eMeeting: A MULTIMEDIA APPLICATION FOR INTERACTIVE MEETING AND SEMINAR

[1]Wing Ho Leung, [1]Tsuhan Chen, [2]Ferdinand Hendriks, [2]Xiping Wang, [2]Zon-Yin Shae

[1]Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
Email: {wingho, tsuhan}@andrew.cmu.edu

[2]IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
Email: {hendrik, xiping, zshae}@us.ibm.com

ABSTRACT

In this paper we present a client/server-based un-intrusive collaborative multimedia application - called eMeeting - that extracts and archives salient visual information from seminars, while allowing interaction among presenters and the audience in real time. The application uses API's of the Lotus Sametime® system such as secure login, directory, meeting and multimedia services, to which we have added our own extensions. These extensions use the binary messaging channel of Sametime and allow us to add stroke-based chat as well as stroke-based annotations of slides. This permits communication among all clients. In parallel with the services just mentioned and in the same Java application, users of eMeeting are presented with slides that the system extracts from the presenter's visual material using an automatic recognition algorithm that classifies video frames as slides. At this time we assume that the system may or may not have access to the visual content in electronic form. Users can follow the seminar in the form of presenter's video, and as a series of extracted images. The server transmits the slides cyclically to each user using IP multicast giving each client the opportunity to select any previous slide. Thus, late joiners can browse slides that have been accumulated by the system.

1. INTRODUCTION

There is an on-going joint effort between Electrical and Computer Engineering Department of Carnegie Mellon University (CMU) and IBM Watson Research Center to broadcast CMU seminars across Internet. The first attempt is to build a system to broadcast seminar from CMU to Watson Research New York across Internet 2. This paper is to report some of the core components of this system. Electronic collaborative systems are widely used for education, corporate meetings and so on. There are several such systems available. For example, AutoAuditorium® by Foveal Systems [1]. It is primarily a video capturing, archiving and retrieval system containing multimedia presentations observed in a specially equipped auditorium to ensure audio and visual quality. The archives consist of video with segments of mixed content that typically originates from different cameras. Thus there is a camera following the presenter, ideally aided by speaker-tracking array microphone, and another camera that captures the projection screen and yet another aimed at the audience. A vision algorithm is used to determine shot selection. In their system, slides are treated as a video sequence. Consequently a large bandwidth is required and also it losses the flexibility for user's interaction. The slide file can be obtained from the presenter, but lack of the synchronization between the presenter's video sequence and the slide. Another system is the Berkeley Internet Broadcasting System (BIBS) [2]. This system allows access to lectures, both in real time and in archived form. A user receives the presenter's video stream in Real® format, as well as slide images and can navigate a timeline to select a particular point in the presentation. Typical views are presenter and (static) slide. Their system is intrusive, i.e., the indication of slide change will have to come from the presenter via either a dedicated machine for the presenter to use in the seminar room or an offline mechanism is required to obtain the synchronization in time between the video sequence and slide file.

The purpose of the electronic collaborative system is essentially to enable users to effectively work together on common objectives in a networked computer environment across geographical and time boundaries. The motivation is to allow users to interact with each other in real time and also permit them to join the meeting at any time without missing the presenter's slides which appeared earlier. In addition, a method should be provided for users to easily track contents in a shared environment during discussion.

Unfortunately, no current system supports such important functionalities. We wish to retain the possibility for user participation and feedback while the seminar is taking place, permit content browsing for late joiners and create a system that can run unattended. Our eMeeting system, discussed in greater detail below, primarily focuses on addressing these issues. It provides an un-intrusive environment for the presenter as well as real time interaction among participants using an enhanced instant messaging tool and supports the late meeting joining service by periodically broadcasting the presenter's slides. The content tracking is achieved by creating annotations over existing contents. The system does not make any special demands on the presenter. All the slides are automatically extracted with a slide detection algorithm from an ordinary camera. A graphical user interface is provided to contain views and controls supporting a presenter's view rendered as video, a slide view rendered as an image, and a real-time messaging view that allows a user to do text chat or stroke-based chat and annotation. In addition to visual information, real time audio is also available to users. The audio source typically belongs to the presenter, although the system also supports multiple audio inputs with microphone arbitration.

This paper is organized as follows. In Section 2 we provide the system description of eMeeting. In Section 3 we describe the real-time slide processing component. In Section 4 we discuss the eMeeting client. The conclusions are given in Section 5 and future work is outlined in Section 6.

2. SYSTEM DESCRIPTION

The system architecture is shown in Figure 1. It models a two-camera seminar broadcasting system. One camera produces the presenter's video sequence; the other is for the slide content shown on the projection screen. The presenter is free to make the slide presentation using his own laptop computer, slide projector, or overhead projector. The presenter need not copy the file to the site's dedicated computer, nor does he have to do perform any other action (e.g. push a button) to signal the system for a new slide. Thus our system does not intrude on the presenter. The "mix video sequence" functional box shown in Figure 1 provides the slide detection mechanism which will automatically select the slides out of the video sequence for the special delivery in order to save the communication bandwidth. It also enables rich user interaction functionalities e.g. arbitrary slide viewing sequence and ink annotation over the slides. This functional box is also able to recognize individual media types, e.g., video, slide, and ink. This system also provides an environment for the presenter to draw his/her impromptu ideas or give more detailed explanation on an electronic whiteboard. We have implemented simple whiteboards with digitizers such as the ones by Mimio® [3] and eBeam® [4].

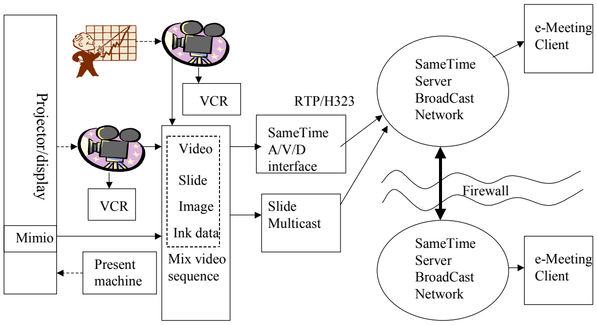


Figure 1 System Architecture

The captured video and audio sequences are fed to the IBM Lotus Sametime®[5] client interface for the broadcasting. Due to the slide's unique property, it is broadcast while bypassing Sametime® to enhance the system's functionality. The slides are broadcast periodically such that the late-join users of a seminar broadcast can access the slide history of the talk. The history log of the slides can be saved in the web server for the late-join users on an on-demand basis. We found the periodic broadcasting mode very useful in supporting a large number of participants. Sametime® servers form a distributed network for better scalability as well as secure tunneling across firewalls. This distributed tunneling architecture dramatically reduces Internet bandwidth. The entire media content can be encrypted for secure broadcasting. The eMeeting client allows users to view any slide that has been presented freely in any sequence. It also allows users to annotate slides for later review. Instant messaging capability is also supported for user interaction with the presenter. The detailed description of the slide processing and the eMeeting Client will be provided in the following sections.

3. REAL-TIME SLIDE PROCESSING

The goal of the real-time slide processing unit is to apply some image processing techniques to analyze the input captured frames. Instead of sending out all the captured frames as video, the frames are processed first to remove redundant information, and only new slides will be sent out. As a result, it results in a reduction in bandwidth requirement. The bandwidth can be allocated more efficiently in order to allow retransmission of previous slides for users to retrieve them in real-time.

Figure 2 illustrates the real-time slide processing unit of eMeeting. The input is a captured frame from a camera. This frame is processed by the slide detection module to determine whether it is a potential slide. A potential slide is passed to the slide change detection module to detect for new slide. A new slide is compressed by the slide compression module and is turned into bit stream. The bit stream is then formatted into packets and the slide transmission module will transmit the packets through the network to the eMeeting client. Each module in the real-time slide processing unit is described in details in the following sections.

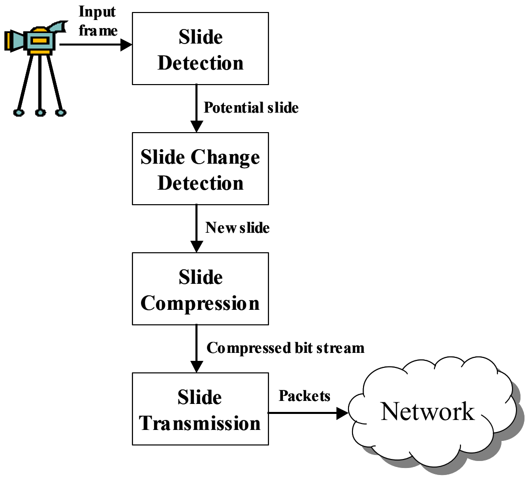


Figure 2 Real-time slide processing unit

3.1 Slide Detection

Given an input frame captured by a camera or other capture device, this slide detection module is used to decide whether the captured frame is a potential slide. A potential slide is the captured frame containing the presentation slide when there is no significant occlusion. For example, if the camera is capturing the scene when an audience is asking a question, then the captured frame is not a potential slide. If the presenter occludes a significant portion of the projected slide, then the captured frame is also not a potential slide. On the other hand, if the camera is capturing the projected slide which is not occluded by any other object, then the captured frame is a potential slide. Figure 3 shows examples of a potential slide and a non-potential slide.

In order to determine whether a captured frame is a potential slide, two features are extracted from the captured frame: the maximum peak of the color histogram and the sum of absolute difference in entropy for the horizontal lines are used. Often in a presentation, the background of the slides has a dominant color which corresponds to a peak in the color histogram. Figure 4 shows the maximum peak of the histogram of each of the RGB components for a potential slide and for a non-potential slide. It can be observed that the maximum peak of the color histogram for a potential slide is larger than that of a non-potential slide.

Moreover, when a slide contains text, then the pixel intensity distribution of each horizontal line will tend to have a larger difference between the text and non-text region. This difference can be detected by considering the sum of absolute difference between the entropy of the horizontal lines, which is defined by:

$$SAD = \sum_{y=2}^{N} \left| L(y+1) - L(y) \right| \tag{1}$$

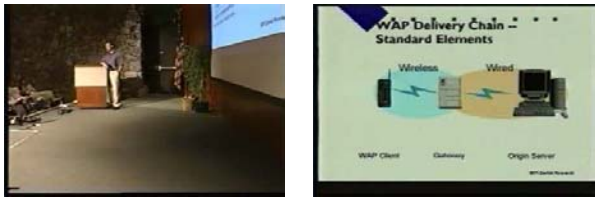
assuming that $I(x,y)$ is the pixel intensity of an $M \times N$ gray-scaled image I at pixel location (x,y) and $L(y)$ is the entropy of the y-th horizontal line and is defined by:

$$L(y) = -\sum_{i=0}^{255} p_i(y) \log_2 p_i(y) \tag{2}$$

where $p_i(y)$ is the occurrence frequency of the pixel intensity for i = 0,1,…,255 and it is defined by:

$$p_i(y) = \frac{\sum_{x=1}^{M} F_i\left(I(x,y)\right)}{M} \tag{3}$$

$$F_i\left(I(x,y)\right) = \begin{cases} 1 & \text{if } I(x,y) = i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$



(a) Frame 375 (b) Frame 500

Figure 3 Examples of (a) non-potential slide and (b) potential slide

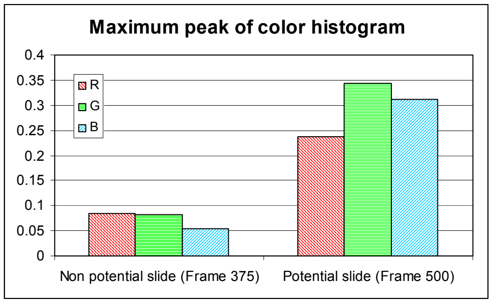


Figure 4 Maximum peak of color histogram

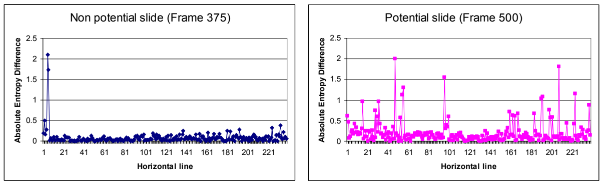


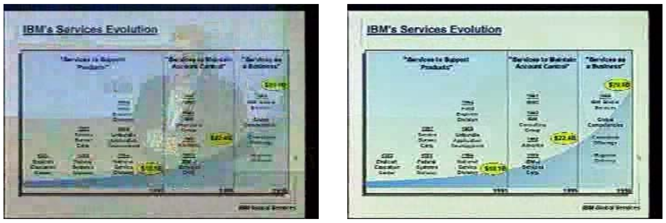
Figure 5 Absolute difference of entropy between horizontal lines

Figure 5 shows the absolute difference of entropy between horizontal lines for a potential slide and a non-potential slide. It can observed that the potential slide has a larger entropy differences than the non potential slide, resulting in a larger sum of the absolute difference of entropy between horizontal lines.

3.2 Slide Change Detection

If a frame is detected to be a potential slide, then it is passed to the slide change detection module to determine whether this potential slide is a new slide or the same slide as previously detected slide. When the presenter is presenting a slide, all the potential slides from the captured frames correspond to the same slide, thus it suffices to keep only one frame identified as the new slide and discard all the other frames.

Sometimes during slide transition, it takes a few frames for the captured frame to get stabilized as shown in Figure 6. As a result, before deciding whether a given potential frame is a new slide, it is first compared with the previous potential slide to determine whether there is any change. This process is illustrated in Figure 7. If there is a change between the current and the previous potential slides, then the current potential slide is not considered as a new slide (potential slide k-1 and potential slide k-2). However, after realizing that the current potential slide (potential slide k-1) is different from the previous potential slide (potential slide k), then the module checks keeps track of whether there is no change in the subsequent potential slides. If there is no change in the subsequent 4 slides (potential slides k+1,…,k+4), then it assumes that the last potential slide (potential slide k+4) is a stable frame and it will be decided as a new slide. Here it is assumed that a potential slide will become stable after 4 frames since a change has been detected. In general, this number depends on the frame rate of the capture device.



(a) Unstable frame (Frame 53) (b) Stable frame (Frame 56)

Figure 6 Examples of (a) unstable frame and (b) stable frame

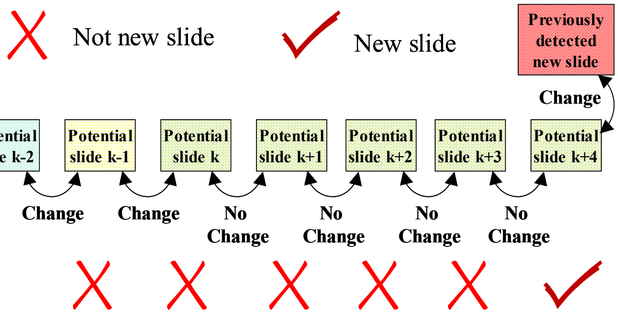


Figure 7 the process of Slide Change Detection

In addition of obtaining a stable frame, the above approach is also good in rejecting the frames where some part of the frame is occluded by the movement of the presenter. As the presenter moves, say going near the projector output and pointing to the slide, then a small part of the slide may be occluded (slide k-1 in Figure 8). The slide detection module may still decide the input frame as the potential slide because the non-occluded part may still have a large maximum peak of color histogram and a large sum of entropy difference of horizontal lines. However, since the captured frame is not stable when the presenter moves (slide k-1 in Figure 8) and the stable captured frame after the presenter moves out of the frame (slide k+4 in Figure 8) is the same as the previously detected new slide, the resulting slide will not be decided as new slide.

In determining whether two slides are the same, the residue frame between the two associated frames is considered, and it is the intensity difference between two frames. The residue frame is divided into 8×8 blocks, and then the local residue (sum of absolute residue) is computed for each block. Two slides are considered as different when there is sufficient number of blocks with large local residue (i.e., local residue larger than a threshold th_{res}). This measure by counting the number of blocks with large local residue is better than using the global residue (summing the absolute difference between two frames) because a frame with camera noise evenly distributed among itself can have a large global residue yet it is undesirable to consider it as a different slide. On the other hand, such a frame will not have a large local residue since the camera noise is assumed to be spread out across the frame thus there will not be sufficient number of blocks with large local residue.

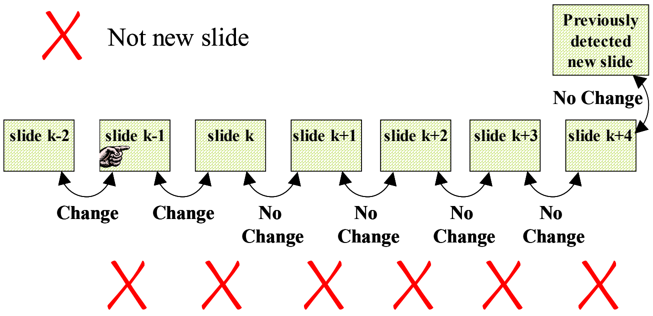


Figure 8 Slide Change Detection for occluded slide

3.3 Slide Compression

The new slide is then compressed using standard JPEG compression to convert the captured frame from RGB domain to compressed bit stream. The compressed bit stream for a frame with resolution 640×480 pixels is typically less than 50KBytes.

3.4 Slide Transmission

The resulting bit stream is chopped into chucks and added with some header information to form packets. The packets are then sent out by multicast. Each packet is around 1KB and two packets are sent out in every 10 ms.

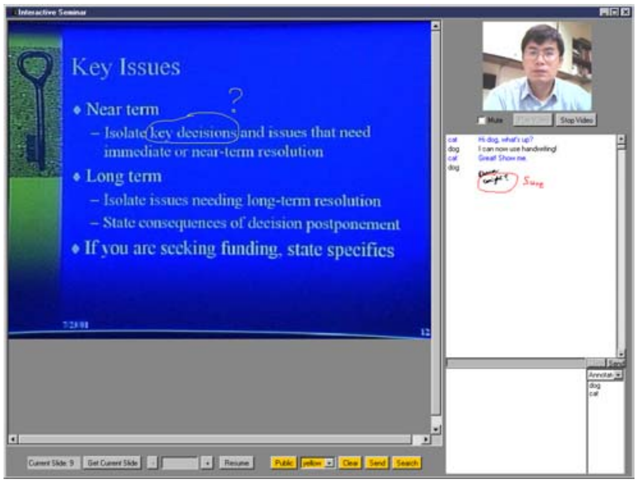
After all the packets associated with the current new slide are sent out, if no new slide has been detected, then the slide transmission module will retransmit the packets of the previously detected slides in sequence periodically. By this periodic retransmission of the slides, the user who joins the meeting late is able to view the previous slides in real-time.

4. eMeeting CLIENT

The client, as mentioned in section 2, plays an important role in the system. It not only receives information from the presenter, but also provides users with a friendly graphical user interface (GUI) that allows them to interact with the presenter and meeting participants. Figure 9 shows the GUI of the client. The slides from the presenter are displayed on the left side together with a slide control panel displayed below the slide display window. The presenter's video shows in real time at the upper right, and an instant messaging panel for meeting discussion is positioned at the bottom-right. In addition to the visual components above, real time audio from the presenter is also provided to all meeting participants so that they can easily follow the presenter's presentation. The combination of slide show, audio, video and instant message enables a powerful presentation and gives a strong sense of presence.

4.1 Slide Viewing

As described in the previous section, each slide of the presenter is compressed with JPEG at the server side and the compressed slides are further packetized and broadcast via UDP with a well-defined payload format. Therefore, a slide is first constructed by assembling consecutive UDP packets and then the assembled slide is fed to a JPEG decompressor. Since UDP is not a reliable transmission protocol, the client must detect packet loss and reconstruct each slide. This can be done in various ways. However, considering the fact that each slide is broadcast periodically to support late meeting joining, the reconstruction of slides can be relatively simple. As one of the possible approaches, we simply drop the slides with missing packets and only display a slide once a complete one is constructed in the following broadcast periods. Extensive experiments using this approach have been done on our intranet and on the Internet and results have proven that this is a very feasible approach in practice although occasionally a slight slide delay may be observed. There are about 50 slides in a typical presentation. Under this assumption, the average slide delay for retrieving a specific slide is several seconds.



Key Issues

- Near term
  - Isolate key decisions and issues that need immediate or near-term resolution
- Long term
  - Isolate issues needing long-term resolution
  - State consequences of decision postponement
- If you are seeking funding, state specifics

cat Hi dog, what's up?
dog I can now use handwriting!
cat Great! Show me.

Figure 9 Client interface

The slide viewing is controlled with a slide control panel as shown in Figure 9. The control panel consists of two parts: a slide retrieving control in gray and a slide annotation control in orange. The slide retrieving control has a modality selector that allows the user to switch the viewing mode between suspend and resume. In suspend mode, users can retrieve a desired slide up to the current slide by typing in a slide number in the slide edit box, while the resume mode lets users view slides in a free run fashion. This gives users a great flexibility to view their desired slides even when they joined the meeting late. Furthermore, the slide annotation control provides a way in which users can make annotations over the presenter's slide either for public discussion or for personal reference. The annotation data is presented in a stroke form and stored in an overlay layer. This makes it possible to perform slide/content search by querying simple graphics marks. The public annotations are visible to all participants in the meeting, while private annotations are only available to a specific user for his/her own reference in the future. The search button in the slide annotation control panel provides slide search functionality offline. The presenter can address questions regarding a specific slide through slide annotation and instant messaging with real time audio.

4.2 Instant messaging

The instant messaging is another powerful part of the client. It consists of text messages, stroke messages and annotations that are combined in a single viewing window. As shown in Figure 9, text messages are input with text edit control, while stroke messages are input via a private drawing canvas. Both text messages and stroke messages can be modified and deleted before sending. The annotation is made directly over the recorded message board. All users are alerted to annotations by means of two hyperlinks: one at the end of the instant messaging record and the other at the annotation itself. All messages including annotations are recorded and displayed on a scrollable whiteboard and visible to all participants in the meeting. Users can easily view all message and annotations simply by scrolling the whiteboard and add further comments if necessary. This combination makes instant messaging even more powerful since it keeps text message's simplicity while allowing users to discuss complex diagrams such as engineering drawings and making annotation right on the desired message.

5. CONCLUSIONS

We have implemented and shown a lecture broadcasting system with rich functionalities both for the presenter and the participants. For the presenter, this system provides a no intrusive environment, i.e., the presenter is free to choose his/her way of presentation using his or her own familiar machine and tools. It also provides presenters a tool-rich environment, i.e., a digital whiteboard for impromptu ideas and detailed explanations. For the participants, this system provides a function-rich, interactive environment available on the desktop, such as late-join user support, real-time review of any presented slide in any sequence, ink annotation over slides for later review, and instant messaging. Integration into the IBM Lotus Sametime® product will enable this system to take advantage of Sametime®'s distributed server network for providing a scalable and reliable framework.

6. FUTURE WORK

Possible extensions of the current work would involve support for video indexing within the framework of MPEG-7. In fact, the slide extraction algorithm already creates metadata for each slide based on color histograms. Currently, the eMeeting client runs only on a PC. Future work would enable pervasive devices, albeit with a restricted set of capabilities.

7. REFERENCES

[1] Foveal Systems, AutoAuditorium®. [Online]. Available: *http://www.fovealsystems.com/*

[2] L.A. Rowe, D. Harley, P. Pletcher, S. Lawrence. "BIBS: A Lecture Webcasting System." Berkeley Multimedia Research Center, TR 2001-160, June 2001.

[3] Virtual ink corp., Mimio device, [Online]. Available: http://www.mimio.com

[4] e-Beam corp., e-Beam device, [On line]. Available: http://www.e-Beam.com

[5] IBM/LOTUS corp., Sametime collaboration product, [Online].Avaliable: http://www.lotus.com/home.nsf/welcome/sametime