

# Perspectives for Lecture Videos\*

Michael Hartle, Henning Bär, Christoph Trompler, and Guido Rößling

Darmstadt University of Technology, Department of Computer Science,  
Hochschulstr. 10, 64289 Darmstadt, Germany  
{mhartle,hcbaer,trompler,roessling}@informatik.tu-darmstadt.de  
<http://www.dlh.informatik.tu-darmstadt.de>

**Abstract.** This article presents an architecture for automated multi-perspective lecture recordings. The implementation switches between several video streams showing different perspectives in order to make the recording more vivid and to reduce the tunnel vision effect of single-perspective recordings. Automatic switching can be based on time intervals or, at a later stage of development, via simple recording rules.

## 1 Introduction

Recently, lecture recordings are increasingly put online in order to support the students with their examination preparations. This is also the goal of the OpenCourseWare Project of the MIT [1].

A survey conducted at the Oxford Brooks University [2] has shown the strong interest of students in such an offering. Our own surveys among more than 500 students yielded the result that more than half of them considered lecture recordings as helpful. Yet, our own experience demonstrates that the acceptance is based on the quality of the material presented. Reductions in quality, e.g. if slides are not readable or the lecturer leaves the field of view being recorded, result in a notable drop of acceptance.

Lecture recordings containing an occasional change of perspective are more vivid for the viewer than a video stream fixed on the lecturer. Students also obtain an impression on the surroundings of the lecture if the recording also includes scenes of the audience present.

For the multi-perspective recording system presented in this article, a lecture hall was equipped with several cameras. These cameras can be operated remotely and can be turned and pitched. Furthermore, the lecture hall was also provided with a recording server for the video streams.

After a brief overview of related systems, we describe the processing steps, our recording approach and post-processing operations.

## 2 Related Work

At the University of Wisconsin - Madison [3], the lecture recordings are costly post-processed, based on knowledge about following scenes and digital zooming.

---

\* The underlying project for this report was supported by the German Federal Ministry of Education and Research under code 08NM208. The responsibility for the content of this publication lies with the respective authors

Microsoft [4] presents a system which localizes both the lecturer during his presentation and even students posing questions in order to direct the cameras automatically. The publication uses the advantages of a non-intrusive localization method which is also a research topic pursued in our research group. In [5], a localization method is described using an additional camera.

In [6], a system is presented which synchronizes video streams with provided slides. Another goal is the improvement of the quality of slides recorded via the video stream. Therefore, the actual lecture recording is generated in a post-processing step.

At the University of Queensland [7], several multimedia streams are synchronized and offered as a single document. This system is not limited to video streams and slides, but can also incorporate other media such as texts, pictures or images.

Several additional recording systems are being presented in [8]. Among other systems, LECTURNITY (<http://www.lecturnity.de>) and the MS PRODUCER (<http://www.microsoft.com/office/powerpoint/producer>) are installed on a computer system also used for presentation which simplifies a mobile deployment. However, a fixed installation with support for changes of perspective and tracking of lecturers is complicated by using both the recording and presentation system on the same computer.

The FAME Room [9] is a meeting room at Karlsruhe, Germany, where the speaker is located via audio-tracking. A digital "lecture assistant" is employed which handles other supporting tasks besides the audio-tracking such as controlling light levels and audio volume.

Typical recording systems are limited to a proprietary format such as Real Media or Microsoft PowerPoint. This severely limits new innovative methods of presentations offered by using open media format alternatives. Furthermore, some systems require additional hardware such as separate cameras needed for localization. Finally, necessary manual post-processing does not scale adequately for practical applications. Many publications do not consider the automation of this process which is essential for keeping the required manual amount of work to a minimum for a large number of recordings.

### 3 Management of the Recording Process

The recording process of our system can be described as a composition of several steps. This section regards the control of the recording process. Section 4 describes the actual recording, followed by the post-processing steps in Section 5. As the emphasis of a slide-based presentation is placed on the readability of the slides, these are captured separately and combined with the video stream. Further details regarding the required technology are presented in Section 6.

The recording is started and stopped manually by an operator. The functionality can also be available on the presentation system, so the lecturer is also able to control the process. The editing process creates the resulting video file which covers the whole lecture.

For starting the recording process, a management application is provided which controls the recording server via local-area network. The operator, either the lecturer himself or an assistant, clicks on the *Start Preview* button. This results in a signal for

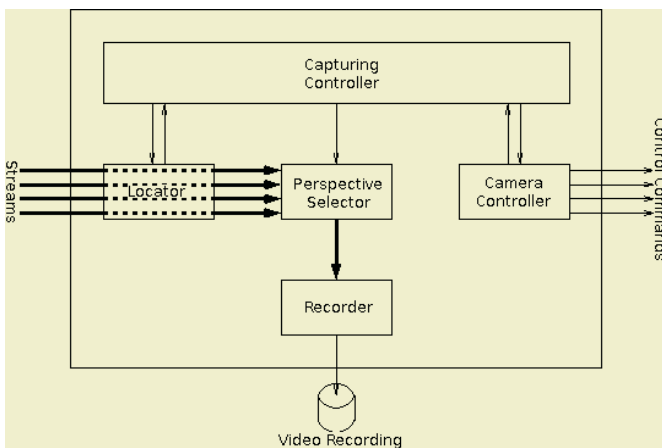
the recording server to send a video preview as a live stream. Using the preview, the setup can be tested, e.g. whether the audio signal is being properly transmitted from the lecturer's wireless microphone, or whether the correct field of view or the correct perspective is being recorded.

When the lecture begins, the *Start Section* button of the management application has to be triggered, which again sends a signal to the recording server. The server stores the current time into a configuration file necessary for the later post-processing. The lecture recording can be paused by using the respective buttons of the management application.

Until the end of the lecture, the name of the lecture to be associated with the recording can be chosen from a list. At the end of the lecture, the *Stop Lecture* button has to be triggered and the recording server is ordered to finalize the configuration file for the post-processing step and to copy the recorded video stream to a central server. On this server, incoming recordings are accepted via a web service and automatically sorted based on date, lecture name and a unique ID of the recording. Stored on the central server, the recordings are held available for post-production via the video editing application.

## 4 Real-Time Video Recording

The recording system was developed based on the architecture shown in Figure 1. The *Capturing Controller* is responsible for the generation of the multi-perspective video. It controls the other modules depending on the necessary steps to be performed. The *Camera Controller* handles commands for turning and pitching the cameras ordered by the Capturing Controller. The *Perspective Selector* changes the perspective when a corresponding command was issued by the Capturing Controller. The *Locator* starts the search for the lecturer only on request based on the camera that was marked as available by the Camera Controller. The *Recorder* is a standard video recording application.



**Fig. 1.** Recording architecture

Achieving a lecture recording of acceptable quality requires a good integration and cooperation of these modules. They can be installed as a single software package on one machine or distributed over several machines, depending on the requirements of the available infrastructure. The communication between the modules is based on web services as an open protocol in order to facilitate the modularity and exchangeability of the implementation of entire modules. Some of these modules are written in Java, which allows modules to be utilized on multiple platforms. For example, the Java-based Camera Controller in our infrastructure is running on a Linux system, but can also easily be deployed on a Microsoft Windows system.

The following sub-sections explain these modules in detail.

#### 4.1 Capturing Controller

The Capture Controller is responsible for all scenes and for the choice of an appropriate perspective for each scene, where a scene is defined as the sequence between two changes of perspective. The length of a scene varies in accordance with the propositions of [4] between several minutes for perspectives showing the lecturer and several seconds for scenes showing the audience. Typically, neither panning nor zooming is performed while a scene is recorded, although there are constellation-specific exceptions. The Capturing Controller manages all other modules. It can order the Locator to localize the lecturer using a separate camera. Using the result of the Locator, the Capturing Controller requests the Camera Controller to point to and zoom into the close-up range of the lecturer with an available camera. Finally, it uses the Perspective Selector to switch the perspective to the new prepared camera.

This example is a typical constellation generating one of many imaginable recordings. The following constellations are examples which can be of use for a lecture recording:

- Close-up recordings of the lecturer
- Overview of the lecture hall centered on the lecturer
- Slow panning over the audience
- Overview of the audience

#### 4.2 Camera Controller

The cameras installed in the lecture halls can be controlled regarding pan, tilt and zoom through a serial interface. The instruction set allows changing the horizontal and vertical angles as well as the zoom for a camera.

Although this instruction set is quite helpful, managing several cameras on this basis is difficult. The same direction of different cameras leads to completely different views. Thus we prefer cartesian coordinates for managing the camera positioning. As a prerequisite, we first have to measure the positions of the cameras after the camera installation, because the absolute cartesian coordinates of a camera target have to be computed into corresponding angle settings for each camera involved.

The Camera Controller internally operates with different spherical coordinate systems, where each system's center is defined by a camera. Using cartesian coordinates is easier to work with especially when describing planar surfaces like the black board

or the walls. So an easy command offered by the interface is *lookAt*, which directs the camera to a desired target.

Close-up views, overviews and localization are examples for different standard settings, which are necessary for a recording. The Camera Controller knows several of such settings, like *setCloseRec*, which offers a close-up view of the lecturer.

Currently this control is performed manually. The network interfaces allow an easy replacement of the calls from a graphical user interfaces by those of an automated process. A diploma thesis is being written about this automatic Camera Controller.

### 4.3 Locator and Perspective Selector

The Locator is needed when the Capturing Controller decides to record a close-up view of the lecturer. The Capturing Controller first chooses a camera for the localization and one for the recording. The current status of development requires setting the localization camera to a predefined localization position. The Locator is then ordered to find the lecturer with the available camera. In a lecture hall, the lecturer usually moves in an area close to the black board or the presentation surface. The Locator only takes this area into account. Other areas like the auditorium which are most likely irrelevant for the localization of the lecturer can be blanked out. For the remaining area, a simple comparison of adjacent frames is sufficient to figure out those areas where movement takes place. The Locator assumes that the lecturer most likely caused those movements and thus can be found at that position. The quality of the Locator has been proven in first tests. In the next term, the Locator will be tested in first real lectures.

During recording, the Perspective Selector chooses the digital video streams which shall be made visible. It offers the currently selected stream as a virtual capture device to the operating system, which can then be used by further modules.

### 4.6 Recorder

Basically any recording software can be chosen for the encoding and storage of the video and audio signals. To eventually synchronize the slides with the video more easily, we customized the recording software so it writes the time at which the recording was started to a file beneath the actual video. This is not strictly required, but makes the manual synchronization with the slides much easier. The procedure of cutting the video depends very much on the recorded format. In our installation, we use *mpeg4ip* [10] to record our video, because

- *mpeg4ip* is available as open-source project, which allowed us to modify it so that the time can be stored at which the recording was started;
- *mpeg4ip* provides us with a suited video codec and file format on the basis of an international normalized standard, which can easily be post-processed with existing software products;
- *mpeg4ip* can generate both local files and video streams which can be transmitted to double-check the recording while it is in progress.

## 5 Post-processing the Videos

All presented slides have to be included in these recordings to enable students to properly prepare for examinations. They are visible as fixed images. We use a presentation environment [11] to create those fixed images, with which images are created out of the slides no matter which presentation software is being used. The file names of the images are based on the time when the according slide was shown.

The creation of the lecture recordings is done in two steps. First, the video is being cut, then both the cut video and the slide images are being referenced by a SMIL [13] file in accordance with their interplay during the lecture.

The editing of the videos is done via Apple's QuickTime SDK [14]. The video is cut to properly match the beginning and ending of the recorded lecture and to remove all sections of the lecture where the cameras provided a video stream, yet the recording was paused via the management application. Finally the edited video stream is put back together without any re-encoding taking place and thus without additional quality loss or computational expenses. The information at which time the cuts are planned can be taken out of the configuration file which is generated by serializing events like starting, stopping, and pausing the recording.

The changing slide images are created in an independent fashion, as mentioned at the beginning of this chapter, and thus are available separately. The video can be referenced directly in the SMIL file, which connects the edited video stream and the presented slides. The synchronization of both media is gathered on the basis of the slide presentation time encoded in the file name of each slide and the starting time of the video. The beginning and ending points of time of the video stream are obtained from the configuration file.

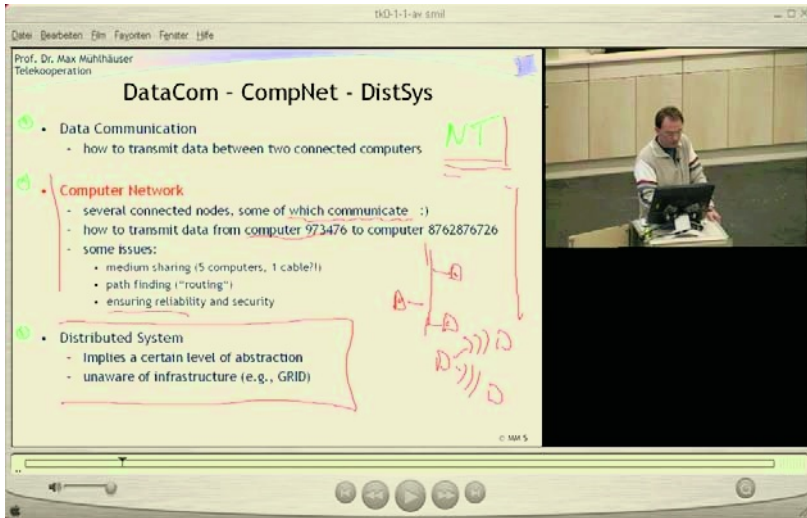
The SMIL file can be downloaded from a dynamically generated website. So we realized an interface to the interaction software TVremote [12], which allows students to use their mobile devices during the lecture to tag a section for later video review. They can access a list of their personal tags and start the lecture video by clicking on one of those tags.

An example of a SMIL-based lecture recording combining and synchronizing separate fixed images and video stream can be seen in Figure 2, shown with Apple's QuickTime Player.

## 6 Practical Operation of the System

We have been offering lecture recordings to our students for more than two years. In the beginning, the cameras had to be controlled manually. Since the winter term 2003/2004, our multi-perspective system is used practically and the cameras can be controlled from remote. That is a great assistance, as controlling the cameras can even be delegated to trained student helpers without in-depth technical knowledge. In the winter term 2002/2003, a poll of the didactical team of our university found out that more than 50% of the students experienced the recordings as helpful. At that time, the recordings were made using only a single perspective. Members of the didactical team recommended including changes of the perspective in our systems to offer an

overall visual impression of the lecture hall. In winter term 2003/2004, lecture videos were made using changes of the perspective. The result of a poll thereafter showed that more than 70% of the students found the recordings partially or very helpful for the test.



**Fig. 2.** A lecture recording ready for distribution among students. The hand-written annotations on the slide were added during the course of the lecture

## 7 Outlook

The work of recent years focused on the automation of the described recording process in order to increase the amount of lecture recordings without increasing the workload for the staff involved. Further works are intended to increase the quality and the cooperation of the modules presented in section 4.

Several features and ideas have led to an investigation of solutions based on Apple's QuickTime file format [15] as an alternative to the currently used SMIL-based system. Solutions for goals such as interactive content-based navigation or the combination of both fixed images and edited video stream into a single file are currently investigated. Concerning these features, the QuickTime specification seems to provide promising approaches.

Although we are investigating these solutions, no stable and satisfactory results regarding the creation and playback of such content are currently at hand. With their availability, these results are expected to be the subject of a future publication.

## References

1. MIT OpenCourseWare – Fact Sheet. Online at <http://web.mit.edu/newsoffice/2001/ocw-facts.html>. (2001) (seen on July 12, 2004)

2. Phillimore, R.: Face to Face or eContent: Student and Staff Perspective. In: Proceedings of the International Conference on Computers in Education. IEEE Press (2002) 211-212
3. Gleicher, M., Heck, R., und Wallick, M.: A Framework for Virtual Videography. In: Proceedings of the 2nd International Symposium on Smart Graphics. ACM Press, New York (2002) 9-16
4. Rui, Y., Gupta, A. und Grudin, J.: Videography for Telepresentations. In: Proceedings of the conference on Human Factors in Computing Systems. ACM Press, New York (2003) 457-464
5. Rui, Y., He, L., Gupta, A. und Liu, Q.: Building an Intelligent Camera Management System. In: Proceedings of the 9th ACM International Conference on Multimedia. ACM Press, New York (2001) 2-11
6. Mukhopadhyay, S. and Smith, B.: Passive Capture and Structuring of Lectures. In: Proceedings of ACM Multimedia. ACM Press, New York (1999) 277-287
7. James, D., Hunter, J.: A Streamlined System for Building Online Presentation Archives using SMIL. In: Proceedings of the Australasian conference on Computing Education. ACM Press, New York (2000) 145-152
8. Lauer, T., Ottmann, T.: Means and Methods in Automatic Courseware Production: Experience and Technical Challenges. In: Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2002). AACE Press, Charlottesville, VA (2002)
9. Rogina, I., Schaaf, T.: Lecture and Presentations Tracking in an Intelligent Meeting Room. In: Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02). IEEE Press (2002) 47-52
10. MPEG4IP – Open Streaming Video and Audio. Online at <http://mpeg4ip.net>. (seen on February 8, 2004)
11. Rößling, G., Trompler, C., Mühlhäuser, M., Köbler, S. und Wolf, S.: Enhancing Classroom Lectures with Digital Sliding Blackboards. In: Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2004), Leeds, UK. ACM Press, New York (2004) 218-222
12. Bär, H., Rößling, G., Mühlhäuser, M.: Improving Interaction During Lectures: A Minimal-Distraction Approach. In: Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-MEDIA). AACE Press, Charlottesville, VA (2004) 1250-1255
13. W3C Synchronized Multimedia Homepage. Online at <http://www.w3.org/AudioVideo/> (seen on May 30, 2005)
14. Apple QuickTime Developer Connection. Online at <http://developer.apple.com/quicktime/> (seen on May 30, 2005)
15. Apple QuickTime Developer Connection, QuickTime File Format. Online at <http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf> (seen on May 30, 2005)