual director.

the minimum and maximum duration of a shot.

Let us have a detailed look at the transitions. Note line 10, 14–15, 19, 23 and 27 are comments led by "#". Line 11 indicates that this rule applies when the audience camera is on air and when the next 3 conditions are met. In line 12, the conditions are specified: the confidence of the speaker camera shall be greater than 2; the confidence of the audience camera shall be less than 5; and the audience camera has been on air for more than 5 seconds. The first condition implies that the speaker camera has successfully tracked the speaker with certain confidence (range from 0 to 10). The second condition states that the audience camera does not have a high confidence. Since the audience camera's confidence comes from the SSL algorithm, it means SSL does not find an audio source in the audience. The third condition simply says that the audience view has been on air for a while. Once all three conditions are met, line 13 specifies the transition probability: the system will switch to the speaker view with chance 1.0. If multiple transition targets are specified, such as line 30, the chances should be proportional to transition probability. Once every second, the system will review these rules from top to bottom. If a certain rule is met, transition will take place with the specified probabilities.

The scripting language described above are extensions to the work in [Wallick et al. 2004]. Comparing with the virtual director designed in [Machnicki and Rowe 2002], the above method allows us to incorporate events in the transition using the "CONFIDENCE" keyword, which is much more powerful.



Event ID :
Speaker :
Title :
Location :
Notes :

Encoder Instantiated
Encoder Time : 00:00:00
Archive Not Running
Archive Time : 00:00:00
Archive Size:
File Name :
Release Status :

May 2005 May 2005

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

ID: 12204 5/6/2005 10:30:00 AM
Speaker: Kevin Sullivan
Title: Are aspects really needed for aspect-oriented programming?
Notes:
Release status: Available
Location: 113/1021
ID: 12206 5/6/2005 1:30:00 PM
Speaker: Frédo Durand
Title: A Frequency analysis of Light Transport
Notes:
Release status: Available
Location: 113/1159

(a)

Event ID : 12296
Speaker : Farinaz Koushanfar
Title : Ensuring Data Integrity in Sensor-based Networked
Location : 113/1159
Notes :

RGB Slide3 sent
RGB Slide2 sent
RGB Slide1 sent
Archiving

Encoder Running
Encoder Time : 00:04:16
Archive Archiving
Archive Time : 00:03:39
Archive Size: 7910 Kb
File Name : \\msrn-colosus\asfroot\12296\asf\12296.asf
Release Status : Lecture in progress

Red 246
Green 329
Blue 34157
Total 34732
Last Peak 3066423 6
Changes N/A
Slide # 3

Delay (4) Seconds

Threshold	Step	Noise	Interval
35000			
35000	8	5	50

(b)

Event ID : 12296
Speaker : Farinaz Koushanfar
Title : Ensuring Data Integrity in Sensor-based Networked
Location : 113/1159
Notes :

HTTP Connect
Multicast Start
Headshot Taken
RGB Initial Taken

Encoder Instantiated
Encoder Time : 00:00:00
Archive Not Running
Archive Time : 00:00:00
Archive Size:
File Name : \\msrn-colosus\asfroot\12296\asf\12296.asf
Release Status : Lecture in progress

Demo Clip 0

http://msrn-enc-11:8080

\\resnet\asxnsc\V9\MSRN-ENC-11 368C-C80E.nsc

(c)

Fig. 9. Management console for the automated lecture capturing system.

6. SYSTEM IMPLEMENTATION

Having discussed the system architecture and the automated lecture capturing, let us next have a look at the implementation details.

Refer to Figure 1, the *iCam2* automated AV device is implemented as two Direct Show [DSHOW] source filters (audio and video). From other applications' point of view, they are just like regular audio capture devices and video capture devices. This modular design greatly enhances the flexibility when integrating them to the broadcasting system [P5]. The slide encoder and the production console resides on the same computer. The slide encoder is a simple JPEG compressor. The production console is an applications that helps manage the whole system. It can be used to select and start/stop encoding of a lecture (Figure 9(a)), to monitor the slide change detection process (Figure 9(b)), and to start/stop/monitor the broadcasting process (Figure 9(c)).

Audio/video encoding is performed with Windows Media Encoder 9 series. Script command information from the production console is embedded into the stream during the live scenario in order to efficiently push the slide change events to the user. The encoder is instantiated and controlled from the production console via Distributed COM (DCOM) [DCOM]. This allows us to compartmentalize the encoding process and use the best hardware configuration for encoding and steaming the lecture audio/video. Multiple machine/encoder/profiles can be instantiated depending on what is required. This allows live streaming for multiple bandwidth targets.

The streaming server runs a Windows Server 2003/Enterprise Edition, which provides multicast capability. In the current implementation, the AV encoder and the streaming server are on the same machine. The live broadcast is one of the consumers of the live encoded stream.

We use Internet Information Services (IIS) [IIS] as our web server. The web application serves up html, xml and images to Internet Explorer (IE) on the viewer's machine. Some processing is offloaded to IE which integrates the html and xml to render the appropriate page. For security purpose, Windows NT challenge/response (NTLM) [NTLM] provides authentication, while various rules and mechanisms including Active Directory Services [ADSI] provide authorization.

The database server runs SQL Server. Currently data persisted in SQL Server falls into three general categories:
1. **Context information**, such as speaker name, talk title, abstract, speaker biography. This information is often written before the lecture.
2. **System data**, such as where archive is saved, current status of the talk (in progress/available/pending), script commands. Most data are created right before the lecture or during the lecture.
3. **Usage statistics**, such as the remote clients' domain, IP address, start time, lecture ID. This is collected for both live viewing and on-demand viewing.

During a live viewing session, when a user joins a lecture that is already in progress, SQL Server provides the last captured slide as the first slide rendered when the user joins. There is no waiting for the next slide in order to start. The later slide changes and other script commands are events raised by the Windows Media Player.

In the on-demand scenario, although the script commands exist in the stream, they are vestigial in nature. Instead, time driven events such as slide changes are driven by a separate client side component that has access to the Media Player's current time index, the script commands and the slide display user interface (UI) (Figure 3(b)). The component receives relevant events raised by the Windows Media Player that indicate a possible change of client state. The component also receives events from the slide display UI, that can direct it to have the Windows Media Player seek to a new position based on a selected slide [P6].

The storage server is a computer with RAID disk array that has 3TB of storage capacity. Each recorded lecture occupies about 100-200 MB, and all the recorded 500- lectures took less than 100 GB space.

Since the presentation is archived as it is broadcast, the contents can be available for on-demand viewing in a very short period of time, typically less than a minute. This is mainly due to encoding latency and the 15 to 45 seconds required by the Windows Media Encoder to build indexes in the video/audio stream [P2].

The system also supports streamlined off-line CD image production. The resulting CD has the same look and feel as the online client without the network dependency. These CDs are popular with the speakers themselves and are often a motivating factor in their agreement to be recorded, and or agreement to wider legal release.

7. SYSTEM USAGE

The automated lecture capturing and broadcasting system was first deployed on June 14, 2001. It had captured 522 lectures by mid-August, 2005, which is about 10 lectures per month. A total of 20,383 sessions are viewed online, among which 11,115 sessions are live sessions, and 9,268 sessions are on-demand sessions. This is in contrast to the results reported in the BIBS system at UC Berkeley [Rowe et al. 2001], where live plays contributed only 5–14%. The BIBS was not designed to replace attendance at live lectures, and the students primarily use the system for on-demand replay when studying for exams. In corporation environment, live broadcasting is much more wanted because people can multi-task in their own office while still be synchronized with the lecture progress, and save the time traveling to

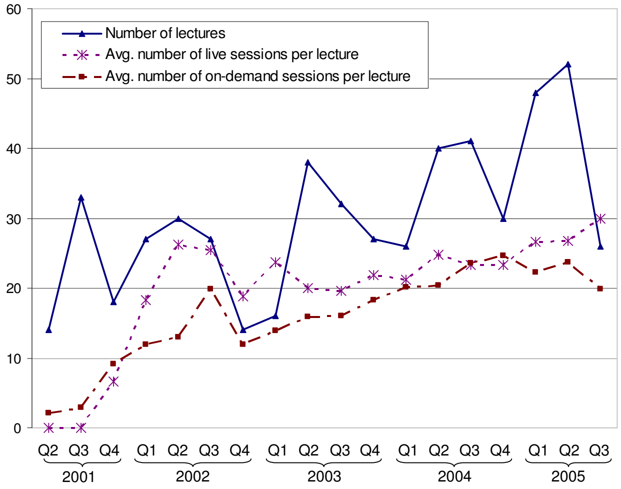


Fig. 10. Lecture and viewing statistics.

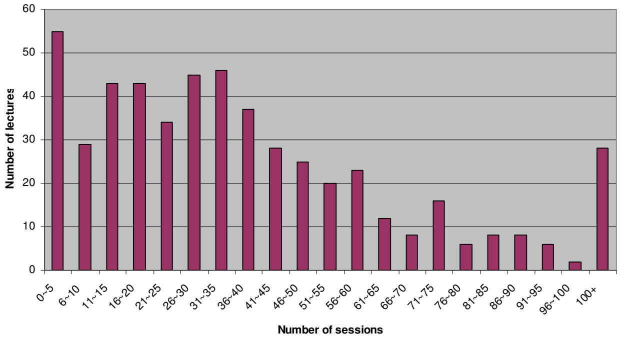


Fig. 11. Distribution of lectures w.r.t. the number of viewing sessions.

the lecture room.

Figure 10 shows the number of lectures broadcast per quarter, the average number of live sessions per lecture per quarter, and the average number of on-demand sessions per lecture per quarter. It can be seen that the number of live sessions is relatively stable – between 20 and 30 per lecture. This is because the talks given in the lecture room are mostly research talks in a specialized area and not targeted to the general public. However, the number is significant considering the local audience in the lecture room is often less than 20.

Figure 11 shows the distribution of lectures with respect to the number of viewing sessions (both live and on-demand). For instance, there are 55 lectures that have only $0 \sim 5$ viewing sessions, while there are also 28 lectures that have more than 100 viewing sessions. These statistics can tell the system administrator which are worth keeping. Lectures rarely watched can be deleted first in case of storage shortage.

Figure 12 shows the time-of-the-day distribution for people watching on-demand

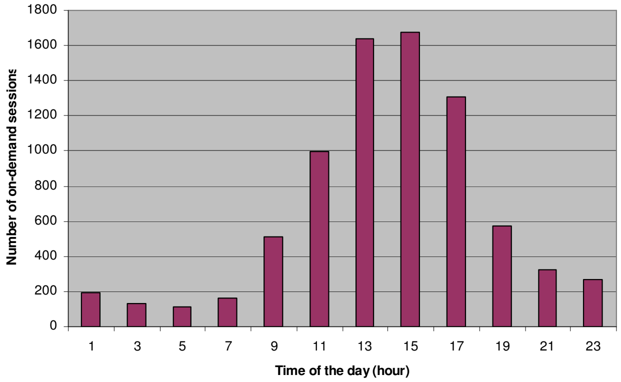


Fig. 12. Time-of-the-day distribution for on-demand sessions.

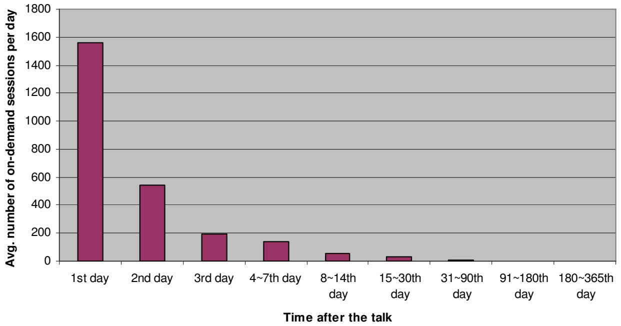


Fig. 13. When do on-demand sessions happen after the lecture.

lectures. It appears that most people like to watch them in the afternoon. Since people are often less productive in the afternoon, online lecture browsing provides them an alternative way to work and partially relieve themselves from tedious programing.

Figure 13 shows how soon people will watch an on-demand lecture after the actual presentation. The statistics are collected on the 345 lectures captured before August 2004. The vertical axis is the average number of on-demand sessions per day in a certain time period. The trend is very obvious: lectures are watched less and less often when time passes. Around 54.8% of the sessions happen within 1 week after the talk, 72.3% of them are within 1 month, and 80.2% of them are within 3 months. On average, only less than 1.6 viewing sessions were made per day for talks over half a year old. After one year, 152 out of the 345 lectures (44.1%) were never visited. This reveals the time aspect of system maintenance: talks that are old and never visited for a while can be deleted if necessary.

8. CONCLUSIONS

We have presented a fully automated end-to-end lecture capturing and broadcasting system. The contributions of this work are two-fold. First, we proposed a system architecture that minimizes the pre- and post-production time. It allows remote clients to view lectures both live and on-demand. Second, we reported our most recent progress on automated speaker/audience capturing, namely the *iCam2* system. Great effort has been made to enhance the system's portability while improving the overall performance. The result is a system that automated both lecture capturing and lecture broadcasting, making the operation cost of the system negligible.

There are still many interesting topics that remain to be explored. For instance, detecting the speaker's head orientation and gesture reliably can help the virtual cameraman to frame the speaker better. More powerful indexing mechanisms such as audio indexing [Yu and Seide 2005] could provide a better experience for the user to browse the lecture archives.

A complete end-to-end lecture capturing and broadcasting systems will make a big impact on how people attend and learn from lectures. We envision that capturing and broadcasting technologies will continue to advance, and making a presentation available online will be as easy as turning on a light switch. When we combine the technologies on capturing and broadcasting with those on browsing and annotating, we see new models for scaling-up our education systems.

REFERENCES

AAP. AnyStream Apreso, http://www.apreso.com/.

Abowd, G. 1999. Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Systems Journal 38,* 4, 508–530.

ADSI. http://msdn.microsoft.com/library/en-us/adsi/adsi/about_adsi.asp.

Baecker, R. 2003. A principled design for scalable internet visual communications with rich media, interactivity and structured archives. *Proc. Centre for Advanced Studies on Collaborative Research*.

BER. UC Berkeley Online Learning, http://learn.berkeley.edu.

Bianchi, M. 1998. Autoauditorium: a fully automatic, multi-camera system to televise auditorium presentations. *Proc. Joint DARPA/NIST Smart Spaces Technology Workshop*.

Cruz, G. and Hill, R. 1994. Capturing and playing multimedia events with streams. *Proc. ACM Multimedia*.

CVN. Columbia Video Network, http://www.cvn.columbia.edu.

DCOM. http://msdn.microsoft.com/library/en-us/dnanchor/html/dcom.asp.

DSHOW. http://msdn.microsoft.com/library/en-us/directshow/htm/directshowfilters.asp.

Finn, K., Sellen, A. J., and Wilbur, S. 1977. *Video-Mediated Communication.* Lawrence Erlbaum.

Gleicher, M. and Masanz, J. 2000. Towards virtual videography. *Proc. ACM Multimedia*.

He, L., Grudin, J., and Gupta, A. 2001. Designing presentations for on-demand viewing. *Proc. ACM CSCW*.

IIS. http://www.microsoft.com/windowsserver2003/iis/ default.mspx.

Liu, T. and Kender, J. R. 2004. Lecture videos for e-learning: Current research and challenges. *Proceedings of IEEE International Workshop on Multimedia Content-based Analysis and Retrieval*.

Machnicki, E. and Rowe, L. A. 2002. Virtual director: automating a webcast. *Proc. SPIE Multimedia Computing and Networking*.

Matsuo, Y., Amano, M., and Uehara, K. 2002. Mining video editing rules in video streams. *Proc. ACM Multimedia*.

Mukhopadhyay, S. and Smith, B. 1999. Passive capture and structuring of lectures. *Proc. ACM Multimedia*.

NOSSDAV. NOSSDAV 2005, http://bmrc.berkeley.edu/research/nossdav05/.

NTLM. http://msdn.microsoft.com/library/en-us/secauthn/security/microsoft_ntlm.asp.

Onishi, M. and Fukunaga, K. 2004. Shooting the lecture scene using computer controlled cameras based on situation understanding and evaluation of video images. *Proc. ICPR*.

Rowe, L. A., Pletcher, P., Harley, D., and Lawrence, S. 2001. Bibs: a lecture webcasting system. *Technical report, Berkeley Multimedia Research Center, U.C. Berkeley*.

Rui, Y. and Florencio, D. 2004. Time delay estimation in the presence of correlated noise and reverberation. *Proc. IEEE ICASSP*.

Rui, Y., Gupta, A., Grudin, J., and He, L. 2004. Automating lecture capture and broadcast: technology and videography. *ACM Multimedia Systems Journal 10,* 1, 3–15.

Rui, Y., He, L., Gupta, A., and Liu, Q. 2001. Building an intelligent camera management system. *Proc. of ACM Multimedia*.

Scott, P. and Mason, R. 2001. Graduating live and on-line: the multimedia webcast of the open university's worldwide virtual degree ceremony. *Proc. Webnet*.

SGR. SIGGRAPH Online, http://terra.cs.nps.navy.mil/distanceeducation/ online.siggraph.org/.

STA. Stanford Online, http://scpd.stanford.edu/scpd/students/onlineclass.htm.

Tang, J. C. 1992. Why do users like video? studies of multimedia-supported collaboration. *Sun Microsystems Lab Technical Report, TR-92-5*.

Tashev, I. and Malvar, H. 2005. A new beamforming design algorithm for microphone arrays. *Proc. ICASSP*.

Wallick, M., Rui, Y., and He, L. 2004. A portable solution for automatic lecture room camera management. *Proc. ICME*.

Wang, F., Ngo, C. W., and Pong, T. C. 2003. Synchronization of lecture videos and electronic slides by video text analysis. *Proc. ACM Multimedia*.

XPM. Xpresence Media, http://www.xpresence.com/.

Yokoi, T. and Fujiyoshi, H. 2004. Virtual camerawork for generating lecture video from high resolution images. *Proc. ICME*.

Yu, P. and Seide, F. 2005. Fast two-stage vocabulary-independent search in spontaneous search. *Proc. ICASSP*.

Zhang, C., Rui, Y., He, L., and Wallick, M. 2005. Hybrid speaker tracking in an automated lecture room. *Proc. ICME*.

Manuscript received in Sep. 2005.