

LECTERN II: A MULTIMEDIA LECTURE CAPTURING AND EDITING SYSTEM

Nikolai Joukov and Tzi-cker Chiueh

Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
{njoukov, chiueh}@cs.sunysb.edu

ABSTRACT

Deployment of asynchronous learning systems is still very limited mainly because of the use of expensive video to record and play back classroom lectures. The key advantage of the Lectern II approach is that all important lecture components can be effectively captured without video. Lectern II employs the touch-sensitive screen technology to build a “digital desk,” which is shown to be able to effectively support and transparently capture the standard classroom lecturing activity. Recorded lectures can be edited and automatically uploaded to a Web server, and then viewed by students via a standard streaming player. As the total cost of a complete Lectern II system is under \$4,000, Lectern II represents the first course lecture recording system that has the potential to be widely deployed in the classrooms of universities and K-12 schools.

1. INTRODUCTION

There is an emerging trend to use the Web as a learning medium for asynchronous learning systems [1]. The benefits of this approach are evident. Students can repeatedly review the lecture material with the pace and at the time that is best suitable for them. Thus, the effectiveness and convenience of classroom lectures can be dramatically increased [2].

However, deployment of such systems is still limited because they typically use digital video as the main delivery medium. Video, while being very expressive, is also very expensive, primarily because of the need of expensive hardware, software, and dedicated personnel during the video recording and editing phases. In addition, the enormous size of the video data streams recorded from lectures creates problems in storage and delivery of the lectures. Thus, the bandwidth requirements seriously restrict the accessibility of the video-based lectures. In particular, they are out of reach to users that are connected to the Internet via ordinary modems. Taking the problems described above into account it is not surprising that most

lecture materials in the Internet are available in a form of lecture slides only, without any voice or annotation information.

The goal of the Lectern project is to minimize the equipment and operational cost associated with course lecture recording and editing by avoiding the use of video capturing. Following the approach described in [3,4] the Lectern II employs the touch-sensitive screen technology to build a “digital desk,” which is shown to be able to effectively capture voice, slides, and pen annotation data. In addition, this way was shown to be effective in supporting the actual presentation process. The key advantage of the Lectern approach is that all important lecture components including slides, voice, and during-lecture remarks can be effectively captured without video.

First generation of the Lectern system [3] has a number of weaknesses. First, it required slides to be prepared as HTML files, and more seriously it assumed that the HTML files look the same with all the possible operation system and browser settings. Second, all system components were implemented for Windows OS only. Finally, it required download of a custom player for the lectures playback.

Lectern II is a completely redesigned system although the general approach of lecture capturing remains the same. Lecture slides used for the presentation can be prepared in any format on the Windows platform. A specially designed virtual printer is used to convert the slides to the internal slide sequence format. The recorded lectures can be played back and edited after the presentation. Lectures are converted into a Web ready format, while all visual information is saved in a rasterized form. The voice, slide changes, and on-screen remarks are synchronized via the Synchronized Multimedia Integration Language [5]. In addition, web server side support is provided in order to simplify the lectures upload, maintenance, and online lecture search. Finally, the recorded presentation can be played back via a modest modem connection using freely available software on most operating systems. Thus, the only action required from the end user is to follow the URL link of interested lecture.

The total hardware cost is estimated to be below \$4000 even including the video projector, which is usually available in most of the lecturing environments. Since no additional infrastructure is required the Lectern system can be set up in almost every classroom.

2. HARDWARE SETUP

Similar to the hardware set up described in [3] Lectern II consists of a high resolution touch-sensitive screen, a PC, a video projector, and a microphone (see Fig. 1). Instructors carry out the entire lecturing activity using the touch sensitive screen, while its contents are mirrored on a big screen via the video projector. A digital pen is used to emphasize important points, draw illustrations and formulas, or simply attract students' attention to the location on the slide that is currently being discussed. Meanwhile, the voice is being recorded through a microphone. The PC and the touch sensitive screen can be substituted with a tablet PC. In this case the whole setup consists of only two pieces and thus can be conveniently carried by the instructor and easily shared among many classrooms.



Fig. 1. Lectern II setup as seen by instructor. Setup consists of touch-sensitive screen, digital pen, microphone, PC, and a video projector.

3. SLIDES PREPARATION

To simplify lecture preparation a special virtual printer driver was created. Thus, the lecture slides can be prepared in any program that can print on a printer on a windows platform. Once the slides are ready instructor can print them on the Lectern printer and the Lectern recording program is activated automatically. In fact, the Lectern printer is an EMF (Enhanced Metafile) [6] printer with a special print processor that extracts the slides from the spool in the vector EMF form. Subsequently, these slides can be rasterized and displayed on almost any system running Windows regarding of the screen resolution. An additional benefit of this approach is that all the textual information in the slide can be extracted and made

available for the search engines. The software drivers are designed for use with Windows NT, Windows XP and Windows 2000. Once generated, the prepared slide sequence can be repeatedly used.

4. LECTURE RECORDING AND EDITING

All the Lectern recording/editing components were written with MS Visual C++ 6.0 using Windows32 API only. One of the main goals of the project was to make the system easy to use while providing all the desirable functionality.

In general, the system operates in one of two modes: recording and editing a recorded lecture. In both cases the toolbar pops up on the side of the screen if a mouse or a pen is moved at the screen edge. Otherwise the toolbar is completely covered by the slide and will not obstruct slide's view.

In the recording mode the instructor can record/pause the lecturing process, choose a pen color, insert a new blank slide or create a copy of the current slide that is clean from pen remarks. As illustrated in Figure 2, a scrollable list of thumbnail slides' images is provided on the same toolbar to ease the slide navigation.

During the editing mode a prerecorded lecture can be played back and edited. The editing operations include cut to clipboard and paste from clipboard or another lecture. All the editing actions can be undone.

After the recording and editing is finished the lecture is saved in the vector format for subsequent editing operations and a set of files is generated and put on the Web through an automatic upload process.

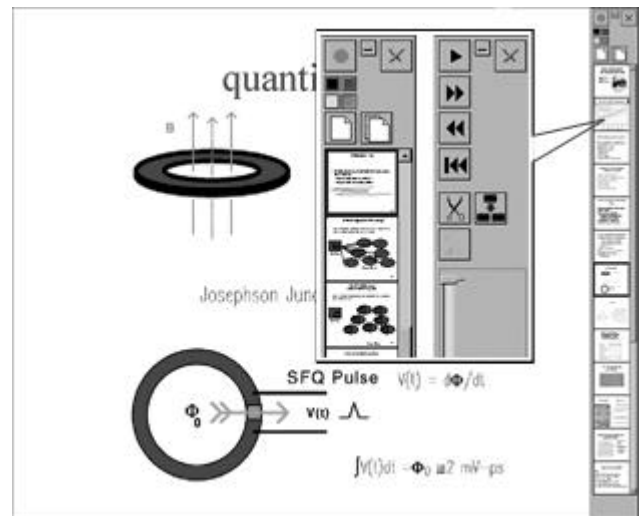


Fig. 2. The main screen of the Lectern II recorder/editor. The toolbar on the right pops up only if the mouse cursor or a pen is over the right edge of the screen. The inset shows the upper portion of the recording (on the left) and editing (on the right) mode toolbars.

4.1. Slides and remarks handling

The slide representation is a list of GDI (Graphics Device Interface) calls in a form of Enhanced Metafile. The pen annotations or remarks are stored as a set of points that are connected with lines when displayed on the screen. Every erase action made by instructor is sampled as a set of fixed size square regions that are required to be cleaned from all previous pen remarks. Thus, erase events are stored as a set of centers of the corresponding square regions. To play back an erase event a set of possibly overlapping square portions of the original slide is copied from the slide image stored in a memory buffer over the screen contents.

To create a Web-ready content all the screen images, remarks and erase regions are stored in a rasterized form. To minimize the size of the images, they are stored in an 8-bit indexed palette mode. The palette is constructed using the popularity algorithm [7]. Pen remarks are stored as a two color transparent images of the corresponding pen-created line segments. The erase regions are stored as portions of the original slide image over which the pen eraser was applied. All the files are compressed and saved in Portable Network Graphics (PNG) [8] format using the libpng library. The playback time composition of the generated images is illustrated in Figure 3.

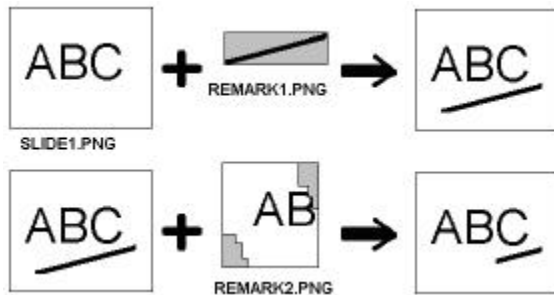


Fig. 3. Slides and pen remarks images composition. Top: a transparent remark image is put over a slide image. Bottom: putting a portion of the original slide over the previously composed image erases a part of the previously drawn remark. Erase region image is composed of several square fragments of the original slide. Grey color denotes transparency in this figure.

4.2. Voice handling

Voice recording and playback is realized through the Windows Media Control Interface (MCI) [9]. All other voice related operations including cut and paste are done directly on the ADPCM voice data. The voice is sampled at 11,025 8-bit samples per second rate that is well above the human voice frequency range. Unfortunately, the Windows MCI has a problem: the recorded voice stream duration is longer than the voice recording time. Thus, the

recorded voice is slightly stretched. The amount of stretching depends on computer performance, but the problem can not be ignored because even on a 2.2 GHz PC with 512 MB of RAM there is a minute long lag of speech behind the presentation process at the end of a 2.5 hours long lecture. To solve the problem the MCI recorded voice data is resampled afterwards.

In order to minimize the streaming playback bandwidth requirements the voice is compressed using the Mpeg 2.5 Layer 3 format by the external Lame [10] program.

5. STREAMING PLAYBACK

5.1. Lecture representation

The Real Player [11] from the RealNetworks was chosen to be the recorded lectures player. The major reasons for this decision are that the player is free, available for all most popular operation systems, already installed by many users, and most importantly, provides support for the streaming playback of images. For this purpose the Real Player supports the RealPix (RP) file format [12]. Basically, this format allows specifying the screen location and time for a particular picture to be displayed. As illustrated in the Figure 3, the pen remarks and erase regions are displayed over the background slide images at the times when they were put on the screen during the actual lecture.

Most importantly the RealPix format allows specifying the sizes for all the images to be played so that the Real Player itself can calculate the bandwidth utilization in advance and preload the data accordingly. This functionality can be further enhanced if a special streaming server [13] from the RealNetworks is used to send the data. The voice and the RealPix presentations are in turn synchronized by the means of Synchronized Multimedia Integration Language [5]. Figure 4 represents the logical dependence among the file formats used.

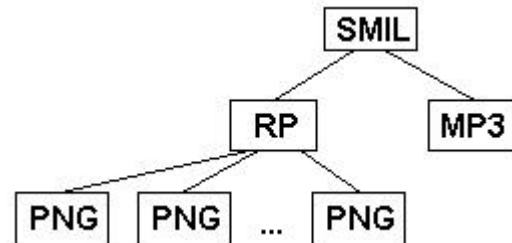


Fig. 4. Slides, pen remarks and erase regions are saved in the Portable Network Graphics (PNG) format. Their onscreen positions and display times are listed in the RealPix (RP) file. The Synchronized Multimedia Integration Language (SMIL) synchronizes RP timeline and voice saved in MP3 form.

5.2. Lectures statistics

Below is the information about the set of files generated from three lectures on “Wireless LAN-based Systems and Applications,” which is an advanced course in the Computer Science Department of SUNY at Stony Brook.

Lecture duration (minutes):	77	99	118
Total files generated:	539	429	1017
Total slides presented:	44	51	17
Total size (KB):	9,930	13,172	14,902
MP3 file size (KB):	9,262	12,158	14,314
Total slide images size (KB):	512	880	178
RP plus SMIL file size (KB):	70	60	226

From the above statistics it is easy to figure out that the average bandwidth required is under 18 Kbit, which is well below the speed of even a modest modem. The voice data accounts for more than 90% of the whole data volume. Therefore, all the images despite being saved in a rasterized form occupy less than 10% of the whole lecture. As soon as SMIL and RealPix files are loaded the Real Player can start the files download according to their position and size so as to make the bandwidth requirement as close to the average as possible. The typical startup latency is on the order of several seconds.

5.3. Synchronization skew

In addition to the MCI voice time stretching problem described above, there is an additional source of potential voice data lag behind the rest of the lecture data. It turned out that some existing MP3 players stretch the voice sampled at the rate proportional to 11.025 KHZ for about 7 seconds per 25 minutes. This problem was observed with Real Players for Linux and for Windows as well as with the Quick Time [14] player. Windows Media Player [15] did not have this problem.

6. BACKEND SERVER

To simplify the recorded lecture files maintenance a special Perl-based set of scripts was implemented. The use of Perl allows the server to be platform independent. The scripts are responsible for the communication with the Lectern main program in the uploading process, placing the files in the appropriate directories and maintaining the set of documents created for the use of the Web search engines. The textual information extracted from the slides is placed into the HTML files that are made accessible to the Web crawlers and thus for the majority of the Web search engines. The HTML files, in turn, have references to the lectures that contain corresponding slides.

7. CONCLUSION

Lectern II is an easy to use, inexpensive lecture capturing system. The system not only allows the lectures to be recorded but also creates an enhanced classroom-lecturing environment. The recorded lectures can be edited and uploaded to a Web server, possibly enhanced to simplify the lecture storage maintenance. The textual information from the slides is accessible for standard search engines. The recorded material can be played back using a free player via a modest modem link. Initial evaluation show that students and instructors found the system to be extremely useful and helpful.

8. REFERENCES

- [1] A. Radford, “The future of multimedia in education,” *First Monday*, Vol. 2, No. 11, 1997
- [2] J. Dutton, M. Dutton, and J. Perry, “Do Online Students Perform as Well as Lecture Students?” *Journal of Engineering Education*, Vol. 90, No. 1, pp. 131-136, January 2001.
- [3] T. Chiueh, P. Deng, “Lectern: A Digital Desk System for Video-Free Course-Lecture Capturing and Playback,” *proc. IEEE Multimedia*, New York City, vol. 1, pp. 3-6, July 2000.
- [4] T. Chiueh, W. Wu, L. Lam, “Variorum: A Multimedia-Based Program Documentation System,” *proc. IEEE Multimedia*, New York City, vol. 1, pp. 155-158, July 2000.
- [5] Synchronized Multimedia Integration Language specification, <http://www.w3c.org/TR/smil20/>
- [6] Enhanced Metafiles, Microsoft Windows SDK
- [7] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, 2nd Ed., 1996.
- [8] Libpng library, <http://www.libpng.org/>
- [9] Media Control Interface, Microsoft Windows SDK
- [10] LAME project, <http://lame.sourceforge.net/>
- [11] Real One player, <http://www.realone.com/>
- [12] RealPix specification, <http://service.real.com/help/library/guides/realpix/htmlfiles/notice.htm>
- [13] Helix Universal Server home page, <http://www.realnetworks.com/products/server/index.html>
- [14] Quick Time Player, <http://www.apple.com/quicktime/>
- [15] Windows Media Player, <http://windowsmedia.com>