This is a preliminary version of an article published in
Proc. of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA), pp. 1475 – 1480, Vancouver, BC, Canada, June 2007
by Fleming Lampi, Stephan Kopf, Wolfgang Effelsberg

Automatic FSM-Based Video Directors for Lecture Recording

Fleming Lampi
Department of Computer Science IV
University of Mannheim
68131 Mannheim, Germany
lampi@informatik.uni-mannheim.de

Stephan Kopf
Department of Computer Science IV
University of Mannheim
68131 Mannheim, Germany
kopf@informatik.uni-mannheim.de

Wolfgang Effelsberg
Department of Computer Science IV
University of Mannheim
68131 Mannheim, Germany
effelsberg@informatik.uni-mannheim.de

Abstract: The manual recording of lectures is a common technique today. To enable an *automatic* recording of lectures we propose an approach based on a sophisticated finite state machine implementing video production rules. The main goal is to improve the quality of the recording and a higher level of engagement of the spectators. In order to test our approach we define the evaluation criteria and compare our approach to a finite state machine using fixed weights. This paper introduces the development of the criteria coming from typical questions of the director of a human camera team. In a second step we test the criteria with two finite state machines. A brief overview of the project status and its first results concludes the paper.

1. Introduction

Lecture recordings can be seen from very different points of view and will of course not replace a lecturer holding its own lecture live. But they provide an additional value to support learning; the recordings are mainly used for preparing exams. Usually the students are highly motivated to concentrate on the material. Nevertheless, learning by watching recordings can easily become a really boring scenario, completely independent of how fascinating the original session was. Lecture recordings are very popular to enrich the learning experience and the content of learning management systems because they are easy to achieve (Lauer, T. et. al., 2002). In many cases they are limited to the recording of the slides and the spoken audio of the lecturer, and therefore they are often quite boring. To overcome this problem we recently presented our approach of an automatic camera control system for lecture recordings (Lampi, F. et. al., 2006), simulating a complete camera team by taking video production rules into account.

The director of a camera team decides which scene goes “on the air”. Based on the screenplay planned and on events occurring during the recording the director (re-)decides quickly which scene to present next. Our goal is to imitate the director using multiple cameras and a finite state machine (FSM) to implement the decision-making process. Up to now, similar approaches use fixed weights to select the next reasonable scene. In order to come closer to the human director we introduce contexts to apply finer grained conditions for the transition functions and the calculation of their probabilistic values. The contexts allow to differentiate between similar states to use different conditions depending on the context.

2. Related Work

Automatic camera recording is used in different fields, some of which are similar to our work. At first there is a large amount of research done on video surveillance (e.g., Hampapur et al. 2005). Other research was done by indexing recordings and recognizing the transitions from shot to shot (Mukhopadhyay et al. 1999). Both types are not directly related to our system. Somewhat closer to our area are meeting recordings which are often done by 360° camera and microphone sets, e.g., (Rui et al. 2001a, Cutler et al. 2002). This is an approach for smaller groups in meeting rooms; it has been transferred to larger rooms at Siemens Corporate Research (Huang et al. 1998) but through the position of the camera it seems more like a surveillance video rather than a lecture recording.

There are two general approaches for recording presentations or lectures with a larger audience. The first approach records the entire scene with a high resolution or wide angle camera and uses the image or a subset of it in standard resolution for framing active parts of the presentation (e.g., Rui, Y. et al. 2001b, Liu, Q. et al. 2002). This is not useful for our work because it varies only the zoom to the lecturer or the slide but not its angle or rather its point of view. Secondly, recording is done by cameras on pan and tilt heads which use image processing for framing and following the lecturer; This approach is closer to ours. A representative is AutoAuditorium (Bianchi 1998). It shows a basic level of automatic presentation recording without any video production rules implemented. More advanced is the system used by Microsoft Research (Rui et al. 2004) which has been improved meanwhile (Zhang, C. et al. 2005a, 2005b). It uses multiple cameras, implements basic video production rules, uses a video director module based on a finite state machine (FSM) and can be configured by a scripting language for cinematography rules. In spite of these correlations to our approach it differs in many ways; it uses image processing to frame and track the lecturer while we will use an indoor positioning system. In addition the authors also mention that there is a strong need for further research, e.g., on eye gaze orientation detection in order to implement more sophisticated video production rules. In particular, the implementation of the cinematography or video production rules differs. By using a scripting language, their rules are simply rewritten in a note form and therefore stand for fixed durations of the shots and predetermined transitions coming from the fixed weights for alternative transition targets given in the script. For the recording of real-time applications, similar basic rules are used in (He, L. et al. 1996).

3. Our Finite State Machine based on the Implemented Video Production Rules

Video production rules are necessary for every camera team, in particular an artificial one. We decided to implement them in a more detailed and sophisticated FSM to generate more engaging lecture recordings. Most of the existing video production rules can be described as a reaction to events or to a change in context. For example, if a lecturer and a questioner discuss a topic they should be shown on opposite sides of the screen to give the impression that they face each other. To detect such a discussion and to react to it properly many details have to be taken into account:

- At first it has to be clear which persons are allowed to talk to avoid a disturbance of the lecture.
- Persons talking to each other must be identified. This can be detected by the audio level of the microphones used.
- In the next step the position of the people and the cameras must be known. So each camera can visualize them on opposite sides of the screen.
- Now the director can switch between two cameras. This gives the viewer the impression that the two people face each other directly. The duration of each shot can be determined by the duration of their talk.
- After several direct switches between these two shots, it would be more interesting to show a “neutral shot”, i.e., the audience and their reactions, a “very long shot” of the whole scenario, etc.; this selection can be done by using a certain kind of history function over the last few shots.

Additionally there could be a change in the context, leading to different conditions for the transitions. Imagine that the discussion between the student and the lecturer refers to a detail on a slide. In this case, it would be good to show the slide and the person who is speaking by using a picture-in-picture mode.

Our approach is based on an XML description of the FSM, its states and its possible transitions. A transition may be accompanied by conditions. A major advantage of our approach is that the cinematography rules are incorporated in the FSM and not rewritten in another language. In order to stay flexible the underlying FSM is loaded at runtime and is therefore not hard coded. An XML file is used which is readable and editable as well by computers as by humans.

Because there are no hard coded transitions in the FSM it is impossible to write detailed video production rules as commands or conditions in a script language. Our approach makes it possible to describe under which conditions a certain possible transition should be preferred. For example, a transition from a very long shot of the lecture room to a medium shot of the lecturer should be preferred if the lecturer is gesticulating. So we amend the possible transition with the condition object ("lecturer") and the condition ("gesticulating"). For all conditions fitting a situation at runtime the associated transition gets a slightly higher probabilistic value, all non-fitting transitions get their probabilistic value slightly decreased. Our goal is to implement cinematography or video production rules in a finite state machine and yet keep them flexible, depending on the situation.

Reacting to the Environment

Because many of the cinematography rules are reactions to their environment, it is necessary to build our FSM in a suitable way. Good examples for an environment a camera team usually reacts to are a gesticulating lecturer, a noisy audience, a questioner posing a question or a slide getting annotated by the lecturer. In case such an action happens the director tries to show an appropriate shot. If such an action is shown immediately it is easier to understand the entire scenario. But there is a restriction: In case the director has just switched to a new shot he or she will wait a reasonable time to let the spectators perceive this shot before he or she reacts on the environment. Usually nothing of the planned or expected actions is so important or urgent to skip an active shot; only exceptional circumstances could be a reason to do so.

According to video production rules a standard duration for a shot is about 6 seconds; 4 seconds are the absolute minimum duration a spectator needs to perceive a shot. Depending on the screenplay, scene, speech or lecture shown the maximum duration may vary from 8 seconds to several minutes. It is possible to switch to another shot while still using the same audio stream, e.g., showing the audience while the lecturer talks. It is hard to give appropriate rules for such a situation; if too many different shots are shown the spectator will be confused; if too few shots are shown the spectator gets bored. Additionally, the best cuts are motivated by the action. For example, if the lecturer marks something important on a slide then it will be best to show the slide in full size in the next shot. If there are too many unmotivated cuts in a recording, the spectator will get confused again.

Using an FSM as a video director leads to another problem; in case the next state shown is chosen randomly and with fixed weights a spectator may predict the next states the machine will choose. This comes from the equal distribution a random number generator is based on. If the spectator identifies a sequence of states and is able to predict one he or she does again not concentrate on the topic. We want our approach to take these considerations into account.

Principle

In our system each state of the FSM does not have a fixed duration but a minimum duration to let the spectator perceive at least a minimum of the shot, a recommended duration which should be long enough to perceive the shot completely, and of course a maximum duration amended by a range. For example a maximum duration of ten seconds may be amended within plus or minus 10%. Within this range the duration will be determined *randomly* every time a new state begins.

Starting from the active state all possible transitions are identified. There are two types of transitions: time-based and event-influenced ones. All transitions will be marked with a probabilistic value of 100% at the beginning. Using a history of the most recently used states all transitions leading to states which have been just recently used get a decrease of their probabilistic value. The more recently a state has been used, the greater the decrease will be.

For the next step we use sensor inputs such as "lecturer moving" or "slide annotated" which are detected by the cameramen-module or "question posed" signalled by the audience-module using the indoor positioning system, just to name a few. A reviewing process correlating the conditions of the transitions and sensor inputs follows in the next step. For each sensor input a factor is defined which can increases or decreases the probabilistic value of a transition if its conditions meet the sensor's input. If no sensor input meets the condition a slight decrease of the probabilistic value is applied. By this procedure it will be more likely that the FSM reacts on the environment. All event-influenced transitions will also be checked against those sensor inputs. If there is a correlation, e.g., when posing a

question, a trigger sets a high probabilistic value for the transition to emphasize its acuteness. After all, the transition with the highest probabilistic value will most probably be chosen.

4. Testing Our Approach

After implementing our algorithm we were interested how it reacts to sensor inputs in comparison to a typical simpler FSM in which the next state is selected at random. In order to compare the two approaches we have defined a measure first. “How long does it take to show a certain shot after an according event has happened?” For example: “How long does it take to show the slide after the lecturer started to annotate it?” It is directly derived from the work of a director; he or she wants to show a new event as fast as possible to the spectators but while considering cinematography rules. Of course it is only one criterion to determine the behaviour of a finite state machine, and at first sight it apparently answers only the question: “How long does it take to show the requested shot?” But furthermore it is a measure of quality: If it takes longer to show the correct shot it is more likely that one will miss an aspect or an action. If someone missed a question it does not make much sense to listen to the answer. In general a reaction to an event should be prompt; otherwise the viewer may loose interest. To generalize this criterion we also ask: “How long does it take to show all requested shots on the average?” But reacting quickly to an event also assures two more aspects: First, the duration of a shot can be easier limited so it will not get boring, and second a transition based on an event is always a motivated transition. So the spectators will be less confused and more engaged. We ask for the percentage of unmotivated transitions compared to all transitions. Evaluating this measure will be a criterion concerning the quality of a director module of the automated lecture recording system.

Setting Up the Test-Scenario

Because we want to compare one single criterion of finite state machines acting as a director, we built the FSM of our approach and a second FSM which used the same random function to determine the duration of a shot as our system, but used a random function to select the next state only weighted by fixed values for each state; it did not react to any sensor input. Let us call the first one “sophisticated FSM” and the second one “simpler FSM”. In the next step, we created an application to record sensor inputs during a lecture. It saved the timestamp for each sensor input. These sensor inputs can be sent to both finite state machines over and over again. The “sophisticated FSM” reacts on these inputs; the “simpler FSM” neglects them. So the “simpler FSM” will always use the same fixed weights to determine the next state. The behaviour of the “sophisticated FSM” is different. Remember that the duration of a shot is variable, so at each run of sending the sensor inputs to the FSM it can be in different states at a certain timestamp. Therefore the possible transitions may be different and the effect of the sensor input of a certain timestamp may also vary. This leads to a similar but not identical behaviour of the “sophisticated FSM” for each “replay” of the lecture. We have recorded the sensor inputs of several lectures like “slide annotating”, “slide change”, “lecturer gesticulating”, “Question posed”, etc. We began with a small number of lectures, namely three, but each included hundreds of shots which we actually want to analyze. After every future improvement of our FSM we will rerun the tests to get more and more precise information. Additionally we are going to record more and different lectures.

Simulation Results

Through our simulations we gained values of 4855 shots shown by the “simpler FSM” and values of 5222 shots shown by the “sophisticated FSM” during the simulated lectures. Our first criterion is the average duration to fulfil a requested shot, which was done by the “simpler FSM” in 4.85 seconds in contrast to the “sophisticated FSM” which did it in 3.56 seconds. Remembering that the faster an action or aspect is shown the easier it is to follow the recorded lecture it comes out that the “sophisticated FSM” is clearly better concerning this criterion.

From the average we have a closer look on the minimum and on the maximum duration to fulfil a requested shot. The absolute minimum duration is zero seconds in case that occasionally the requested shot is already shown. This is true for 67.72 percent of shots shown by the "simpler FSM", but it is true for 71.49 percent of shots shown by the "sophisticated FSM". The maximum duration to surely fulfil a requested shot is the same as the time after which 100 percent of the requested shots has been fulfilled. The "sophisticated FSM" reaches this goal after 193 seconds which

is nearly 90 seconds faster than the "simpler FSM" reaching it after 286 seconds. So again for these two measures the "sophisticated FSM" performs faster than the "simpler FSM".

Looking at the percentage of unmotivated cuts, which are cuts, not requested by a sensor input, but forced by the maximum length of a shot, their values are in the same range. To be more precise it is 15.03% for the “simpler FSM” and 12.52% for the “sophisticated FSM”, which is a slight advantage for the latter. An overview over these values is given in table 1.

	Simpler FSM	**Sophisticated FSM**
Number of shots	4855	5222
Percentage fulfilled after 0 seconds	67.72%	71.49%
Average duration to fulfil 100% of the requested shots	286 sec	193 sec
Average duration to fulfil a requested shot	4.85 sec	3.56 sec
Percentage of unmotivated cuts	15.03%	12.52%

Table 1: Simulation results of both finite state machines

To give you an impression of the characteristics of both finite state machines concerning the percentage of fulfilled requested shots over the time, figure 1 shows the values of the “simpler FSM” as triangles and the values of the “sophisticated FSM” as diamonds, which are printed darker. As assumed from the faster average and from the shorter duration to reach 100% the “sophisticated FSM” got a steeper curve and therefore fulfils more requested shots in less time.

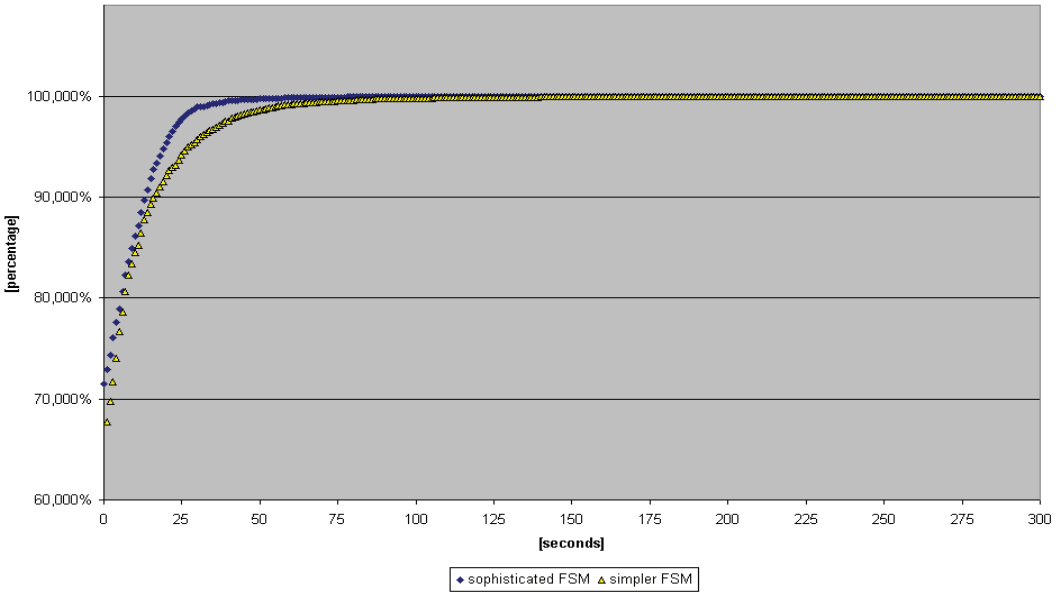

Figure 1: Percentage of fulfilled requested shots after n seconds

The principle of the “simpler FSM” has already proven its ability to act more or less satisfyingly as a video producing director in some implementations even though it tends to be predictive and uniform; but our approach, the “sophisticated FSM”, has shown that it is able to act much faster as the “simpler FSM” and is able to diversify the following states more easily based on the sensor inputs.

5. Project Status

The project status is still “work in progress”. After building the basic director module, we are now stepping forward to endorse it feature by feature. Therefore the work has been pushed in various directions to improve the underlying FSM description, to implement step by step the processing of the environmental influences, such as sensor inputs. At the moment we are still in the development and testing phase, so there is no practical, productive experience with the automatic recording system yet.

6. Concluding Remarks and Future Work

In this paper we compared two approaches of finite state machines acting as video producing directors. Based on the fact that an event occurs during the recording, it is best to show it immediately or at least as soon as possible. Therefore we defined the time to fulfil a requested shot as a measure and analyzed it in different ways: “Mean number of fulfilled shots after a certain time” or “How long does it take to fulfil all requested shots on the average?” Additionally, if there are too many so called “unmotivated” cuts, which are not requested by a sensor input, but forced by the maximum length of a shot, in a recording it affects the spectators in a confusing way. Therefore, we evaluated the percentage of unmotivated cuts compared to the number of all cuts. In all three categories our proposed “sophisticated FSM” achieves better values.

In our future work we are going to improve the system by adding a cameraman module and an audio engineer module into the system even a lighting control module is planned. Although the complexity of our system will increase we are expecting that the values will improve. Additionally we are working on the enhancement of our FSM description and its probabilistic transition functions. Our long term goal is to implement many cinematography or video production rules to improve the quality of lecture recordings and to do thorough testing with real students.

Literature References

Bianchi, M. (1998). AutoAuditorium: A fully automatic, multicamera system to televise auditorium presentations, *Proceedings of the Joint DARPA/NIST workshop on smart spaces technology*, Gaithersburg, MD, USA

Cutler, R. et al. (2002). Distributed Meetings: A Meeting Capture and Broadcasting System, *Proceedings of ACM Multimedia 2002*, Juan-les-Pins, France, 503-512

Hampapur, A. et al. (2005). Smart video surveillance: Exploring the concept of multiscale spatiotemporal tracking. *IEEE Signal Processing Magazine 03/2005*, Vol. 22, No. 2, 38-51

He, L. et al. (1996). The virtual cinematographer: A paradigm for automatic real-time camera control and directing, *Proceedings of ACM SIGGRAPH: 23. International Conference on Computer Graphics and Interactive Training 1996*, 217-224

Huang, Q. et al. (1998). Content based active video data acquisition via automated cameramen, *Proceedings of IEEE International Conference on Image Processing ICIP 1998*, 808-812

Mukhopadhyay, S. et al. (1999). Passive capture and Structuring of Lectures, *Proceedings of ACM Multimedia 1999*, Vol.: 1, Orlando, FL, USA, 477-487

Lampi, F. et. al. (2006). Automatic Camera Control for Lecture Recordings, *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2006)*, Orlando, FL, USA, 2006, 854-860

Lauer, T. et. al. (2002). Means and Methods in Automatic Courseware Production: Experience and Technical Challenges, *Proceedings of E-Learn 2002*, Montreal, Canada, 2002, 553-560

Liu, Q. et al. (2002). FLYSPEC: A multi-user video camera system with hybrid human and automatic control, *Proceedings of ACM Multimedia 2002*, Juan-les-Pins, France, 484-492

Rui, Y. et al. (2001a). Viewing meetings captured by an omni-directional camera, *Proceedings of ACM CHI 2001*, Seattle, WA, USA, 450-457

Rui, Y. et al. (2001b). Building an intelligent camera management system. *Proceedings of ACM Multimedia*, Ottawa, Canada, 2-11

Rui, Y. et al. (2004). Automating lecture capture and broadcast: Technology and videography. *ACM Multimedia Systems Journal* Vol.10, No.1, 3-15.

Zhang, C. et al. (2005a). An Automated End-to-End Lecture Capturing and Broadcasting System, *Technical Report MSR-TR-2005-128*, Microsoft Research, September 2005

Zhang, C. et al. (2005b). An Automated End-to-End Lecture Capturing and Broadcasting System, *Proceedings of ACM Multimedia 2005*, Singapore, 808-809