A question managing suite for automatic lecture recording

Fleming Lampi, Hendrik Lemelson, Stephan Kopf and Wolfgang Effelsberg
Department of Computer Science IV, University of Mannheim, Mannheim, Germany

Abstract

Purpose – The purpose of this paper is describing the seamless integration of the question-answer interaction into automatic lecture recordings (ALRs). This includes the design and implementation of the question management (QM) software for a virtual camera team.

Design/methodology/approach – Coming from the human role model the interaction and its management to the virtual world is transferred and integrated it into a virtual camera team. All events are translated into sensor inputs which get processed by the virtual director and are used for the collaboration of the team in order to implement more complex cinematographic rules.

Findings – It is found that it is possible to record the whole interaction, to record the original voice of the questioner out of an audience without handing out a microphone or forcing him/her to walk to one, and to record a video of the questioner while asking. So, it is easier to follow a lecture recording as more details are recorded automatically.

Practical implications – First experiences on using this software clearly show the small weaknesses of the first version. As mentioned in the outlook, these are currently being addressed, e.g. by looking for a more natural interface.

Originality/value – The paper demonstrates how to use the question-answer interaction as sensor input for an automatic lecture recording (ALR) system based on the roles of the according human originals. It ensures that many details of a lecture can be recorded seamlessly to keep the lecture context continuous and therefore to make the lecture recording more vivid and interesting.

Keywords E-learning, Lectures, Learning methods

Paper type Research paper

1. Introduction

Nowadays, lecture recordings are very widely accepted as they are easy to achieve (Lauer and Ottmann, 2002). This is especially true if only the slides and the lecturer's speech are recorded. Unfortunately, these recordings tend to be boring, independent of how fascinating the original session was. It is hard for students to keep focused on the topic while watching the recordings.

A main reason why students become uninterested after a short time comes from our habit of watching television content produced at a very high quality level. This has pushed our expectations, and we expect the same quality for lecture recordings. By using cinematographic rules, television is able to keep content interesting. Unfortunately, camera teams necessary for such a production consist of a large number of people. It is far too expensive for universities to hire a real camera team to record lectures, especially when they are obliged to save money. A solution to this problem is to develop an automated system.

Some work on related topics, from the automatic recording in video surveillance up to the recording of presentations and meetings, has been done prior to our efforts. A first example for video surveillance using multiple cameras, including Pan-Tilt-Zoom-Cameras, to track objects in a 3D-virtual world is presented in (Hampapur *et al.*, 2005). Meeting recordings, e.g. captured by a 360° camera to track the individuals, and microphone arrays or human annotations to track the speaking person, like in (Rui *et al.*,

Interactive Technology and Smart Education
Vol. 6 No. 2, 2009
pp. 108-118

1741-5659
DOI 10.1108/17415650910968116

2001a) and eventually amended by different cameras for the different video sources (Cutler *et al.*, 2002), focus on smaller groups and do not use cinematographic rules. Transferring such an approach to larger rooms as it was done by Siemens Corporate Research is more similar to surveillance videos than to lecture recordings (Huang *et al.*, 1998), because even though reacting on detected events or faces, it only shows the according part of the room and does not use these events as an input for cinematographic rules. Therefore, these earlier approaches are completely different from our system.

Closer to our approach is the use of pan and tilt operations and image processing for framing and following the lecturer. A well-known sample application is "AutoAuditorium" (Bianchi, 1998), using a basic level of automatic presentation recording but without any cinematographic rules. Another example was built by Microsoft Research (Rui *et al.*, 2001b) and has been improved in (Zhang *et al.*, 2005). While Microsoft is working towards watch ability without subject knowledge, our focus is to implement more complex cinematographic rules in order to keep the recordings more vivid. Microsoft's and our approaches are somewhat similar as we both use multiple cameras, virtual camera operators and virtual directors based on finite state machines (FSM). But, unlike our model, Microsoft uses a scripting language to rewrite shot determining rules in note form, implying fixed durations of the shots. This leads to predetermined transitions which we consider undesirable as they distract the interest from the topic to the awaiting of the next transition. To be more precise, our approach does not use fixed rules to determine a transition of the FSM, but describes a selection of possible transitions which get weighted based on the input of different sensors as reaction to the environment. The resulting transitions are therefore always similar, but seldom identical. Of course, the number and the type of employed sensors can vary in order to take future developments into account.

While Bianchi's approach only uses image processing to track the lecturer and to determine the image framing, Microsoft uses hanging microphones and sound triangulation to locate questioners. Our approach also uses sensors to identify the positions of the students in the room and can be amended easily with sound triangulation as well. As we use a microphone per student to record the question, our approach is less sensitive to room noise. Determining a questioner's position is the prerequisite not only to focus a camera on him/her, but to implement more sophisticated cinematographic rules. For example, if a student asks a question, the lecturer and the student may be framed in such a way that they face each other while the system switches between their shots. Similar basic rules have been proposed by (He *et al.*, 1996) for the recording of real-time applications.

In our approach, we identify the different roles of a human camera team, their tasks and their collaboration and use this information to build an automatic, distributed computer system for the recording of lectures. Distinctive features of our new approach are the collaboration of different virtual team modules to implement cinematographic rules, an easily exchangeable FSM to cope with different purposes, and its transitions having variable probabilities and reacting on sensor inputs.

Section 2 gives a brief overview of the automatic lecture recording (ALR) system. More details can be found in (Lampi *et al.*, 2008). In previous work we have focused on transferring the roles of a human camera team to our virtual system. In the case of lectures we have to consider the students in their role as questioners. This paper specifically presents our approach to take the interactions between students and the lecturer into account. In section 3, we examine this interaction and transfer it into the world of our virtual camera team. Section 4 shows how the software manages questions and answers and presents the question manager software suite in detail. In section 5 we present our first experiences with the system before summarizing our work and giving an outlook on the next steps in the last section.

2. ALR

Our ALR software is implemented as a distributed system. The modules are based on the roles of a human camera team and consist of a director module, a cameraman module and a sensor tools module. Figure 1 gives an overview, in which the modules not yet fully implemented are shaded in grey.

Derived from real recordings, the director plays the central role, communicates with all modules and decides which camera goes on air. The director is based on an extended FSM with three contexts: a lecture context, a question context and an answer context. This gives us the advantage that states showing the same shot but located in a different context can be reached by different transitions, and therefore the conditions triggering a transition may differ. The firing of these triggers is based on a variable probability depending on sensor input. When the duration of the previous shot ends the transition to the next shot has to be selected.

Our approach starts by determining all transitions going out from the currently active state and initializing their probabilities. During the next steps, these probabilities are modified: They are decreased first depending on how recently the possible new state was active. Then the sensor inputs are evaluated; they influence the transition probabilities, e.g. if a sensor detects that someone wants to ask a question. Also, we take the visual activity in each camera output into account.

The transition with the highest probability is selected. Additionally, the duration of the next shot is determined based on a time interval individually assigned to each state. The basic rule is that shots of complex scenes get a longer duration than scenes with less complexity. The final duration of a shot lies within this time range. If necessary, time can be added to the duration depending on the motion rate in the scene. These measures guarantee that the behaviour of the director module is always similar in similar situations but usually not identical.

In addition to the director, we analyzed the tasks of a cameraman in order to transfer them into the virtual world as a software module. We discovered three major aspects of

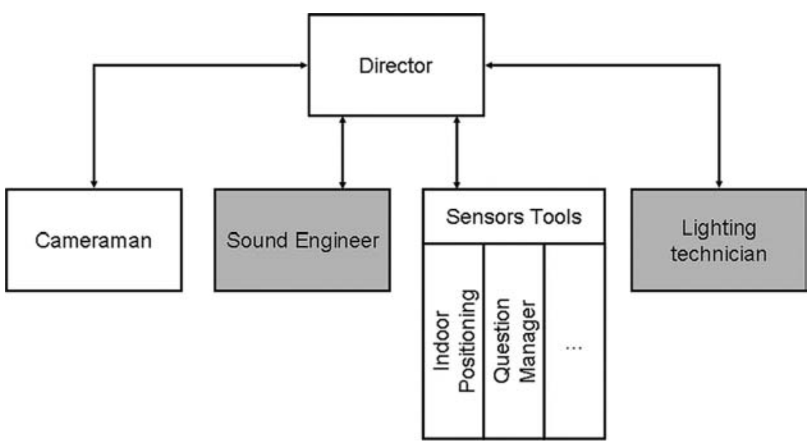

Figure 1.
Overview of the system's modules

a cameraman's job and implemented them: a technical, an aesthetic and a communicative aspect. Aiming at a target, adjusting the iris and drawing the focus are the main activities of the technical part. As shown in Figure 2, we can describe it as a control-loop executed repeatedly during the whole recording.

The communicative aspect is crucial as it is the prerequisite for teamwork. A human camera team communicates over an intercom system; our virtual equivalent uses XML messages over TCP. The cameraman gets the orders from the director, e.g. what to show and how to frame. At the same time, it reports the motion rate in the image and the camera status, which are used as camera sensor inputs by the director module.

Finally, we implemented the aesthetic aspect of a cameraman's job, e.g. how to frame a scene or which zoom factor to use. In contrast to the human role model, this aspect cannot be realized by the cameraman module alone but relies on additional information pre-processed by the director module. A typical example of such information is the set of coordinates where to aim next. Sometimes the next position can be estimated by tracking a person, but e.g. in case of questions from the audience, it is very difficult to calculate it by using image processing only.

This is where the sensor tools module that analyses the physical environment comes into action. Up to now this module includes two parts, an indoor positioning system and a question manager software suite. Additional sensor parts can be added, depending on the development of new sensors and algorithms and on their necessity. Generally spoken these parts provide environmental information to the virtual director. In the following, we will focus on the human interaction in a typical lecture and how we transfer it to the virtual world of the ALR.

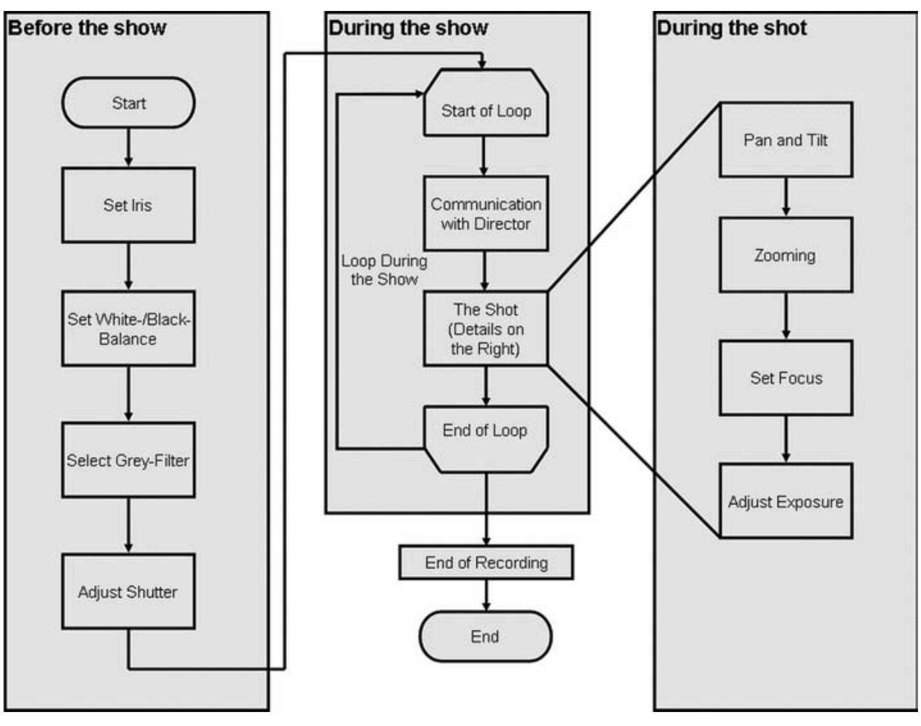

Figure 2.
Flowchart of a cameraman's job

3. Questions and answers

In a standard lecture, questions are important in many ways. Posing questions shows that students pay attention to the lecture. Moreover, it often encourages others to think about the question as well as about the topic itself. Finally, it gives feedback to the lecturer about the level of understanding among the audience.

Let us look in detail on the workflow of a typical question and answer session during a lecture. In the first phase, a questioner raises his/her hand, tries to draw the attention and waits for being given the floor by the lecturer. In the next phase, the question is asked, and then the lecturer answers. In many cases, the interaction is over now. But sometimes either the questioner does not understand the answer at the first try or additional questions about other details arise, so that the answer does not meet the questioner's expectations. Under such circumstances the questioner intervenes and insists on getting a better answer (third phase). The standard two-phase sequence is visualized by the interaction diagraph part between questioner and lecturer of Figure 3.

Even if it seems that question and answer sessions can be easily described by this interaction diagram, there are several possibilities how such a scenario can evolve. Just think of a discussion starting with a single question; in the best case it is a highly fruitful discussion. On the other end of the scale, it can lead into a kind of quarrelling which the lecturer has to stop soon. In between, there are many different possibilities involving a number of different people or leading into any direction or detail of topics.

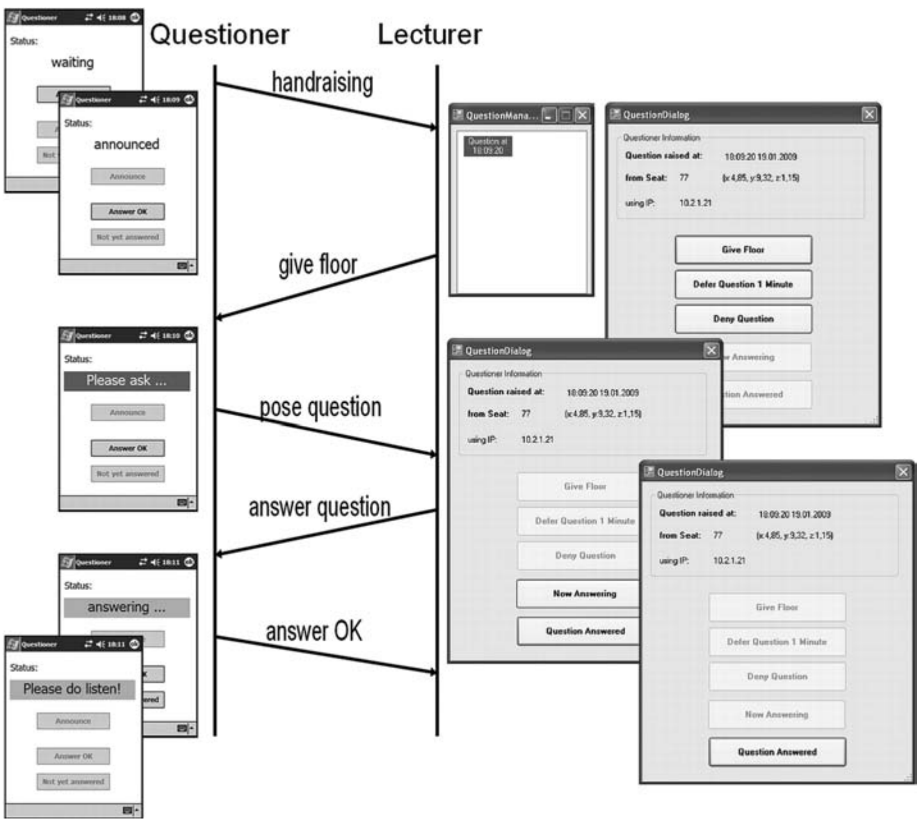

Figure 3.
Question-answer interaction amended with client screenshots

Not only in the worst scenario a lecturer may have a good reason to stop the interaction, e.g. for didactic reasons to keep the planned order of details, for giving the floor to another questioner, or for starting another explanation of the relevant part.

The whole interaction is mostly done by speaking, so it is really important that the audio is clearly audible. In contrast to the common lecture habit not only the voice of the lecturer, but additionally the voice of the questioner needs to be understandable to enable others to keep up with the context. While a lecturer is used speaking loudly and clearly a questioner may need to be encouraged to do so.

Let us now transfer these real-world observations into the virtual world of our ALR system.

4. Question management

We implemented four components in order to bring the question and answer interaction from the standard lecture into a distributed software solution:

(1) a mechanism supervising the interaction between lecturer and questioner;

(2) a component recording the questioner's audio and mixing it with the audio of the entire recording;

(3) a communication interface using the events of the interaction as sensor inputs for the virtual director; and

(4) a module to determine the questioner's position and transmitting it to the virtual director.

As the software may have to manage more than one hand raising at the same time it is more than just an implementation of the interaction workflow but a complex question management (QM) software. It consists of at least three parts, one for the lecturer, one for the questioners and one for the communication with the virtual director. This makes it a distributed software suite. As the lecturer already uses a computer to present slides, and another computer is used for the virtual director, it suggests itself to use these machines to run the according part of the software suite on them. In order to equip the potential questioners in the room with electronic devices, we use personal digital assistants (PDAs) which we employ anyway for interactive quizzes during the lecture (Scheele *et al.*, 2005; Kopf *et al.*, 2007). They provide wireless LAN (802.11) access, a built-in microphone and a pen-operated touch screen. These three features are used to implement the questioner's client for the QM software.

In the next section we present the details of the three parts our QM suite consists of:

4.1 Questioner's client

The questioner's client, on the PDA, fulfils three purposes: The first purpose is to estimate the user's current position within the lecture hall which is necessary for the virtual cameraman to know where to aim next. To achieve this task, we use an advanced indoor positioning system based on 802.11 and fingerprinting (King *et al.*, 2007), developed in our team. Estimating a user's position with fingerprinting is split up into two phases: the training phase and the position determination phase. During the training phase fingerprints are created for several reference positions within the area of interest, and then they are stored in a database. Such a fingerprint basically consists of the signal strength of every single access point receivable at this position. So, it represents the unique properties of the signal space at that specific position. In our case, we created one fingerprint for each seat in the lecture hall.

In the position determination phase, the user's device measures the signal strengths of each receivable access point at its current position. These values are then matched against the fingerprints in the database. The position of the fingerprint that offers the best match to the currently collected data is returned as a position estimate. As position estimation based on 802.11 fingerprints offers a positioning accuracy of approximately two meters on the average. This accuracy is not optimal for our case because its granularity covers up to three seats. We reduce this problem by defining the maximal camera zoom to show at least these seats. There is no necessity to show a questioner centered as long he/she is completely visible.

The second purpose of the questioner's client is to provide the user interface for the questioner. After the initialization the software shows the position estimation as a coloured area which the user can confirm. The client connects to the server and presents the user interface. On top, there is a status message box. It informs the questioner at any time about the status of the interaction, controlled by questioner and lecturer. Of course, the lecturer can override a questioner in order to keep the lecture going. The status messages shown in the box change according to the interaction from "waiting", "announced", over "please ask" to "answering". But in some cases there are different messages to read e.g. "sorry, not possible" if the question was blocked by the server, "just a minute" if it was deferred by the lecturer who will be automatically reminded after one minute, or "please do listen" if the lecturer denies the question.

The third purpose of the client is to sample the audio and transfer it to the server. As the PDAs are typically within one's arm reach, they are in a distance well suited for audio recordings. The audio transmitting instance is created when the questioner announces a question and starts transferring the audio data over wireless LAN when given the floor.

4.2 Lecturer's client

The central task of the lecturer's client is to indicate that a student would like to ask a question, and to provide an intuitive user interface to accept or decline questions. At a certain point in time, a question may disturb the progress of a lecture or can be very undesirable in a didactical sense. Therefore, the client also offers the possibility to temporally postpone a question.

We assume that the lecturer uses PowerPoint slides to explain a new topic. The recorded video switches between the current slide, the head of the lecturer, the full view of the whole audience and the full view of the lecture hall. The audio of the lecturer is recorded together with all sounds of animations and simulations the lecturer plays. During teaching it is important that the client software does not disturb the lecturer. He should only focus on the client in case of a question. Therefore, we have implemented the client in such a way that it is invisible and running in the background.

If a student wants to ask a question the questioner's client submits this request to the lecturer's client via the QM server. A foreground window pops up at the lecturer's computer which stops the current presentation. A mouse click on a button of the lecturer's client is sufficient to give the word to the student, to ignore the request or to postpone it. In the last case, the window pops up again after a predefined time. The student gets different feedback on his/her PDA depending on the decision of the lecturer. Figure 3 shows the steps of the question-answer interaction and the according client screenshots of the questioner (left) and of the lecturer (right).

After the lecturer has given the floor, the student asks the question and the ALR system records his/her audio. The lecturer acknowledges the question and indicates the beginning of his/her answer by clicking the answering button of the lecturer's client, and the audio recording switches back to the lecturer. A last click by the lecturer signals the end of the question-answer interaction. The lecturer may allow additional questions, remarks or discussions, but he can also continue with the lecture.

4.3 Server

Let us finally look at the server application which represents the central component of the QM suite. It handles the communication with the questioner clients concerning all interaction events, of receiving the questioner's audio when given the floor, of communicating with the lecturer client for the relevant interaction events, sending consolidated events to the virtual director as sensor input, and providing an graphical user interface (GUI) to monitor and control the client events.

During start-up every questioner client automatically registers itself at the server by sending its IP address and its position coordinates. The connected client is represented by switching the background colour of the according field of the GUI to green. When a questioner announces a question the server receives the corresponding event message and performs three actions:

(1) Create an instance of the audio receiver for the questioner client.

(2) Raise an event for the lecturer client .

(3) Raise a sensor input event for the virtual director.

While a window pops up at the lecturer's client, the virtual director gives the orders to the audience's cameraman to aim at the position coordinates of the questioner and to zoom in. When the lecturer gives the floor to the questioner, the server receives the according event from the lecturer's client. Now the server sends out the sensor input event "questioner acknowledged" to the director, resulting in the audience camera being switched on air. Meanwhile, the audio stream over wireless LAN is sent from the PDA to the server, and the questioner is requested to ask his question.

When getting the signal from the lecturer's client that the lecturer starts to answer the question, the server stops receiving the questioner's audio and announces that the answer starts to the questioner's client as well as to the virtual director. At last, when the server gets signalled the end of the answer either from the questioner or from the lecturer, it resets all displays settings and audio components and reports the fact that the system should be switched back to the lecture context by the virtual director.

Furthermore, the server can block any question announcement if the "block clients" property is ticked; announcing students will now see "sorry, not possible" on their PDAs. By clicking on a number of the lecture hall abstraction shown in the GUI the audience camera is aimed at the respective seat. Finally, the "home position" button resets the camera adjustment. Figure 4 shows the GUI of the server, registered questioner clients are shown in dark shading while the speaking questioner is shown in medium shading.

5. Experience

As we are still at the beginning of using the QM system with our ALR, we have little practical experience up to now.

From the very first time of testing the questioner's client in the lecture, there was a spontaneous reaction of the students how to attract the lecturer's attention almost immediately by being able to interrupt his/her presentation. This led to the implementation of the feature to block clients, just to make sure that the lecture can be

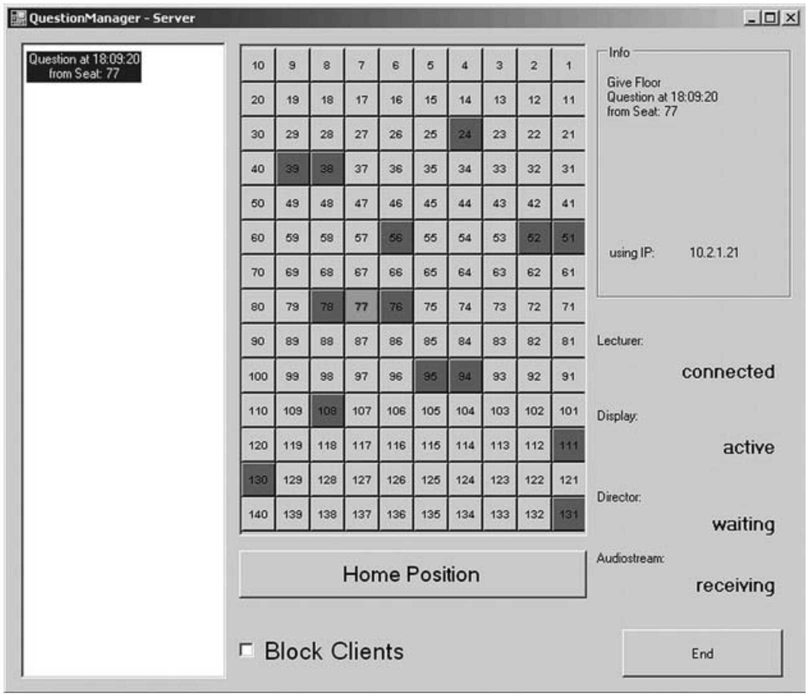

Figure 4.
GUI of the QM server, client on seat 77 asking

continued normally. The new way of asking questions was immediately accepted by the students without any additional training, even though it was not obvious for them how to handle repeated inquiries through the user interface. Another observation during our early tests concerns the acceptance of the questioner being recorded. If his/her audio was amplified for the lecture hall the questioner was irritated hearing his/her own voice slightly delayed due to the transmission but the rest of the audience appreciated it as they now could easily understand the question.

A new experience for the lecturer is the possibility of being interrupted during the lecture by the client window popping up. It takes time getting used to it, and then the additional load of the lecturer is very small. Concerning his/her interface the lecturer wants his/her own "block clients" feature in addition to the one on the QM server interface. Both of them should then be synchronized, and we still have to evaluate whether the blocking of clients may be removed automatically after a certain time. In addition, amending the interface with a possibility to manually mark animations or simulations by the lecturer will improve the system. So the ALR system can continuously record important details shown on the lecturer's computer. Finally, for both types of QM clients, we have observed that the user interface with the clickable buttons is not yet intuitive and should be improved to be less intrusive and to be more versatile.

The experience with the QM server is fine, especially as the camera adjustment can be manually overridden in case of a position estimation error.

6. Conclusion and outlook

In this paper we presented our new approach to transfer the question and answer interaction of a lecture into the virtual world of our ALR system. We have mapped each involved role from the real world to its digital equivalent and implemented all the necessary tasks. Our distributed system provides support for the complete question and answer dialog and aggregates all upcoming events in order to derive suitable sensor inputs for the director module of the ALR system. So, the question manager is an excellent complement to the ALR system as both are modelled on the roles of their human pendants. Additionally, it solves the well-known problem of recording the questioner's audio in an elegant way. By using the built-in microphone of the PDAs it is easy to sample the voice and finally transfer the audio stream by wireless LAN.

In our future work, we will address at first the improvement of both user interfaces to get them more versatile and more intuitive, e.g. by presenting the position of the questioner in the lecture hall. Additionally, we currently do research on how to detect a hand-raising questioner in the audience and determine his position by using video and audio processing techniques. In order to get a less intrusive system, we think of implementing additional and more flexible models of student-lecturer interactions, as e.g. discussions which normally do not occur in pairs. Finally, empirical evaluations will be made to investigate the influence of the system on the lecture, e.g. to which extent a lecturer or a questioner is distracted by the system, and on the recording, e.g. to which extent the spectators of the final video benefit of the recording of questions.

References

Bianchi, M.H. (1998), "AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations", *Proceedings of the Joint DARPA/NIST Workshop on Smart Spaces Technology, Gaithersburg, MD*, pp. 30-1.

Cutler, R., Rui, Y., Gupta, A. and Cadiz, J.J. (2002), "Distributed meetings: a meeting capture and broadcasting system", *Proceedings of the ACM Multimedia 2002, Juan-les-Pins*, pp. 503-12.

Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H. and Pankanti, S. (2005), "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking", *IEEE Signal Processing Magazine*, Vol. 22 No. 2, pp. 38-51.

He, L., Cohen, M.F. and Salesin, D.H. (1996), "The virtual cinematographer: a paradigm for automatic real-time camera control and directing", *Proceedings of the ACM SIGGRAPH: 23rd. International Conference on Computer Graphics and Interactive Training*, pp. 217-24.

Huang, Q., Cui, Y. and Samarasekera, S. (1998), "Content based active video data acquisition via automated cameramen", *Proceedings of the IEEE International Conference on Image Processing (ICIP 1998)*, pp. 808-12.

King, T., Haenselmann, T. and Effelsberg, W. (2007), "Deployment, calibration, and measurement factors for position errors in 802.11-based indoor positioning systems", *Proceedings of the Third International Symposium on Location and Context-awareness (LNCS 4718), Springer*, pp. 17-34.

Kopf, S. and Effelsberg, W. (2007), "New teaching and learning technologies for interactive lectures", *Advanced Technology for Learning Journal*, Vol. 4 No. 2, pp. 60-7.

Lampi, F., Kopf, S., Benz, M. and Effelsberg, W. (2008), "A virtual camera team for lecture recording", *IEEE Multimedia Journal*, Vol. 15 No. 3, pp. 58-61.

Lauer, T. and Ottmann, T. (2002), "Means and methods in automatic courseware production: experience and technical challenges", *Proceedings of the E-learn 2002, Montreal*, pp. 553-60.

Rui, Y., Gupta, A. and Cadiz, J.J. (2001a), "Viewing meetings captured by an omni-directional camera", *Proceedings of the ACM CHI 2001, Seattle, WA*, pp. 450-57.

Rui, Y., Gupta, A., Grudin, J. and He, L. (2004), "Automating lecture capture and broadcast: technology and videography", *ACM Multimedia Systems Journal*, Vol. 10 No. 1, pp. 3-15.

Rui, Y., He, L., Gupta, A. and Liu, Q. (2001b), "Building an intelligent camera management system", *Proceedings of the ACM Multimedia 2001*, *Ottawa*, pp. 2-11.

Scheele, N., Wessels, A., Effelsberg, W., Hofer, M. and Fries, S. (2005), "Experiences with interactive lectures – considerations from the perspective of educational psychology and computer science", *Proceedings of the International Conference on Computer Supported Collaborative Learning 2005*, *Taipeh*, pp. 547-56.

Zhang, C., Crawford, J., Rui, Y. and He, L. (2005), "An automated end-to-end lecture capturing and broadcasting system", *Proceedings of the ACM Multimedia 2005*, *Singapore*, pp. 808-09.

About the authors

Fleming Lampi is a PhD Student and Research Assistant in the Department of Computer Science IV, University of Mannheim, Germany. His research interests include video recording, processing and transcoding. He has an MS in computer science and multimedia from the University of Applied Science in Karlsruhe, Germany. Fleming Lampi is the corresponding author and can be contacted at: lampi@pi4.informatik.uni-mannheim.de

Hendrik Lemelson is a PhD student and research assistant in the Department of Computer Science IV at University of Mannheim, Germany. In 2006, he received his diploma in business information technology from the University of Mannheim. His research interests include 802.11-based tracking, location-based services and wireless sensor networks.

Stephan Kopf is a postdoctoral researcher in the Department of Computer Science IV, University of Mannheim, Germany. His research interests include multimedia content analysis and new learning technologies. Kopf has a PhD in computer science from the University of Mannheim, Germany.

Wolfgang Effelsberg is head of the Department of Computer Science IV at the University of Mannheim, Germany. His research interests include computer networks, multimedia systems and e-learning. Effelsberg has a PhD in computer science from the Technical University of Darmstadt, Germany. He is a member of IEEE and ACM and serves as an editorial board member of several multimedia journals.