

Capturing Conference Presentations

Lawrence A. Rowe
University of
California,
Berkeley

Vince Casalaina
Image Integration

Many organizations have developed technology to capture and stream presentations.¹⁻³ Yet, presentation capture is impractical at many professional meetings and conferences because of high costs. For example, the typical expense of capturing and publishing presentations using conventional technology is \$5,000 to \$20,000 per day, depending on the capture complexity and how you produce the final product.

We've developed an approach that's similar to the live-to-videotape recording process the broadcast industry uses, except we record compressed material onto a computer disk. Captured media files can be published immediately without offline editing or postproduction, significantly reducing publication cost. We tested our new approach by capturing presentations at the Association for Computing Machinery (ACM) Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV) 2005.⁴

The total cost of the equipment we used in our experiment (including audio, video, and computer equipment) was approximately \$12,000. The production team included one production assistant and one person who acted as both webcast

producer and director. Based on this experiment, we estimate it's possible to capture and publish conferences for approximately \$3,000 per day plus expenses (such as travel, room, and board). This estimate includes equipment rental.

This article describes the technology and process we used to capture and publish the NOSSDAV presentations. A longer version of this article and a slide show showing pictures of the equipment and the room are available at <http://bmrc.berkeley.edu/research/nossdav05>.

Capture process

The basic idea behind presentation capture is to capture audio, video, and graphics (that is, RGB output from a presentation computer) and encode it into a compressed digital media file that users can replay on demand. The challenge is to capture high-quality images of the projected presentation material inexpensively. Most conference presentations use relatively static slides with transition effects and builds. Occasionally, a presentation will include animations to illustrate dynamic behavior. Some presenters use continuous media (such as audio and video) and live demonstrations in their presentations. Dynamic behaviors, continuous media, and live demonstrations are especially difficult to capture.

The conventional approach to lecture capture is to use one or two cameras focused on the speaker—for example, for a close-up of the speaker and a wide-angle shot of the stage—and a wireless microphone to capture the speaker's audio presentation. Some productions use a third camera to capture audience members as they ask questions.

Problems arise when you try to capture the graphics material projected to the audience. Typically, the computer's RGB signal is converted to a video signal by using a scan converter or pointing a camera at the projection screen. Both approaches have limitations because the RGB sig-

Editor's Note

Working on multimedia and e-learning areas, you might have heard about the Berkeley MPEG-1 Tools, the Berkeley Multimedia, Interfaces, and Graphics (MIG) Seminar/Lecture Webcasting System, or the Open Mash Streaming Media Toolkit. All these achievements were produced from the group led by Larry Rowe.

In this issue, we invite Rowe to introduce his latest low-cost system on automated presentation capture, which covers both the technology and the process. In particular, he shares with us some valuable thoughts on improving the quality of the captured material, the process from capture to postproduction, the system usability (such as the user-friendly interface), and the media streaming protocols to support playback.

—Qibin Sun

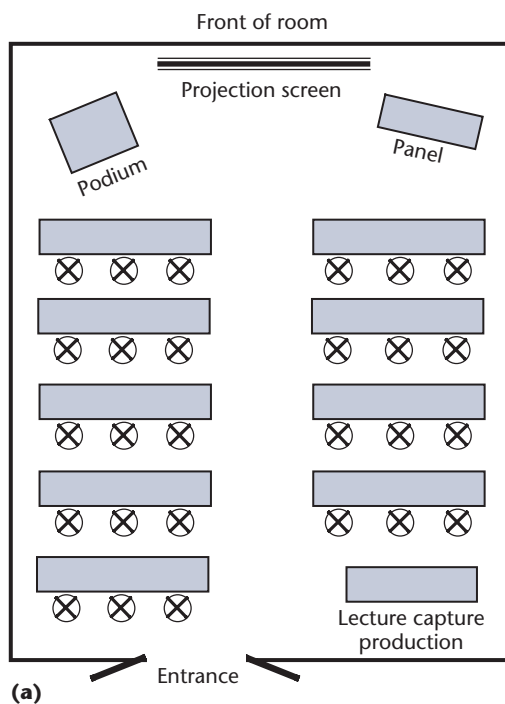
nal has too much data compared to a video signal. Digitizing and compressing these images discards 50 to 70 percent of the image's information, which often results in unreadable presentation material.

Another approach is to acquire the presentation source files and create images in postproduction that can be synchronized with the speaker's audio and video.⁵ This approach produces high-quality slide images but raises the cost of production unless speakers are constrained to a limited set of presentation packages. Capturing dynamic material is still problematic. Moreover, some speakers won't provide copies of their files. A prior experiment at ACM Multimedia 2001 used this approach, and the published material contained only 30 percent of the slides.⁶

In our approach, we directly capture the RGB signal using an **NCast Telepresenter G2**. (We should note that this article's first author, Rowe, is a cofounder and investor in the company.) With this method, the image quality is substantially better, and we capture the dynamic material. The G2 (see <http://www.ncast.com/telepresenterG2.html>) contains an embedded computer that runs software to digitize audio and RGB signals and then compresses them using MPEG-4 codecs. It can webcast live streams and archive the material in an MP4 file for on-demand replay. The G2 produces material compatible with Internet Engineering Task Force (IETF) and International Telecommunications Union (ITU) standards that users can play using the Apple QuickTime Player. The G2 can be controlled by using the embedded Web interface or by a program that accesses the G2 through a Transmission Control Protocol (TCP) or serial connection. The G2's retail cost is \$5,500.

The G2 captures RGB images, so we need to convert the National TV Standards Committee (NTSC) video signal produced by cameras recording the speaker into an RGB signal. We used a Kramer VP-720DS seamless switcher (see <http://www.kramerelectronics.com>), which accepts up to four video inputs and one RGB input and produces an RGB output selected from one of the inputs.

The VP-720DS has been discontinued, but Kramer makes many similar products that can be used for this application. The switcher scales the selected input to the specified output format and uses frame-accurate switching. It also provides a picture-in-picture (PIP) function that will show the RGB signal composed with one of the video



(a)



(b)

Figure 1. Pictures of the room in which the NOSSDAV conference was held. (a) Room configuration, and (b) view of the room from the speaker's podium.

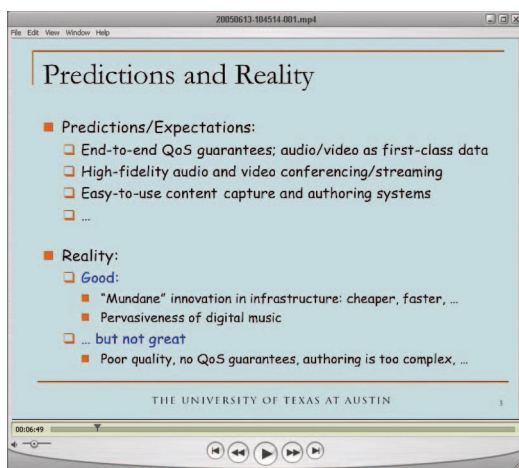
signals or one of the video signals composed with the RGB signal. The retail cost of the VP-720DS is \$1,595, although they're widely available for \$1,200.

We used two cameras: a manually controlled camera at the back of the room and a pan, tilt, and zoom (PTZ) camera located in the aisle between the classroom seating tables. Figure 1 shows a schematic of the room and a picture from the podium. We used the manual camera to provide a wide-angle view of the stage and to show people asking questions. The PTZ camera was used for close-ups of the speakers and panel members. The captured presentation is a single video stream that shows the speaker, the presentation material, or the presentation material with

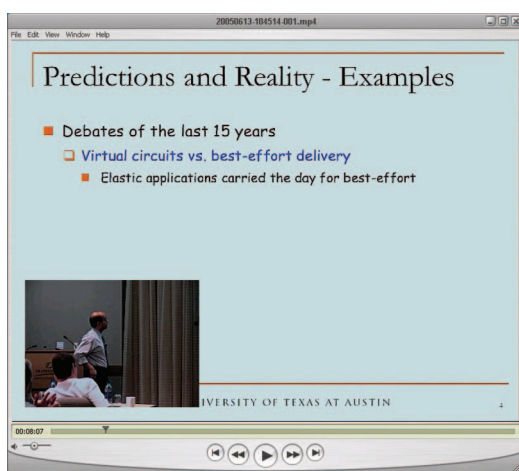
Figure 2. Three examples of the material captured for a presentation. (a) A close-up of the speaker, (b) the presentation material, and (c) a composition showing the speaker and the slides using picture-in-picture.



(a)



(b)



(c)

the speaker in a PIP window. Figure 2 shows examples of each.

Figure 3 shows the equipment configuration we used during the capture. The director (Casalaina) operated the wide-angle camera, an

audio mixer to control sound levels, and two GUI applications that ran on a laptop to control the PTZ camera (a Canon VCC4) and the capture and switching hardware.

The house audio system provided a single audio signal that combined output from the wireless microphone, a wired podium microphone, and audio from the speaker's presentation computer. The podium microphone captured audience questions and speaker introductions.

We designed the control software to be easy to use and to provide only the functions required for lecture capture. Our hope was to automate as much as possible of the production process.

One application controlled the PTZ camera and a second application controlled the capture and switching hardware. The camera control application provided an interface to pan, tilt, or zoom the camera smoothly at a user-configured speed and to set or recall up to six preset positions.

The capture/switching application provided functions to control capture (for example, to start, pause, resume, or stop), select the video source (such as a wide-angle camera, close-up camera, or RGB signal), control use of PIP, and configure selected hardware properties such as the capture format (video graphics array [VGA], super video graphics array [SVGA], and Extended Graphics Array [XGA]).

We wrote the control applications in Tcl/Tk, which together include approximately 3,500 lines of code. The code sends commands to the VCC4 camera and VP-720DS using serial connections and to the G2 using a TCP connection. (More details on these applications, including screen dumps that show the interfaces, are available at <http://bmrc.berkeley.edu/research/nosdav05/capture/>.)

The conference was scheduled for Monday and Tuesday, so we arrived Sunday to prepare. It took approximately three hours to setup and test the equipment on site.

During the event, the director operated the equipment and monitored the capture. We ran the audio signal through a small mixer so we could easily control sound levels. Video was monitored on an RGB display from the G2 that showed the captured video. The G2 display can be configured to show a sound meter for captured audio. Hence, we were able to verify that the sound was intelligible and the captured signal was acceptable. The production assistant (Rowe) solicited performance releases, helped speakers with RGB output settings, and tweaked

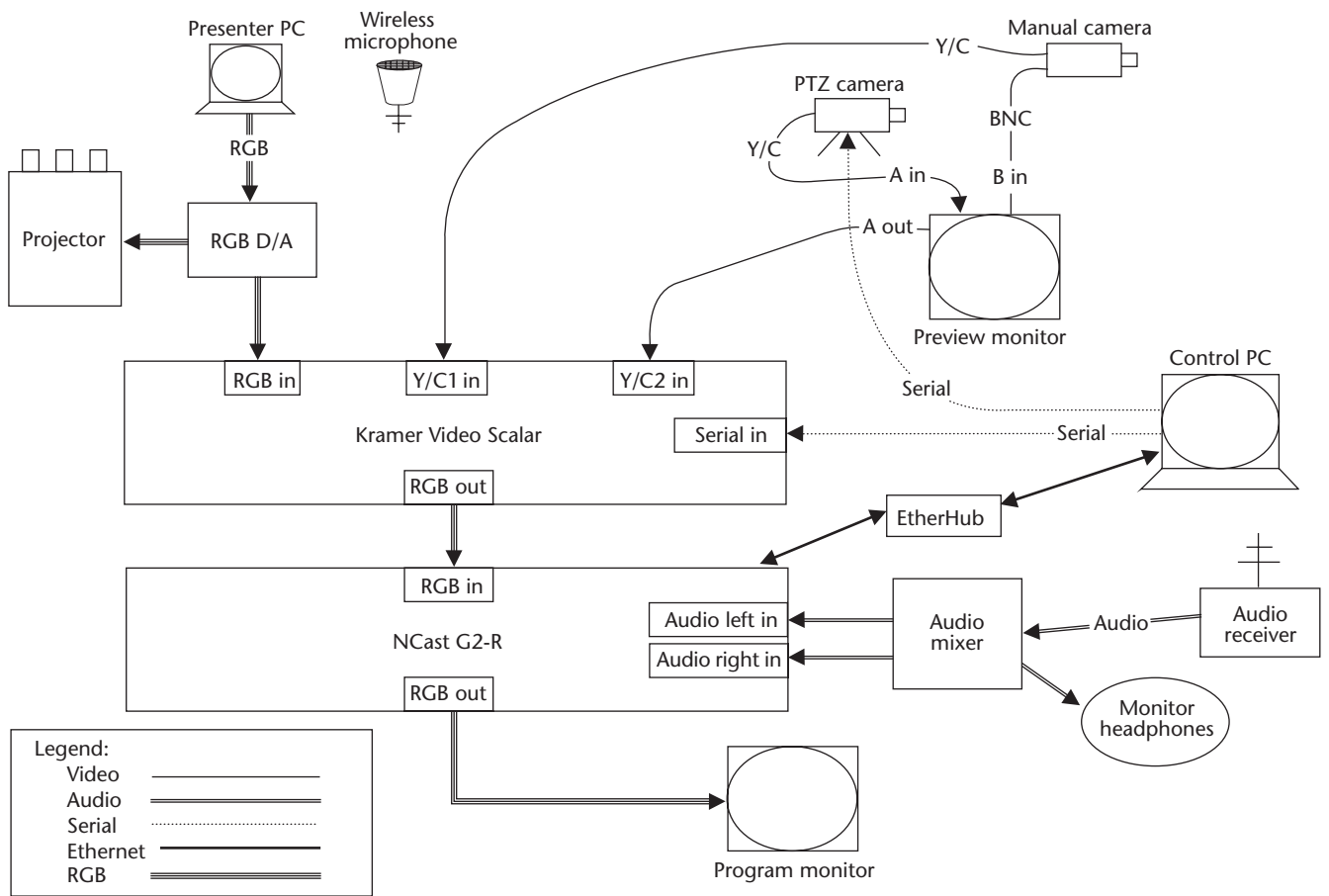


Figure 3. The audio, video, and computer equipment used to capture the presentations. The presenter's PC and wireless microphone and the presentation projector are in the upper left corner. We brought all the other equipment to capture the event. The schematic depicts the various interconnections and signals (such as video and audio). The producer uses the audio mixer and monitor headphones, the preview and program monitors, and the control PC to capture the presentation. The preview monitor lets the producer set up the camera not currently selected for program output.

the control software. (ACM approved the performance release, which is available at <http://bmrc.berkeley.edu/research/nosssdav05/capture/acm-vid-release.pdf>, before the event.)

Previous experience suggested that 67 percent of the presenters would sign the release. Often presenters decline because they are uncertain whether they had releases for material used in the talk or because they worked for organizations that required that corporate lawyers sign the releases. All the NOSSDAV presenters signed the release, probably because most speakers were from universities.

The G2 stores the captured media files on an internal disk. We captured the conference organizers' welcome and introduction, presentations for the 33 accepted papers, the keynote address, and nine question and answer sessions, which

produced 44 media files that occupied 8.7 Gbytes of disk space. Sadly, the video for one talk wasn't recorded correctly for unknown reasons. It might have been an operator error starting or stopping a capture or a software bug. We still published the audio for the talk.

It took approximately an hour to tear down the equipment and repack it for transportation after the conference ended. We also made a copy of the media files on a separate disk just in case there was a problem on the return trip.

Postproduction

As we mentioned earlier, the NCast G2 produces files that users can play on a QuickTime Player. We installed a Darwin Streaming Server (DSS) on a FreeBSD PC located at the University of California, Berkeley, and loaded the captured

**It took some effort, but
eventually we were able
to get the HTML to work
correctly on all Web browsers
using the embedded
QuickTime Player.**

files onto it. We then played the material using various Windows and Macintosh PCs from different places including high-speed connections at Berkeley and other universities and broadband connections at home. The captured material didn't play well for two reasons:

- The material was captured at 1.5 megabits per second (Mbps), which is too demanding for many broadband home connections.
- The material was captured at the native image size of the presentation, typically XGA, at 30 frames per second (fps). A relatively new PC was required to decode this material.

Consequently, we decided to recode the material so that more people could play it. We had trouble finding an inexpensive software package to transcode the files. We found several packages that appeared to work, but they cost between \$400 and \$1,000. While we were searching for the best alternative, Apple released QuickTime V7 Pro, which includes the required transcoding functionality, runs on Macintosh and Windows PCs, and costs \$30.

After experimenting with different formats, we decided to publish two versions of each presentation, specifically a low-quality version that users can play anywhere and a high-quality version for people with fast network connections and computers. The low-quality version uses 384×256 images at 15 fps that require 600 kilobits per second (Kbps), and the high-quality version uses 512×384 images at 15 fps that require 1,200 Kbps. We used the recently released QuickTime H.264 video codec for the published material because the transcoding software supported it

and it appeared to produce better results than the MPEG-4 video codec.

Transcoding all the material was time consuming because it required three and nine times real time to produce the 600 and 1,200 Kbps material, respectively. The H.264 codec in QuickTime V7 Pro has one- and two-pass encoders. We used the one-pass encoder even though the results were better with the two-pass encoder because the two-pass encoder required 40 times real time to transcode a file. We had 14 hours of material to transcode at two settings. Using the one-pass encoder, it still took approximately 170 hours to transcode the material.

We produced Web pages to play the material, including a listing of all talks and popup windows to play each talk. It took some effort, but eventually we were able to get the HTML to work correctly on all Web browsers using the embedded QuickTime Player.

Publication

The conference was held 13–14 June 2005, and we published the material on 1 September 2005. (The presentations are available at <http://bmrc.berkeley.edu/research/nosssdav05/>.) The ACM SIG Multimedia and NOSSDAV Web sites and mailing lists advertised the material's availability. Users were able to play the material successfully 329 times, which is 60 percent of the attempted plays (548), in the 11 months between September 2005 and July 2006. We've omitted from these statistics plays by the site producer during development and testing.

Of the 219 failed attempts, 179 (80 percent) were logged as server timeout errors, which are caused by users trying to play the material on a computer behind a firewall or network address translation (NAT) router using a datagram protocol (such as Real-Time Streaming Protocol [RTSP]) rather than a TCP-based protocol (such as HTTP). Most of these errors occurred in the first two months.

The player or server software didn't work during November and December 2005, which we discuss in more detail in the next section. We changed the way the videos were played in early January so that all playback used HTTP transport. Since that time, we've noticed a significant decrease in server timeout errors. The other 40 errors are bad requests (for example, the URL doesn't exist or a wrong format is requested).

Looking at the successful plays, 35 percent used the high-speed version and 64 percent used

the low-speed version. The remaining 1 percent played the audio-only talk. We're surprised that more people didn't play the high-speed version because we expected that most people interested in the material would be at universities, which typically have high-speed connections that can access the Berkeley server.

Each talk and Q&A session was played between 0 and 40 times with a 7.7 mean number of plays (standard deviation 9.1). Surprisingly, three talks have never been played.

The most popular talks are

- Keynote address "Multimedia Systems Research: A Retrospective" by Harrick Vin from the University of Texas (40 plays),
- "Supporting P2P Gaming When Players Have Heterogeneous Resources" by Brian Neil Levine from the University of Massachusetts (36 plays), and
- "Mirinae: A Peer-to-Peer Overlay Network for Large-Scale Content-Based Publish/Subscribe Systems" by Yongjin Choi from KAIST (31 plays).

The most popular panel discussion was on "Network Gaming," which has been played five times and included researchers Brian Neal Levine from the University of Massachusetts, Chris Chambers from Portland State University, Grenville Armitage from Swineburne University, and Kuan-Ta Chen from National Taiwan University.

We're disappointed the material has been played only one to two times per day. Although we expected replays to decline over time, we thought people interested in the topics who didn't attend the workshop would play the material. The problem might be publicity, since it's difficult to advertise the material's availability, and the content's one-time nature.

Lessons learned

Several things worked well, including the switching and capture hardware and the low-cost model for capture and publication. Although we believe it's possible to capture and publish a single-track conference for approximately \$3,000 a day plus expenses, this price will of course be higher if you use additional equipment. Still, it's reasonable to expect the cost to remain well under \$5,000 per day. Also, we were able to cap-

ture all the workshop presentations, regardless of the slide and computer technology the speaker used, including all dynamic material. We believe the resulting published material is of reasonable quality given the playback constraints (network bandwidth and computer processing power) and a few production glitches we'll discuss shortly. Nevertheless, as in any production, there's room for improvement.

Improving quality

Generally speaking, the material we captured is good quality, but it can be improved. First, we captured the material at 30 fps using the native resolution of the presenter's projected material if the resolution was XGA or smaller and XGA resolution if larger. Although it reduces visual quality, a lower resolution capture (such as SVGA) at 15 fps is good enough given the constraints of current playback technology.

Scaling higher-resolution images to SVGA and applying typical video coding algorithms produced some "ringing" around text on the slides—that is, ghost edges around the characters. Modern computers are exceptionally good at displaying material at different resolutions. Where possible, we need to encourage presenters to use lower resolution when projecting their material. This problem is related to bandwidth available for transmitting the material during playback and decoding efficiency of the playback computer. Over time, these constraints will be relaxed, and it will be practical to capture larger images at higher frame rates.

We could also improve audio capture. Some speakers didn't use the wireless microphone. The captured audio was good if they stayed at the podium, but sometimes they strayed away from the podium or looked at the screen, which impacted quality. The obvious advice is to force speakers to use the wireless microphone.

Audio capture of audience questions must be improved because they were sometimes difficult to hear. We thought the podium microphone would pick up most audience questions, which it did. However, sometimes the audience member didn't speak loudly, and it was difficult for the director to change the sound level quickly enough during interaction between the speaker and audience. We should have used several microphones pointed at the audience and controlled them separately at the mixer to capture questions.

Finally, we had only one wireless microphone. We needed several microphones so the session

moderator could always be wired and the next speaker could get ready before it was time to talk.

We also had a minor problem positioning the RGB image on the projection screen and at capture. The projector in the room had a remote control to move the image left/right or up/down, but we didn't notice the problem during testing. As a result, the RGB images in the first few talks were shifted up and to the left when captured, which led to video noise across the bottom of the captured images. Both the VP-720DS and the G2 have controls to move the image, but we didn't have access to them in our control software. This problem can be easily fixed.

Another way to improve the captured material's quality is to use more cameras and provide the director with more control. We didn't incorporate an audience camera positioned at the front of the room because we didn't have an extra camera. We will do so in the future as long as audience members don't object. And we will use PTZ cameras for all sources rather than a manual camera because it will simplify operation for the director.

Wide-angle views of the stage were unusable when slides were being projected because the bright light bouncing off the screen caused the camera auto exposure to close the aperture, which produced a dark image that made it difficult to see the speaker. A good spotlight on the speaker will fix this problem.

NCast has released the Telepresenter M3 with additional production features including a PIP function and a graphic overlay function that can be used for titling. Reducing the hardware simplifies setup and operation and improves reliability.

Lastly, Automatic Sync Technologies (see <http://www.automaticsync.com>) is a commercial company that offers an automated captioning service for streaming media. They produce a multimedia title using any of the popular streaming media formats that scrolls text of an audio transcript synchronized with the audiovideo material. They can also produce a word-based search index to the material. The service costs approximately \$185 per hour of source material, which means the NOSSDAV material could be processed for less than \$3,000. We think future publication of presentations and discussions at conferences should include this capability with the material they publish.

Improving process

We could also make several changes to

improve the capture and postproduction process. First, a preconfigured custom hard-shell case for the production equipment would greatly simplify preparation before an event and setup at the remote location. The case can incorporate a small rack for the equipment with sound dampening and access to the front and back panels. We can also add small rack-mounted LCD displays for monitoring various video sources in place of the heavy, awkward-sized professional video monitor we used in this experiment. These cases are relatively inexpensive and many companies will custom design them for a specific application.

Second, we can substantially improve the postproduction and publication process. Because it was the first time we used this approach in a conference setting, it took almost two months to publish the captured material. This delay was caused in part because we had to determine the best playback representation and transcode the material. In future productions, we will change the capture parameters to avoid transcoding. We also had to setup the media server and author Web pages for the conference program and individual presentations. Most of this work only needs to be done once or can be automated.

During the event, we spent considerable time keeping track of the speaker and the recorded file that corresponded to each talk. The G2 identifies the talk by encoding the beginning date and time of the capture into the file name. We copied the material off the G2 by hand and then used scripts to produce the Web pages given the files and information about the talks (for example, title, authors, affiliation, speaker, talk duration, and start time). We can easily automate this step by entering the conference program ahead of time and relating it to the capture files. Moreover, we could open up the G2 interface to the embedded FTP server and automate the entire postproduction process.

Finally, several research groups have explored automating the decisions made by a webcast director during a live event.^{2,7,8} Clearly, this technology should be incorporated into conference presentation capture.

Improving usability

Numerous changes can be made to the control software used to capture the event. First, we need to fix the PIP interface. The control software needs a simple configuration interface that lets the director change the PIP location (to bottom left or right) to more easily adapt to the spatial

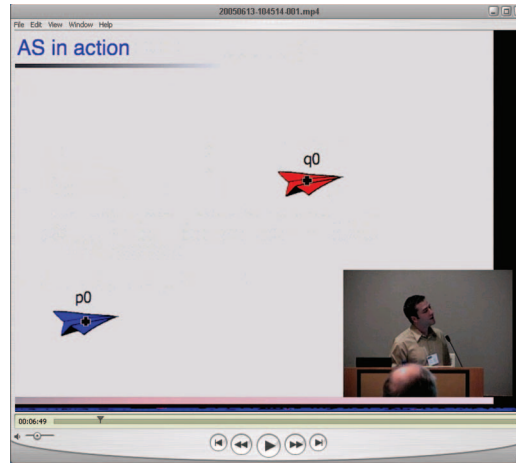
positioning at the conference venue. If the PIP window is on the lower right side of the image and the speaker is standing to the left of the screen as you face the stage, the speaker's gesture to the screen on his left is off the right edge of the captured image. By moving the PIP window to the lower left, the speaker gesture points to the slide in the captured image. Figure 4 illustrates this problem. If the speaker is on the screen's right side, you need to change the PIP position. Hence, the control software must make it easy for the director to change the PIP location dynamically.

This feature is easy to add because the Kramer interface has the function. But one problem with the Kramer is the absence of a function to switch the PIP and main window source. This function exists through the VP-720DS onscreen display interface, but we couldn't execute that function remotely, even when we tried to mimic the onscreen operations. The device clearly has the function, but it's unavailable through the serial control interface.

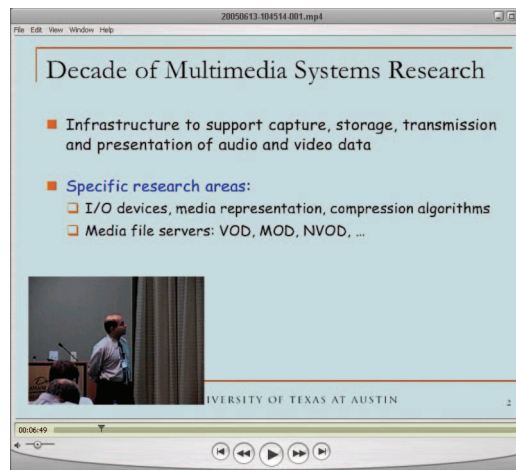
This limitation caused problems because several times the director wanted to swap the PIP and main window images. To do it, he had to turn off the PIP window, switch to the alternative source, and turn on the PIP window, which was distracting and time consuming. Moreover, it probably increased the compressed bits because the codec produces encodings for the intermediate images.

To improve camera control, we need to rewrite the PTZ camera control software. The software we used was developed originally for a Canon VCC3. We used the VCC3 emulation mode on the VCC4 camera. The VCC4 also has more functions (such as variable speed moves) that we can exploit to improve the captured images. The VCC3 has manual iris and focus controls, but we couldn't get them to work in the emulation mode on the VCC4. Presumably, the VCC4 interface to these controls works.

Finally, we need to define some presets to move the PTZ camera in one or two dimensions and to add more presets. A preset in the current software defines an absolute setting for pan, tilt, and zoom. Several times the director wanted to pan to the right or left at the same tilt and zoom settings. In effect, he wanted a delta from the current position, rather than an absolute setting for a preset. We also need to add groups of presets so the director can easily switch between them. For example, individual speakers and panel sessions require different settings.



(a)



(b)

Figure 4. Spatial relation between speaker and PIP window. It must be easy for the director to change the PIP location dynamically to avoid having (a) the speaker gesture off screen. (b) Preferably, the speaker will gesture onto the screen.

Improving playback

We used the QuickTime Player embedded in a Web page to play the recorded material. Several users had problems playing the material. Generally speaking, it worked well on Macs running OS X using the Safari Web browser. Although users were able to play the material using Windows computers and other browsers (such as Firefox and Internet Explorer), most had problems with streaming transport because the user had to manually configure it. Users have no patience for configuring software to view material like these presentations. Playback must work like TV: go to a Web page and it works.

The QuickTime embedded player can transport content using either RTSP or HTTP streaming. Given the state of the Internet today, nearly everyone uses HTTP streaming because of firewalls and NAT routers. However, the player uses RTSP streaming by default so the user must reset the transport parameter manually. Most users,

including experienced computer scientists, were confused by this requirement even though our Web pages described the problem and explained how to change the setting.

Moreover, a recent release of the QuickTime software for Windows (version 7.0.3) exacerbated this problem. Prior to this release, the user could set the transport to use port 8000 with HTTP streaming. This release doesn't let users change the port—they must use default port 80. This restriction, or more likely bug, caused problems because we run the DSS server on the same machine as a Web server. We didn't notice this problem with the material for more than two months because no one notified us that the material was unplayable. The server logs show that people just stopped playing the material.

We fixed this problem by explicitly including the port number (8000) in the RTSP URL we used to launch playback. This port uses HTTP streaming by default, so it removed the requirement that people explicitly set the transport parameter.

Conclusion

This experiment demonstrates that it's possible to capture conference and workshop presentations for on-demand replay for \$3,000 per day. We believe professional organizations such as the ACM and IEEE should consider capturing presentations for many, if not all, conferences. Over time, this cost should decline and the quality of the captured material will improve. **MM**

Acknowledgments

We thank the ACM Special Interest Group Multimedia Chair Ramesh Jain for funding this experiment. We also thank Bobb Bottomley, who is responsible for the audiovideo technology at Skamania Lodge where the conference was held. Lastly, we thank all the speakers who agreed to be captured for posterity.

References

1. L. Rowe et al., *BIBS: A Lecture Webcasting System*, Berkeley Multimedia Research Center Technical Report, Univ. of Calif., Berkeley, 2001; <http://bmr.berkeley.edu/bibs-report>.
2. Y. Rui et al., "Automating Lecture Capture and Broadcast: Technology and Videography," *Multimedia Systems J.*, vol. 10, no. 1, 2004, pp. 3-15.
3. A. Steinmetz and M. Kienzle, "The E-Seminar Lecture Recording and Distribution System," *Multimedia Computing and Networking 2001, Proc. Int'l Soc. for Optical Engineering*, vol. 4312, SPIE, 2001, pp. 25-36.
4. W.-C. Feng and K. Mayer-Patel, eds., *Proc. 15th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video*, ACM Press, 2005.
5. S. Mukhopadhyay and B. Smith, "Passive Capture and Structuring of Lectures," *Proc. 7th ACM Int'l Conf. Multimedia*, ACM Press, 1999, pp. 477-487.
6. SOMA Media, *ACM Multimedia 2001: Conference Presentations DVD*, ACM Press, 2002.
7. M. Bianchi, "Automatic Video Production of Lectures Using an Intelligent and Aware Environment," *Proc. 3rd Int'l Conf. Mobile and Ubiquitous Multimedia (MUM 04)*, vol. 83, ACM Press, pp. 117-123.
8. E. Machnicki and L.A. Rowe, "Virtual Director: Automating a Webcast," *Multimedia Computing and Networking 2002, Proc. Int'l Soc. for Optical Engineering*, vol. 4673, SPIE, 2002, pp. 208-225.

Readers may contact the authors at rowe@cs.berkeley.edu and VinceSail@aol.com.

Contact *Multimedia at Work* editor Qibin Sun at qibin@i2r.a-star.edu.sg.



IEEE distributed systems ONLINE
Expert-authored articles and resources

IEEE Distributed Systems Online brings you peer-reviewed articles, detailed tutorials, expert-managed topic areas, and diverse departments covering the latest news and developments in this fast-growing field.

Log on for **free access** to such topic areas as

**Grid Computing • Mobile & Pervasive
Cluster Computing • Security • Peer-to-Peer
and More!**

To receive monthly updates, email
dsonline@computer.org

<http://dsonline.computer.org>