Automatic Video Editing

AMIDA technology package description

Pavel Žák, Kubíček Radek

Graph@FIT
Brno University of Technology
Faculty of Information Technology
Božetěchova 2
612 66 Brno, Czech Republic

Technology developers:
Stanislav Sumec, Pavel Zemčík, Radek Kubíček,
Pavel Žák, Michal Hradiš

Contact persons:
Zemčík Pavel, zemcik@fit.vutbr.cz, responsible for AMIDA project
Herout Adam, herout@fit.vutbr.cz, responsible for Graph@FIT group

Last modified: January 3, 2010

Contents

1 Purpose of the technology

This technology provides automatic video editing, that allows automatic producing of single video stream from more cameras by choosing the one that contains most of relevant information about what is happening in the observed scene and also preserves several aesthetic aspects in the final video that makes it more attractive to the viewer.

It can be used for example in meeting environments when the meeting room is observed by several cameras. Here the technology enables meeting summarization, distatnt meeting participating or just for storing only important informations in distilled form (instead of all recorded streams).

2 Features

This technology has been developed primarily for meeting scenarios and is capable of processing either online or offline data. Its core forms versatile rule-based editing algorithm that can be configured for specific scenarios.

Online editing can be done in any meeting room without specific camera setup and its design enables customizing for specific scenarios. No special hardware is needed, the basic setup for realtime processing contains personal computer with several firewire cameras(see Technical specifications).

Figure 1: Online video editing system setup with three firewire cameras.

Offline editing can be done from meeting transcriptions or annotations, which enable more accurate output.

The final package we propose contains implementation of this technology together with its additional more detailed description. The implementation is ready in configurable application for either online or offline processing. Demonstration video of online editing capabilities can be seen in [1].

3 Technical description

This section summarizes how the technology works. More detailed technical information can be found in references[2][3].

3.1 Video Editing process

The function of the proposed algorithm can be formulated as a problem of one camera or of several combined cameras selection in each time point of the recorded meeting. The image from the selected camera has to preferably represent what is happening in the meeting room according to different (user specified) aspects. Satisfaction of these aspects warrants that produced video contains as much of the relevant information as possible.

The input of the editing process is a set of facts - results of audio/video stream processing(and/or video annotations), set of rules - the predefined general video editing rules and a dynamically changeable set of rules based on the details of the audio/video processing, user requirements, etc. The output of the video editing decision process is a text description of the editing decisions that is afterwards interpreted in a video editing engine that actually processes the audio/video data. The whole algorithm works in the following way captured also in figure 2:

1. Source video streams of all cameras are simultaneously processed from the beginning to the end of the meeting.

2. Specific set of events (or features) is detected in each stream(or just loaded from annotations).

3. Based on detected events (which specify each stream information importance) and aesthetic editing rules the output stream is chosen for specific time.

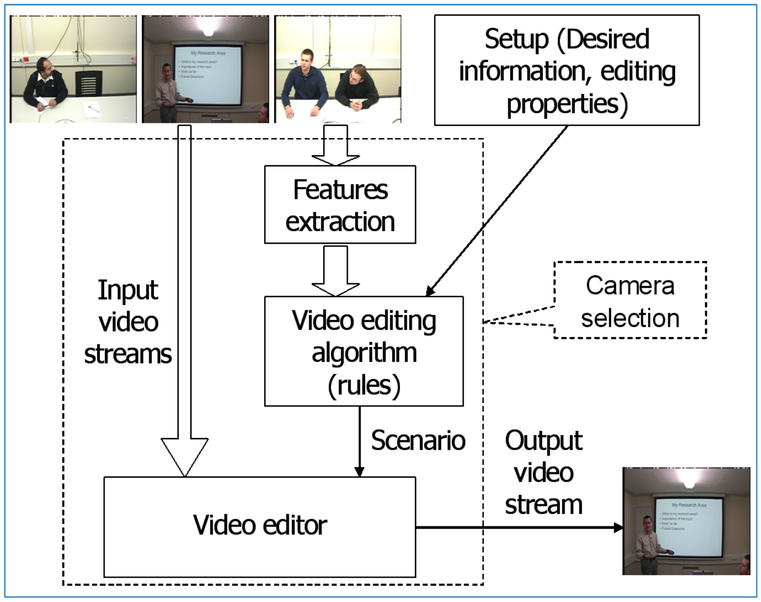


Figure 2: High level scheme of whole editing process

3.2 Editing algorithm

An image from the selected camera has to preferably represent what is happening in the meeting room according to different (user specified) aspects. Mainly, technical and aesthetical aspects warrant that produced video contains as much of the relevant information as possible.

The main idea is that a methodology of the video editing can be described through a set of various rules. An application of these rules frame by frame to the whole meeting produces a scenario that can be used for generation of the final video. All rules can be divided into two basic classes. The rules of the first class can be used in realtime processing while the rules of the second class process data from the whole meeting and produce more accurate output in offline editing. The goal of the rules application is assignment of weight to every camera in the meeting room. After the weight of all the cameras is known, the camera with the highest weight is selected.

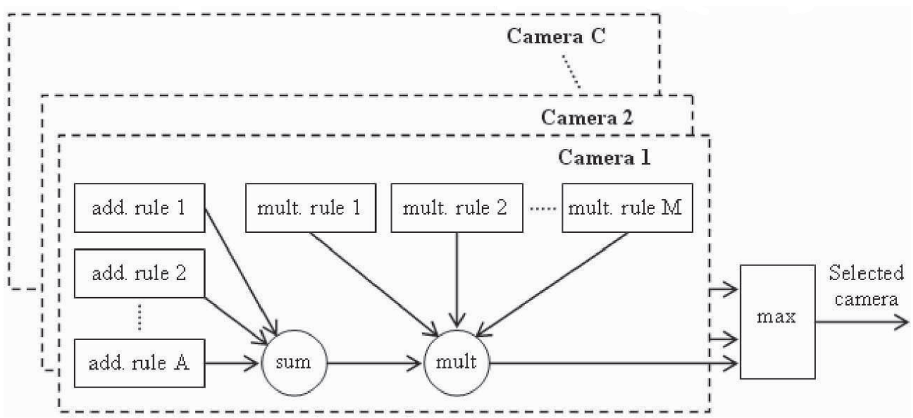
The main editing algorithm works as follows (see figure 3):

1. Input events detected in each camera streams are simultaneously processed frame by frame through the meeting.

2. Rules are used for evaluation of every camera in given time point.

3. The image from the camera with the highest weight is selected and presented as output in the given time.

Figure 3: Video editing algorithm, evaluation of different rules



Technical aspects of video editing are mainly represented by rules that evaluate e.g. activity of meeting participants or other interesting things such as slides projecting. The resulting weight is computed according to the activity and visibility of given person on certain camera. Other set of rules is important for satisfaction of the aesthetical aspects of video editing.

The first significant aspect of the participants activity is the information about whether the participant is speaking or not. The source data for these rules is obtained from meeting transcription or from automatic speaker identification. More important person is that person who starts speaking the first, also more important is that person who is speaking longer. It is also more important to show particular participant if he/she is gesturing by his head or hands.

Further, the rules can be also used to simulate some aesthetical aspects. Rules for handling periodic alternating of cameras add a little weight to cameras, which were not selected during long time period. This causes changes of cameras if no other activity is detected. Other rules are designed to assure that certain camera is selected at least for minimum given time period and the selection does not exceed maximal time period. These rules avoid quick camera changes that are not acceptable for the viewer and guarantee an interest of the produced video.

See [3] for details of proposed algorithm. There is also explained an approach for designing all kind of rules mentioned above and their detailed description.

3.3 Input Data

The input of the audio-video editing system is constituted by the multimedia audio-visual data themselves with metadata describing their properties and structure. As an input serve also events appearing in the data. These events – that are potentially somehow important for the video editing – are either manually annotated by a tool or are extracted automatically. The input data can be divided into two main parts – offline data and online data.

3.3.1 Offline

Offline data are used during the video editing process, when we already have collected audio and video records. In these records are some important events for the video editing process and it is necessary to have them annotated. In case of manual annotation, all the data is collected before the editing process starts, the automatic event extraction can happen during the video processing, whose output is the edited video sequence. All the manually annotated and automatically extracted events need to be stored for later use and for the editing process.

After all the necessary events are annotated, the offline input data are ready for use by the video editing process. It must be made an XML file describing these events and the used audio and video streams. This file serves as an input of the automatized video editing process.

The main disadvantage of this concept is in the fact, that is needed to have all the input data and detected events stored along, in most cases locally, and with a large number of data it is really space consuming and ineffective. So that concept of using annotated XML files and input data streams is usable especially in the case of processing a small amount of events and multimedia data, one time video editing or if the reusing of outputs is needless.

But in the case that is collected a huge amount of data and extracted events for them or when reusing of extracted events is needful, it is better to use an input data distributed system. To allow this, a component serving as the data storage of the annotations has been defined, that allows a unified way

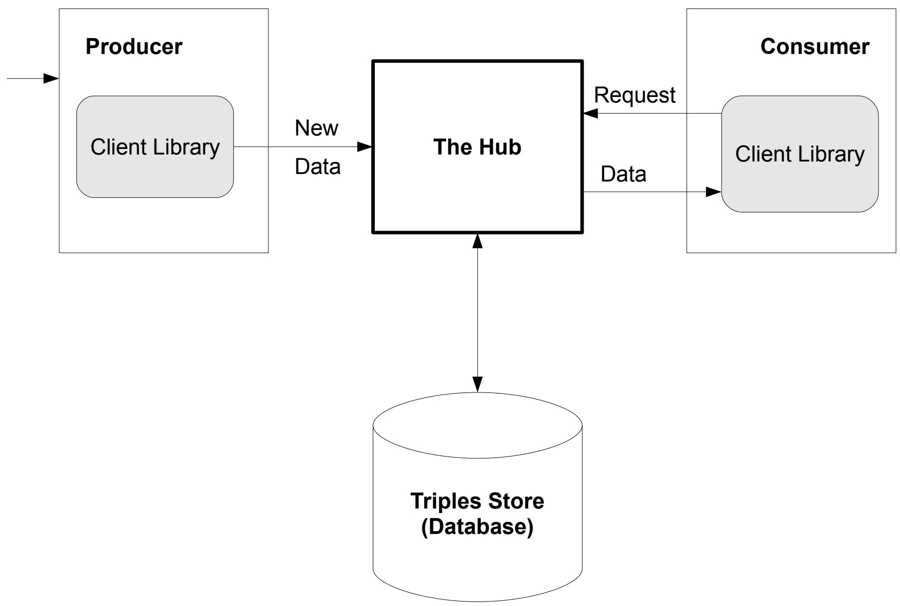


Figure 4: Producer-consumer scheme of Hub.

of storing and retrieving the data. This complete infrastructure is referred to as the Hub.

The Hub is intended to provide all of the storage that a group or a company needs for annotations about their archived meetings in one place. Each such Hub contains a database, which keeps the stored annotations and serves requests to the data. Along with the annotation/extracted data it contains all related documents, presentation slides and notes shown during the meeting. At the same moment when one event-extracting process inserts data to the Hub, the consuming application can retrieve it. This allows the Hub to be used both in real-time on a pending meeting to provide a newly connected client with all available information about the meeting as well as to post-processing the meeting data and production of summaries and similar material.

Records in the Hub have to be stored in hard defined format. In the video editing software it is necessary to have all needed data already stored in the

Hub before the editing process starts because of the editing process makes only one shot data reading from the Hub and then works with result data.

3.3.2 Online

Sometimes the audio and video records are not reachable at the moment and only data that can be processed by the video editing process is an online stream of a few cameras. For these cases it was developed an possibility to use an online (real-time) input data. The audio and video streams are acquired by number of cameras and microphones, in the next step they are sent to the editing process. During the editing the detectors are applied on the input multimedia streams to check and obtain the important events, which are sent into an editing process and according them the editing algorithm evaluates the best possible shot in every moment.

This online option is still in the form of some technological demo and for the real deployment is necessary to create the appropriate audio and video detectors searching for important events in the audio and video streams.

3.4 Output

After the editing process is finished, the results can be divided into two types, according to used mode of editing. At first, it could be the resulting set of events – marking the shot boundaries – which could be saved into the file or they could be saved to the Hub for future use. This type of output is necessary for the re-editing process, which results into edited video movie.

Edited video is the other type of an editing process output. If an editing process is launched in cutting mode, audio and video streams are available and a set of camera cutting events is already created, the editing process follows these camera events by choosing the proper camera and progressively creates resulting movie.

4 Limitations

Crucial limitation of presented video editing tool is lack of wider variety of real-time feature extractors, that could process the input video streams and detect more distinct events. However the design of the tool is prepared for implementing new extractors which makes it easily extendable.

5 Technical specifications

For the realtime processing computer with at least Quad-Core processor and 4GB of RAM is needed. The online setup is ready to use together with Unibrain firewire cameras.

- Processor: Intel Core 2 Quad or better
- Memory requirements: 4GB of RAM
- Disk space: 1.5GB (including demonstration data and configurations)
- Operating System: Windows
- Camera type: Unibrain Fire-i camera(s)
- Camera interface requirments: IEEE-1394a (FireWire) port

6 Package content

- Papers containing more detailed information
- Video editing software (both binary and source files) with additional tools
- Software using instructions
- Prepared demonstration configuration and data
- Demonstration videos

References

[1] GraphAtFit Youtube channel. Automatic video editing demo, 2009.

[2] Adam Herout, Radek Kubicek, Pavel Zak, and Pavel Zemcik. Automatic video editing for multimodal meetings. *Proceedings of International Conference on Computer Vision and Graphics*, 0, 2008.

[3] Stanislav Sumec. Multi camera automatic video editing. In *Proceedings of ICCVG 2004*, pages 935–945. Kluwer Verlag, 2004.