ICT-214793

TA2

Together Anywhere, Together Anytime

Large Scale Integrating Project
ICT – Networked Media

D6.6 Summary report – Audio capture and data analysis

Due date of deliverable: 30 November 2011
Actual submission date: 30 November 2011

Start date of project: 1 February 2008 Duration: 48 months
Lead contractor for this deliverable: IDIAP
Final version, 30 November 2011

Confidentiality status: Public (not yet approved by the European Commission)

Abstract

The challenge of TA2 is to develop new, representative, ICT based media experiences that support the social interaction between families and friends. This report provides an overview of underlying analysis approaches for extraction of the semantic information necessary for intelligent audio capture and higher-level media and stream manipulation to support TA2 goals.

Target audience

This deliverable has been constructed having the wider academic and industrial community in mind as target audience.

Impressum

Full project title: Together Anywhere, Together Anytime

Title of the workpackage: WP6: Data Analysis and Interpretation

Document title: D6.6 Summary report – Audio capture and data analysis

Editor: Danil Korchagin, Idiap Research Institute

Workpackage Leader: Hervé Bourlard, Idiap Research Institute

Project Co-ordinator: Peter Stollenmayer, Eurescom

Technical Project Leader: Doug Williams, BT

This project is co-funded by the European Union through the ICT programme under FP7.

Confidentiality status: Public

Executive Summary

The main purpose of this report is to document the intelligent audio capture and data analysis components developed within the TA2 project, reflect the challenges and evaluations. The described components are used by several higher level components of the TA2 system. These include the audio communication engine, the orchestration and composition engine. The semantic information is used for higher-level stream manipulation and automatic editing, for example to cut close-up shots from single persons who are currently speaking or to focus on a group of two persons having a dialogue.

The document is divided into three main chapters, reflecting the division between different concept demonstrators and architectural components within the TA2 system.

Chapter 2 (Intelligent Audio Capture) describes an acoustic audio capturing interface for immersive multi-channel group communication. The interface extracts directional information that allows multiple participants to be tracked in a consumer environment using a MIMO setup (multiple microphones and speakers) and provides noise reduction and acoustic echo compensation.

Chapter 3 (Data Analysis for Synchronous Scenarios) summarizes analysis approaches enabling multimodal stream manipulation for open, unconstrained environments. They comprise the detection and tracking of multiple faces, identification and recognition of multiple persons, estimation of head poses, depth information and visual focus of attention, detection and localisation of verbal and paralinguistic events, detection of some hand gestures (pointing and wiping), as well as the detection, recognition and tracking of specified patterns (on a sheet of paper).

Chapter 4 (Multimedia Structuring and Content Extraction) reports on several novel techniques devised to enhance the content annotation stage in the MyVideos concept demonstrator. It comprises the techniques on content synchronisation, detection of rapid camera movements, detection of heavy unsteadiness or people crossing the picture close to the camera, face detection and tracking, semantic information integration into MyVideos database for further exploitation.

List of Authors

Danil Korchagin, Idiap Research Institute
Stefan Duffner, Idiap Research Institute
Petr Motlicek, Idiap Research Institute
Carl Scheffler, Idiap Research Institute
Fabian Küch, Fraunhofer IIS
Rene Kaiser, JOANNEUM RESEARCH
Andreas Kriechbaum, JOANNEUM RESEARCH
Albert Hofmann, JOANNEUM RESEARCH
Michal Hradis, Brno University of Technology
Pavel Žák, Brno University of Technology
Rodrigo Laiola Guimarães, Centrum Wiskunde & Informatica
Michael Frantzis, Goldsmiths

Table of contents

Abbreviations

ACDE	Audio Cue Detection Engine
ACE	Audio Communication Engine
API	Application Programming Interface
ASR	Automatic Speech Recognition
CM	Communication Manager
DirAC	Directional Audio Coding
DOA	Direction Of Arrival
EC	Echo Control
ERB	Equivalent Rectangular Bandwidth
KCDE	Kinect Cue Detection Engine
KF	Key-Frame
KWS	KeyWord Spotter
MDM	Multiple Distant Microphones
OpenCV	Open Computer Vision Library
PCM	Pulse Code Modulation
SBD	Shot Boundary Detection
SVM	Support Vector Machine
UCDE	Unified Cue Detection Engine
VCDE	Video Cue Detection Engine
UGC	User-Generated Content

1 Introduction

The research project "Together Anywhere, Together Anytime" (TA2) seeks to understand how technology can help to nurture family-to-family relationships to overcome distance and time barriers. This document describes underlying audio-visual data analysis approaches for extraction of the semantic information necessary for intelligent audio capture and higher-level media and stream manipulation to support TA2 goals.

A generalized component architecture for the TA2 system is shown in Figure 1 (TA2 public report D4.5 "Component architecture"). It indicates the separation between server and client side components, and for simplicity only shows a single set of client-side components. The intelligent audio capture components, described in the chapter 2, are located within the block "Audio communication engine". The data analysis components, described in the chapter 3, are located within the block "Analysis". The components for multimedia structuring and content extraction, described in the chapter 4, belong to external server(s) dedicated for these tasks and therefore not shown in the diagram. For a detailed description of the scenarios, please refer to TA2 public report D3.5 "Summary report: application design and implementation".

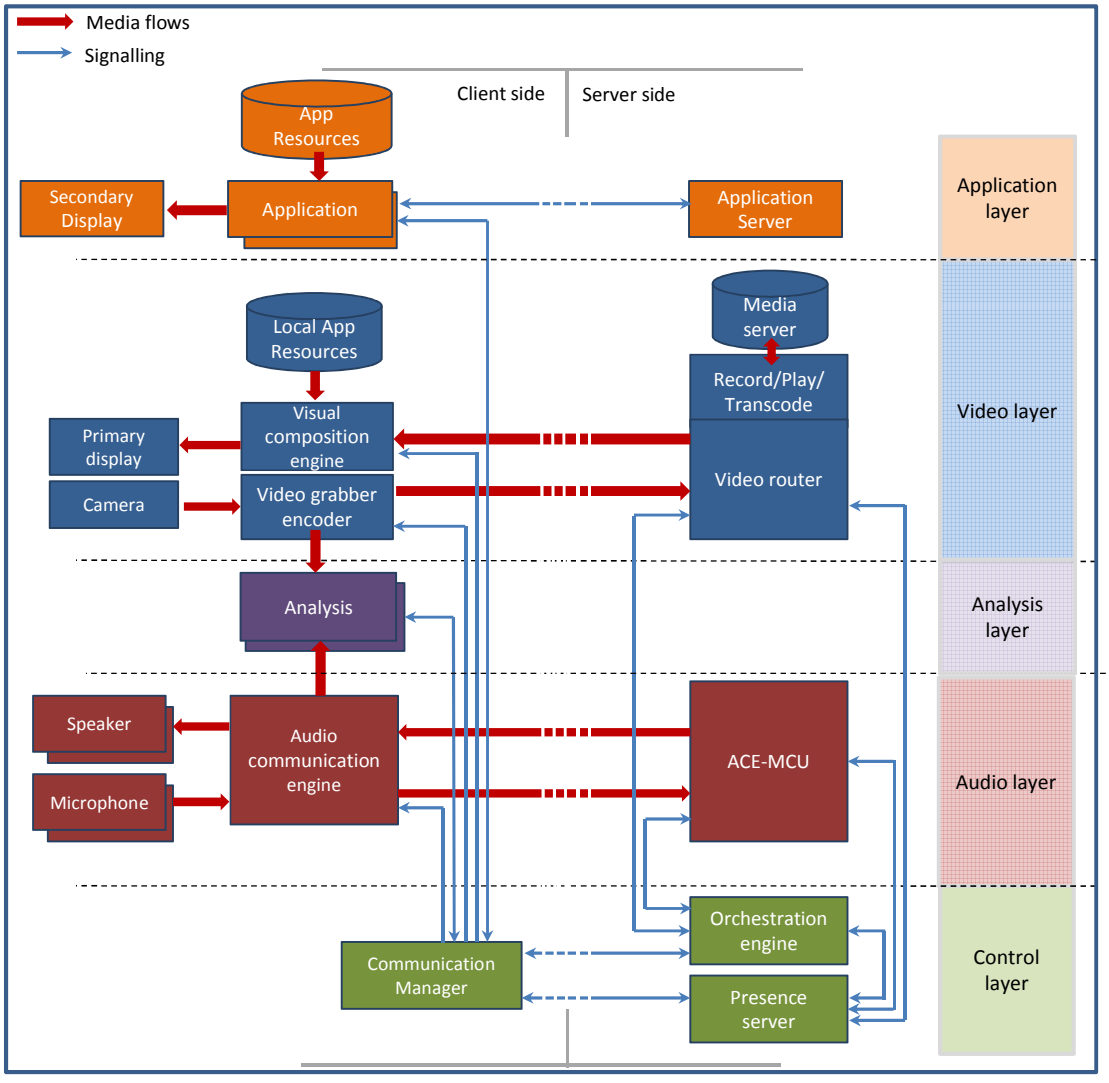

Figure 1: TA2 component reference architecture (D4.5 "Component architecture")

2 Intelligent Audio Capture

In this chapter, we describe the different functional modules included in the intelligent audio capturing which are intended to enhance natural real-time group to group audio communication. There are different signal processing tasks addressed within the acoustic interface. On the one hand they integrate the extraction of directional information for spatial audio recording, source localisation, and the manipulation of spatial audio sound scenes. Furthermore they provide speech enhancement for hands-free communication, such as acoustic echo control, noise reduction and beamforming by directional filtering.

As illustrated in Figure 2, the different components are integrated into the audio communication engine (ACE). The block denoted as DirAC (Directional Audio Coding) [1] includes the analysis of the sound field to extract directional information, whereas the block EC corresponds to acoustic echo control. Note that the beamforming (BF) is integrated into the DirAC block. Analogously, the noise reduction (NR) is performed within the EC module. The output of the EC module on the right side is used to provide with echo-cancelled audio signals to analysis engine (see chapter 3).

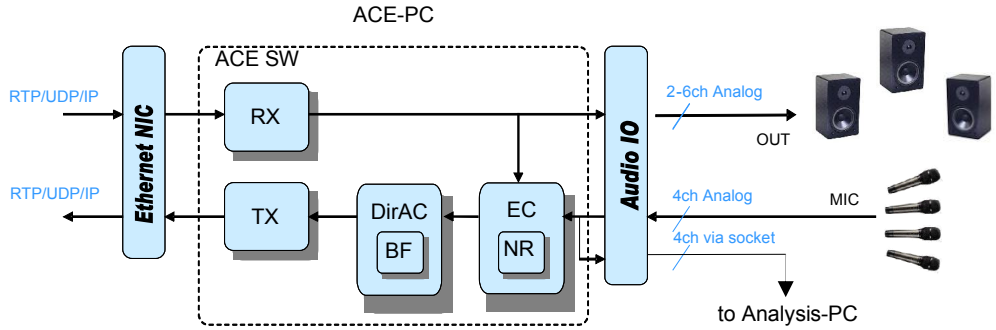

Figure 2: Basic functional structure of the intelligent audio capturing components within the audio communication engine (ACE)

The audio input signals are recorded by an array of multiple microphones. Different designs of the microphone configurations are discussed in section 2.1 in more detail. Based on these microphone input signals, the observed sound field is analysed by DirAC and specific parametric information is extracted based on a simple physical model.

DirAC represents an efficient approach for the analysis and reproduction of spatial sound, which has been adapted to the specific requirements of the different TA2 application scenarios. The method uses a parametric representation of sound fields based on the features which are relevant to the perception of spatial sound, namely, the direction of arrival (DOA) and diffuseness of the sound field in the frequency subbands. In fact, DirAC assumes that interaural time differences and interaural level differences are perceived correctly when the DOA of a sound field is correctly reproduced, while interaural coherence is perceived correctly, if the diffuseness is reproduced accurately. These parameters, namely DOA and diffuseness, are carried as side information which accompanies a mono audio signal. The DirAC parameters are obtained from a time-frequency representation (frequencies are grouped into so-called critical bands) of the signals captured by specific microphone arrays and are output for each frequency and time frame.

In the synthesis step, the information contained in the mono DirAC stream allows for an accurate spatial rendering with almost arbitrary loudspeaker configurations. As described in [1] in more details, direct sound is rendered as localisable sound events with vector based amplitude panning [2], whereas diffuse sound components are played back from all loudspeakers using decorrelated versions of the mono audio stream. At the preferred listing position, a listener will perceive the same spatial sound scene as it has been observed at the location of the microphone array.

2.1 Estimation of directional parameters

As indicated in the previous section, the most important parameters within intelligent audio capturing are the direction of arrival (DOA) of sound and the diffuseness of the observed sound field. In the context of TA2 scenarios, the playback configurations are assumed to have the loudspeakers within one plane, such as stereo or 5.1 surround sound setups. Accordingly, we consider the estimation of the azimuth angle to be sufficient to specify the DOA of sound. In order to cover the complete range of possible DOAs within the horizontal plane, planar arrays of microphones are used. In some application scenarios, it is desirable to place the microphones on top of a TV screen. In this case, linear microphone arrays are better suited due to form factor constraints.

The DOA of the sound can be estimated using a planar microphone array, where four omnidirectional capsules are arranged on the corners of a square. The picture on the right of Figure 3 shows such a microphone configuration integrated into an arc-lamp. A typical recording situation using this setup is depicted on the left of Figure 3. Due to the diamond-shaped arrangement of the microphones, it is also referred to as D-grid in the following.

Figure 3: Microphone array embedded in arc-lamp as shown at IFA 2009 (left) and detailed look from below showing the four omnidirectional microphone capsules (right).

As discussed in [3], the sound pressure and the particle velocity vector of the sound field at the center of the array can be determined from linear combinations of the four microphone spectra. These physical measures are used to compute the active sound intensity vector, representing the direction of energy flow in the observed sound field. The DOA of sound is then given as the opposite direction of the active sound intensity vector. Note that with a D-grid shown in Figure 3, the DOA can be determined within a range of 360° within the horizontal plane.

The diffuseness of the sound field can be determined by evaluating the ratio between the overall energy density of the sound field and the energy which is travelling into a certain direction, as indicated by the active sound intensity [1]. Additionally, the temporal and/or spectral fluctuation of the intensity vector can be considered for determining the diffuseness of the sound field. It allows for an improved temporal resolution for estimating the diffuseness. Consequently, the concept of measuring diffuseness becomes less dependent on the actual microphone array configuration.

The directional parameter estimation is exemplified in Figure 4 for a two talker scenario based on a recording with the microphone array shown in Figure 3. The left plot shows the spectrogram of the measured audio signal. In the centre, the estimated instantaneous DOA is shown for each time/frequency tile, where for this example, each band has a width of approximately 43 Hz. In the first third of the simulation time, only the talker located at an azimuth angle of 90° is active. In the second third, a talker located at -120° is active, and the last third represents a double-talk situation of the two sources. As can be seen, the DOA estimates reflect the true DOAs well during speech activity. During speech pauses, e.g. around time frame index 500 and 1000, the DOA estimates are rather

random, as expected for diffuse ambient noise. The increase in the estimated diffuseness for these time ranges is clearly visible in the plot at the right of Figure 4. It can also be noted that even during double-talk, the DOA of the two sources can be resolved due to high temporal and spectral resolution used in the DirAC analysis. It should be mentioned that the increased values of diffuseness during speech activity are resulted from reverberation in the room, which can be considered as diffuse sound components.

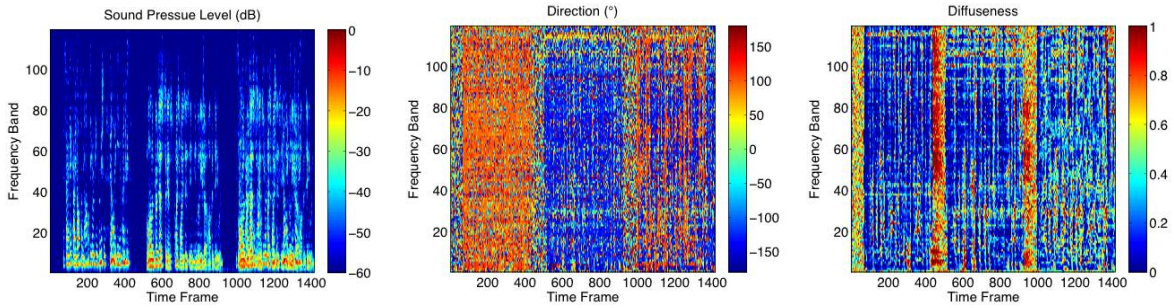

Figure 4: Estimated DirAC parameters of the spatial sound for a two-talker scenario: (left) sound pressure level, (center) direction, (right) diffuseness

Alternatively to planar microphone configurations a uniform linear array (ULA) of at least two omnidirectional microphones can be employed. An example of an ULA of four microphones arranged with equal spacing on a line is shown in Figure 5. The ULA geometry allows, for instance, an easy integration of the microphones into a TV screen. As a drawback of this geometry, one cannot distinguish between sound events arriving from front and back of the array, meaning that the DOA can be determined only within a range of 180°. The DOA estimation using linear arrays is explained below, where we follow the approach presented in [4].

Figure 5: Example for a linear array of four microphones

Estimating the DOA of sound with linear arrays is a well-studied problem. Two popular algorithms for this purpose are (root) MUSIC and ESPRIT [5], which are capable of computing multiple DOA estimates per frequency band. However, the drawback of these algorithms is a comparatively high computational complexity as they require eigenvalue decompositions. The interested reader is referred to [4] for investigations on the computational time in context of the current project. This is especially problematic for real-time applications and live-communication. When a single DOA estimate per frequency is sufficient, which is usually the case in communication systems, one can derive a computational more efficient DOA estimator similarly to ESPRIT. In principle, it works as follows: The DOA estimation is carried out in the short-time frequency domain. For each time-frequency bin the correlation matrix of all microphone signals is computed containing the corresponding auto and cross power spectral densities. The correlation matrix is separated into two sub-matrices such that all elements of the second sub-matrix contain the same phase shift compared to the elements of the first sub-matrix. This phase-shift, which contains the information on the DOA of the sound, is then determined via a least-squares approach and the DOA is computed.

Exemplary results for a DOA estimation using a linear array of three microphones with 3.2 cm spacing are depicted in Figure 6. The plots show the estimated DOA as function of the true DOA for different frequencies. Microphone noise has not been considered in this simulation.

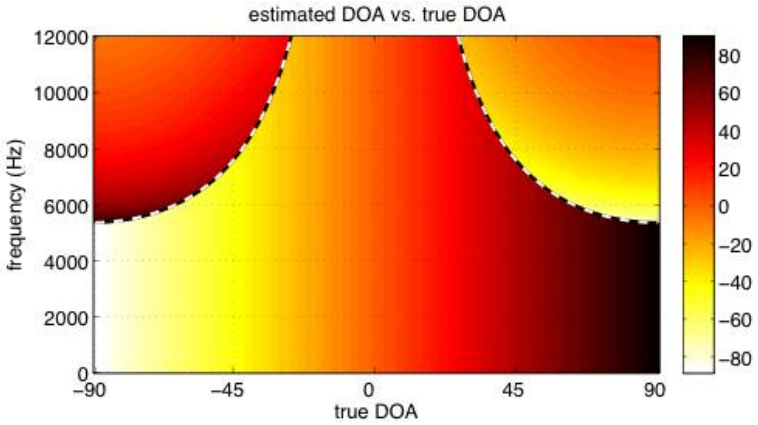

Figure 6: Ideal DOA estimation using a line array

As can be seen, the estimated DOAs are correct for all frequencies below the spatial aliasing frequency (indicated by the dashed line). For higher frequencies, errors due to spatial aliasing occur resulting from the non-zero microphone spacing. The spatial aliasing limit, however, is above 5 kHz.

To obtain a regular DirAC stream, the diffuseness of the sound field has also to be considered. As shown in [4], the diffuseness estimation can be efficiently integrated into the mathematical framework of the line array based DOA estimator. The diffuseness is estimated with line arrays based on spatial coherence between two microphones, where the coherence is low for diffuse sound and high for non-diffuse sound. Since the coherence computation requires the power spectral densities of the microphone signals, it can directly be computed from the correlation matrix used for the line array based DOA estimation.

Exemplary results for the line array based diffuseness estimation are illustrated in Figure 7. The plots show the estimated diffuseness for different angles of the direct sound as a function of the amount of direct sound energy.

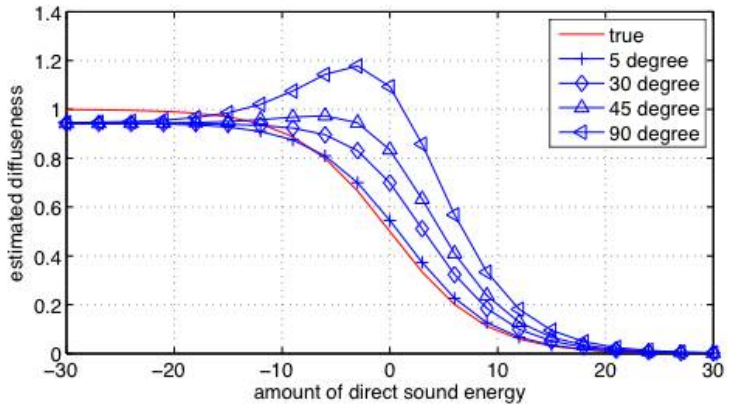

Figure 7: Diffuseness estimation using a linear array

Notice that 0° means that the sound arrives at the broadside of the array (frontal sound). The red curve represents the correct results. It can be observed that the line array provides accurate results for frontal sound events, which is the most relevant direction in practice when using a line array.

2.2 Estimating the number of sound sources

The sound source localisation algorithm [6] aims at estimating the directions of active sound sources with respect to the microphone array as well as their number. The individual steps of the algorithm are illustrated in Figure 8.

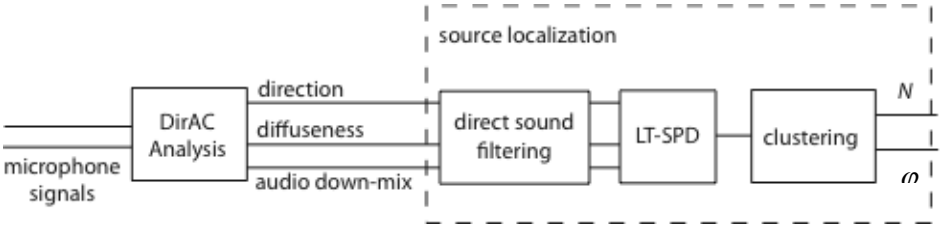

Figure 8: Single steps of the DirAC bases source localisation algorithm

Since the DOA estimates of the DirAC stream are corrupted by sensor noise and room reverberation, they cannot directly be exploited for determining the number and positions of the active sound sources. For this reason, a filter is applied on the DirAC stream that extracts the DOAs of the direct sound. The filter considers the diffuseness parameter and discards all DOA information whose corresponding diffuseness exceeds a specific threshold. In fact, the filter assumes that the reliable DOAs of the direct sound are characterised by a low diffuseness. This is in contrast to the reverberant sound, whose corresponding diffuseness is high and whose DOAs do not give any useful information.

After separating the direct sound, an energy weighted histogram of the direct sound DOAs is computed, the so-called long-term spatial power density (LT-SPD). To generate this histogram, the direct sound DOAs are collected over a specific time period. The longer this integration time, the more accurate is the source localisation, but the smaller is also the temporal resolution. The LT-SPD is computed by summing for each angle of the direct sound DOA the corresponding power in the mono audio down-mix signal. Figure 9 illustrates an example of an LT-SPD (black solid line) for the case that three sound sources are active at the same time. The plot shows three distinct clusters whereas the center of each cluster represents the direction of a sound source.

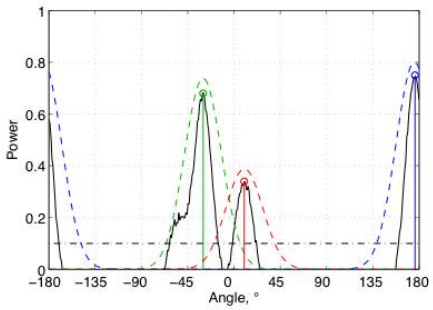

Figure 9: Exemplary LT-SPD for three active sound sources

In the final processing block, a clustering algorithm on the LT-SPD is applied which estimates the number of the clusters and the corresponding centres. This information corresponds to the number and the directions of the sound sources. As proposed in [6], one possible clustering algorithm is the so-called sparse maximum detection (SMD), whose main advantage is its low computational complexity. The SMD operates as follows: the global maximum of the LT-SPD is detected and considered to represent the first source position. We then overlay a Gaussian window with a certain maximum and width (blue dashed line) on this peak. All angles for which the LT-SPD falls below the window are considered to be associated with the detected source and are excluded from further processing. We then continue with detecting the next global maximum and repeat the steps until all remaining maxima fall below a specific power threshold (black dash-dot line). As result from the clustering block, we obtain the number of active sound sources and the corresponding directions.

2.3 Directional filtering

Among the acoustic sources included in the observed spatial audio scene, there can also be undesired interfering sources, e.g., air conditioning or noise entering though an open window. The intended directional filtering scheme, often also referred to as beamforming, includes the possibility of suppressing acoustic sources from certain directions while leaving other spatial regions unmodified. In addition to interfering directional noise sources, ambient noise is also present in the recorded microphone signals and should be removed. The reduction of stationary ambient or microphone noise is implemented within the acoustic echo control module and will be discussed in section 2.6.

The suppression of a single undesired interferer is considered in section 2.3.1. As shown in section 2.3.2, a similar concept can also be applied for extracting multiple sound sources in the context of object related spatial audio coding.

2.3.1 Directional filtering for a single direction

Traditionally, spatial filtering is realised by combining the signals of the microphone array such that for a desired look direction, the recorded sound superimposes coherently, whereas sound from other directions is captured with less sensitivity. Within the TA2 project, an alternative method to beamforming has been developed which is based on the DirAC parameters [7]. Due to the parametric approach, more flexible directional patterns are achievable compared to the physically constrained traditional beamforming techniques.

The general concept of directional filtering is illustrated in Figure 10. The first part corresponds to the DirAC analysis, i.e., the DOA and diffuseness are estimated for each time/frequency block. Based on the DirAC parameters, a spectral gain function $D_{\mathrm{DiFi}}(k)$, is determined. The spectral weights are then applied to the original mono signal of the DirAC stream $W(k)$ to perform the directional filtering. The gain function reflects the desired manipulation of the sound scene, i.e., signals components impinging from undesired directions are attenuated while others are kept unchanged. As a result of the design we obtain the spectral weights which are multiplied with the signal spectrum.

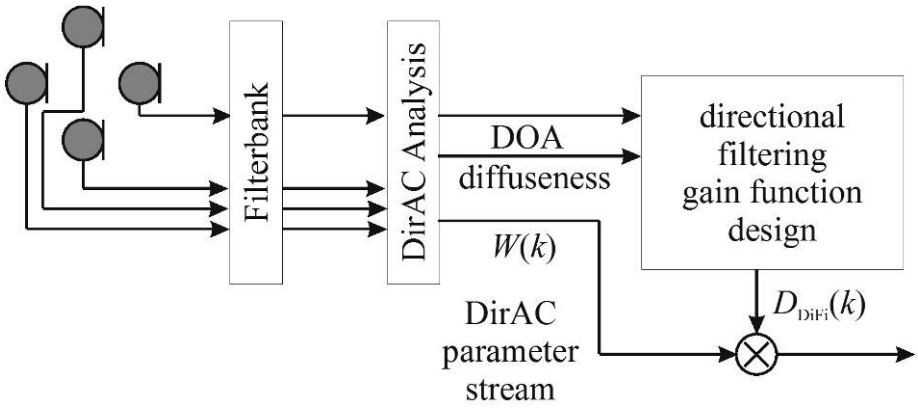

Figure 10: Schematic structure of the directional filtering method

The directional filtering weights are designed such that the distortion of the desired signal is minimised, while a sufficient level of interference attenuation is achieved. As shown in Figure 10, not only the DOA of sound is considered, but also the diffuseness is taken into account. Following the interpretation of section 2.2, high diffuseness values indicate that the current DOA does not reflect the precise location of a sound source. In this case, the directional filter introduces less attenuation to avoid distortions of the desired signal. On the other hand, low diffuseness values indicate strong direct sound components, which can be processed less conservative.

It should be mentioned that due to the high resolution in time and frequency, also more complex audio scenes having multiple takers may be addressed with the directional filtering concept.

It is important to note that this speech enhancement method does not affect the spatial distribution of sound sources after a DirAC synthesis step. Thus, it preserves the feature of spatial audio in the communication system. Since the directional filtering approach is directly based on the DirAC parameters, the beamforming functionality can be provided with high computational efficiency.

More algorithmic details of the directional filtering and the spectral gain function design are presented in [7].

2.3.2 Directional filtering of multiple acoustic sources for sound scene manipulation

If active sound sources within a room are localised correctly, a desired source can be selected and enhanced by means of directional filtering, as discussed in the previous section. However, if the goal is to manipulate an entire acoustic scene, we need to modify multiple sources' gain and/or position simultaneously. A practical approach is to operate multiple instances of a directional filtering processing unit, which will be discussed in the following.

First, we consider the general scheme illustrated in Figure 11. We assume that the position of the different sound sources has been detected by the localisation method as described in section 2.2. The corresponding source signals can be extracted by applying directional filtering with respect to the location of the sound sources. Then, a separate DirAC stream is assigned to each sound source by assigning a desired DOA and diffuseness to the audio signals. Usually, the diffuseness value is chosen to zero, reflecting the characteristic of point sources like talkers. Additionally, the volume of the audio signal may be adjusted. The desired spatial audio scene is then obtained by merging the different DirAC streams into one corresponding overall DirAC stream, which is then used for determining the loudspeaker signals in the DirAC synthesis. For more details about the merging of DirAC streams, the interested reader is referred to [8].

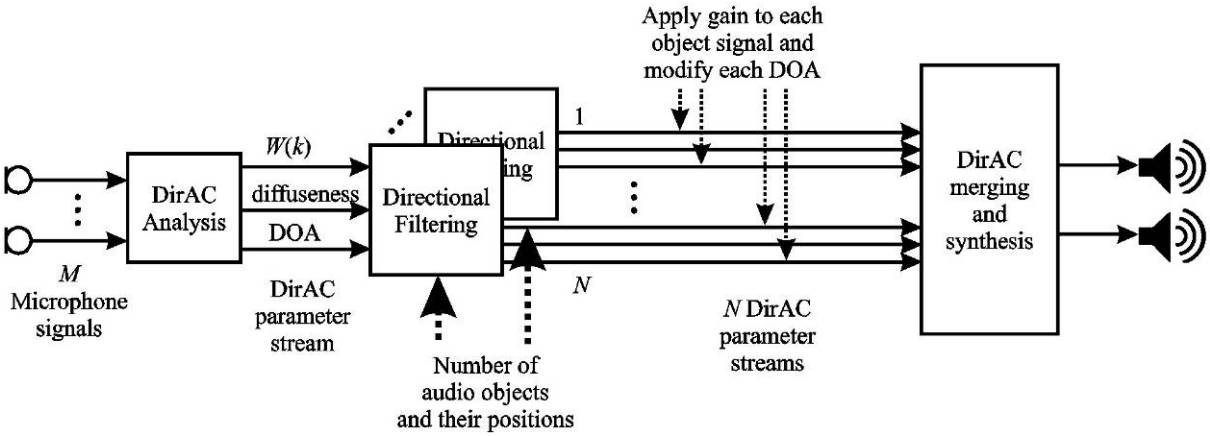

Figure 11: Schematic structure of capturing multiple audio objects, edit each object's DirAC parameter stream, merge and render the edited acoustic scene

The aforementioned concept offers a wide range of manipulation options while it can be efficiently integrated into the DirAC framework. However, it does not allow to manipulating the sound scene at the rendering location. Therefore, an alternative approach has been considered in the TA2 project: the separated source signals serve as input signals for the object-oriented spatial coding technique called spatial audio object coding (SAOC) [9], [10]. As described in [9] in more detail, the SAOC's rendering unit provides various options for interactive manipulation of single audio objects within a spatial audio scene. The general structure for combining directional filtering and SAOC is depicted in Figure 12.

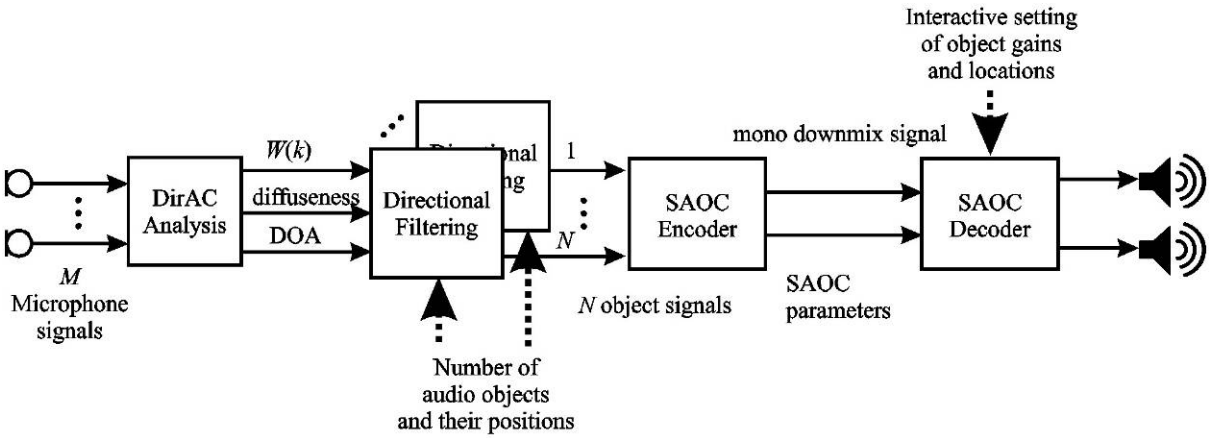

Figure 12: Schematic structure of capturing multiple audio objects and forwarding the separated source signals to an object-oriented spatial codec

It turns out that the parametric side information extracted by the SAOC encoder can be directly derived from the directional filtering weights used for separating the sound sources. An efficient method for transcoding the gain functions corresponding to the different objects to the SOAC parameters is proposed in [10].

It should be mentioned that all instances of the directional filtering processing unit capture their respective sound sources by means of narrow beams pointed to the corresponding position. This implies that a huge number of remaining directions cannot be controlled as there is no spatial object associated with them. For example, ambient sound field components such as reverberation are observed from these directions. In order to access this so-called *residual object*, we assign the remaining angle scope, which is not covered to capture active sound sources, to an auxiliary object. Figure 13 illustrates the case of two sound sources together with the corresponding residual object in a polar plot, where the angle refers to the DOA and the radius to the spatial gain of each respective directional filter.

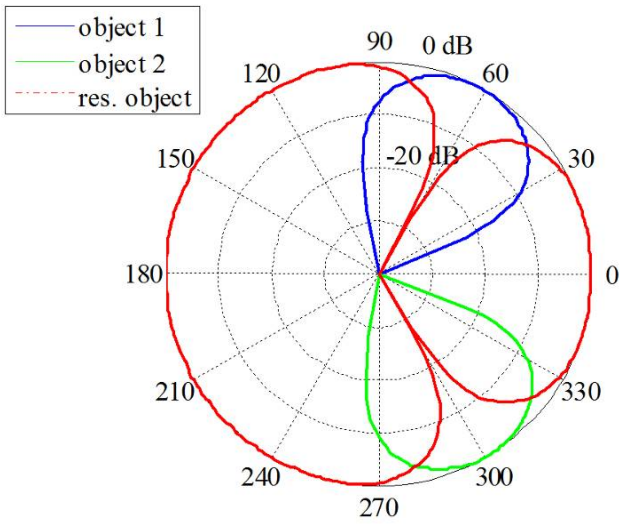

Figure 13: Polar plot of directional patterns to extract sound sources at +60° and -60° (the remainder is assigned to a residual object)

The spatial gain for the residual object can be derived from the requirement to preserve the overall sound energy level within the audio scene represented by the spatial audio objects [10].

2.4 Dereverberation

Directional filtering in the DirAC parameter domain aims at the attenuation of interferences which can be separated from desired sound sources by means of the DOA parameter. In contrast to these so-called directional sources we now consider disturbing sound, which are characterized by a high diffuseness parameter, i.e., the direction-of-arrival parameter becomes less reliable to perform a signal enhancement task. An important example of diffuse disturbing sound is reverberation: in rooms a desired speech signal is reflected from the surrounding walls multiple times. The delayed and attenuated reflections arrive from various directions at the microphone grid. The earliest reflections, which are reflected only once or very few times, may sometimes be recognized by DirAC as separate sound sources. However, after few more reflections the sound can be analysed as being diffuse – we obtain a high diffuseness parameter. This diffuse sound arrives rather late at the microphone grid compared to the direct sound and the early reflections. As a consequence, in a speech signal, phonemes start to overlap temporally and the intelligibility of speech is reduced.

In reverberant environments like typical rooms, the level of late reverberation in the microphone signal increases with increasing distance of a talker from the recording position. Thus, dereverberation becomes especially important for large distance hands-free talking as common in case of people are sitting on a living room couch while using a microphone array on the TV screen for communication purposes.

To overcome the problems of reverberation we present a method to design a filter for time-variant attenuation of spectral components which are associated with reverberant sound [11]. Clearly, the filter should attenuate the signal, when high diffuseness parameters occur, and leave it unmodified for low diffuseness parameters. The transition between these two extreme settings is configured differently for rising and decreasing signal levels. At rising levels we expect speech onsets, which should be preserved – the dereverberation filter changes quickly from attenuation to neutral properties. Constant or slowly decreasing levels indicate speech activity. In this case the attenuation level is kept rather constant to prevent artifacts. A positive side effect of this kind of smoothing behaviour is the preservation of early reflections. These are not attenuated as it is well-known that early reflections support speech intelligibility.

It should be noted that dereverberation on the basis of the DirAC diffuseness parameter is agnostic of the direction-of-arrival parameter and can be applied without source localisation. The schematic structure is similar to the procedure for directional filtering as depicted in Figure 10. However, in contrast to the directional filtering gain function design, the dereverberation gain function design does only use the diffuseness as an input parameter, whereas the DOA is ignored. At the output of the dereverberation filter design we obtain a spectral gain function D_{derev}, which is multiplied with the signal spectrum. Figure 14 shows the diffuseness parameter and the derived spectral gain as a function of time and frequency for a reverberant speech sample of 1.6 s duration. The sharp transition from high to low diffuseness values reflects a speech onset at block time index 50. Accordingly, the dereverberation filter increases its values to about 0 dB. In the subsequent period the gain is kept close to 0 dB before it is reduced to attenuate reverberation. At approximately block index 70 we have the next speech onset, where the filter gain increases again very quickly.

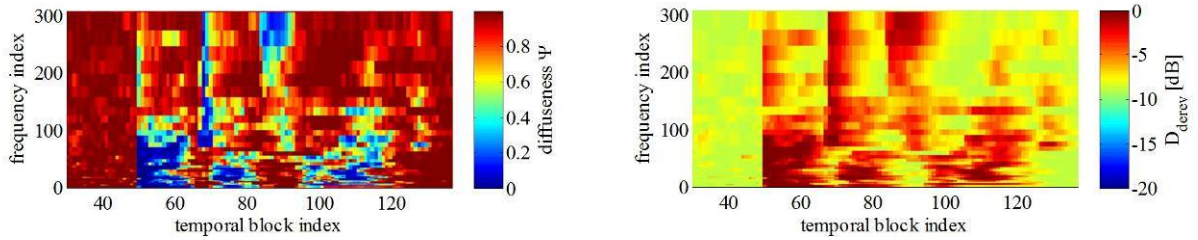

Figure 14: DirAC diffuseness parameter (left) and the dereverberation gain filter (right)

2.5 Acoustic echo control and noise reduction

Acoustic echoes arise from an acoustic coupling between the loudspeakers and the microphones of telecommunication devices. This phenomenon is especially present in hands-free operations. The general problem of acoustic echo control is illustrated in Figure 15.

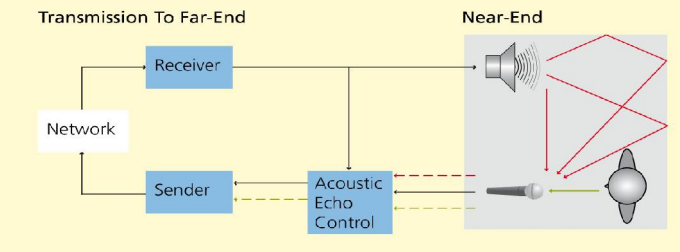

Figure 15: General acoustic echo problem

The acoustic feedback signal is transmitted back to the far-end subscriber, who notices a delayed version of his own speech. Echo signals represent a very distracting disturbance and can even inhibit interactive, full-duplex communication. Especially in case of high signal delays as is common with, e.g., VoIP applications or satellite transmission, echo control is crucial to allow for unimpaired conversation. Additionally, acoustic echoes can result in howling effects and instability of the acoustic feedback loop. In a full-duplex hands free telecommunication system, acoustic echo control (AEC) is therefore a necessity to cancel the coupling between loudspeakers and microphones.

In the AEC [12], [13] used in the TA2 project, the frequency spectrum of the microphone signal is modified so that the undesired echo components are completely removed from the signal transmitted to the opposite end. The basic concept is depicted in Figure 16. Both the loudspeaker signal and the microphone signal are first transformed into the frequency domain by a spectral transform (ST). Based on these input signals, the AEC determines an optimal gain factor for each individual frequency band separately in order to remove the echo components. After applying this echo attenuation filter to the spectral representation of the microphone signal, the echo-free signal is transformed back to the time domain. In typical application scenarios, robust attenuation of the echo by 60 dB can be expected and is also achieved reliably.

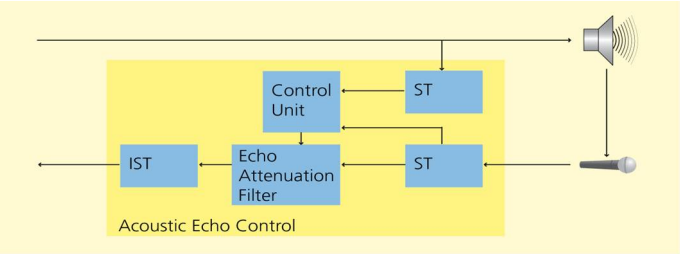

Figure 16: Basic approach to acoustic echo control in the TA2 project

The computation of the optimal gain factor is based on an estimate of the power spectrum of the echo signal captured by the microphone. The power spectrum of the echo is determined by applying an adaptive estimate of the acoustic echo path to the known power spectrum of the loudspeaker signal.

All estimation algorithms used within the current AEC rely solely on the *power or magnitude* spectra of the loudspeaker and microphone signals. Since the phase information of the signal spectra are discarded, the performance of this approach is independent of any phase changes or distortions introduced by the acoustic echo path. As an immediate result, the AEC becomes insensitive to effects like time drift due to sampling rate mismatch. This problem typically arises when the loudspeaker signal and the microphone signal are captured using different soundcards or A/D converters leading to asynchronous sampling.

The approach used in the TA2 project does not require an exact identification of the impulse response of the acoustic echo path. Thus, it is also very robust against movements of the microphone or other changes in the acoustic environment.

To provide the best possible speech quality, modern telecommunication systems often use audio coder such as AAC-ELD to compress the signals before transmission. Typically, these codecs are operated at sampling rates of 44.1 or 48 kHz to include the full audio bandwidth. To be also applicable in high quality communication systems, the AEC supports sampling rates of up to 48 kHz.

In order to achieve a low complexity implementation, which is especially important when operating at high sampling frequencies, the AEC uses a reduced frequency resolution compared to, e.g. Fourier transform based approaches. The spectral smoothing is performed in critical bands, i.e., in accordance with human perception. It is applied to the loudspeaker spectra, the microphone spectra, and the model of the acoustic echo path. By doing so, the echo components can be represented by much less than a hundred parameters as opposed to the multiple of thousands spectral values of an exact Fourier representation. As an example, Figure 17 depicts the magnitude frequency response of a room impulse response measurement at a sampling rate of 44.1 kHz (blue) together with the smoothed version (red). The black squares indicate the values to be determined by the acoustic echo control algorithm.

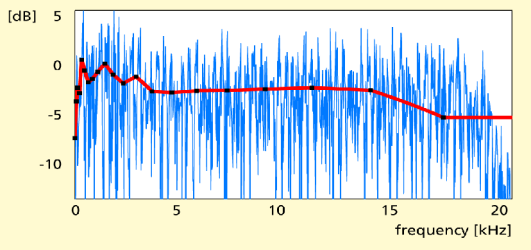

Figure 17: Magnitude frequency response of a typical room impulse response and its simplified version using perceptually motivated frequency domain smoothing

The AEC approach is not limited to single channel communication only. It also supports stereo operation or an even larger number of loudspeakers and microphones. For example, common 5.1 home theatre systems can be used for playback of surround sound in natural spatial audio communication applications as considered in the TA2 project.

The general concept of the echo suppression described above can be extended to multiple loudspeaker and microphone channels straightforwardly. Due to an efficient combination of the loudspeaker and microphone channels, the CPU consumption scales insignificantly for increased number of channels.

The AEC implementation additionally includes a noise reduction module to remove undesired stationary background noise captured by the microphone. Speech is known to be a highly non-stationary signal, which can be separated from stationary noise sources like fan noise or microphone noise. The noise reduction is performed via time-variant attenuation of spectral components of the microphone signal, where the level of attenuation depends on the estimated instantaneous signal-to-noise ratio at the microphone. The spectral resolution and the noise reduction gain computation takes into account relevant properties of the human perception of sound to achieve high quality audio output signals. The integration of the noise reduction module into the AEC module is advantageous, since a joint optimisation of the suppression of stationary echo signals and the reduction of stationary background noise can be achieved. The level of noise reduction can be adjusted to given preferences. Although an attenuation of up to 15 dB is possible, a limit of 10 dB is recommended.

In hands-free communication where the talkers are located far away from the microphones, the problem of too low speech levels often occurs. In this case, the level of the microphone signal should be appropriately adjusted by an automatic gain control (AGC). To avoid any undesired amplification of echo components, an AGC has been integrated directly into the AEC. The amplification of the microphone signal is controlled by the AEC and only amplifies local speech segments.

2.6 Evaluations

In this section we evaluate the performance of the algorithms discussed earlier in this chapter.

Figure 18 shows the localisation results for a setup where two fixed sound sources are placed at angles -75° and +75° with respect to the centre of the microphone array. The distance between the sensor array and the sound sources is approximately s = 80 cm. The sound (human speech) is captured in a relatively reverberant environment with a reverberation time of RT_{60} = 1.1 s. The first speaker (+45°) is exclusively active between 0 < t < 20 s, whereas the second one (-45°) is exclusively active between 20 s < t < 40 s. Moreover, there is a double talk situation during the last 20 s where both speakers are active. The settings of the localisation algorithm where chosen that the system provides a temporal resolution of about 174 ms. Both talkers could be localized for most time instances. However, the variance of the estimated source positions increased due to the stronger influence of the room reverberation. Nevertheless, the results are still reliable.

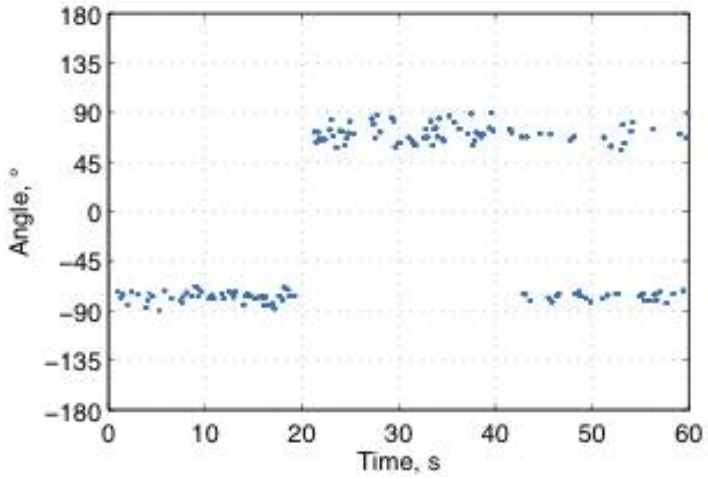

Figure 18: Localised source positions (s = 80 cm, $\varphi_{L,R} = \pm 75°$)

The following table shows the mean and variance of the estimated source positions for both talkers. It can be observed that on average, the source positions could be determined with relatively high accuracy in all cases. It should be mentioned that the variance of the estimated source positions usually increases for larger distances between the microphone array and sound source due to the stronger influence of room reverberation.

Setup	φ_L	φ_R	σ_L	σ_R
s = 80 cm, $\varphi_{L,R} = \pm 75°$	-77.2°	+71.2°	3.9°	8.0°

In an additional experiment, four talkers were sitting around the planar microphone array, where a D-grid configuration with microphone spacing of 4.5 cm has been placed on a table. The four persons are located at approximately -45°, 0°, 45° and 180°. Localising this relatively large number of persons is difficult especially when the persons are talking at the same time and when additionally room reflections and reverberation is present. Nevertheless, the localisation algorithm provides accurate results, as can be seen from Figure 19. It turned out that for each time frame the number of estimated sound sources does not exceed the true number of four. Notice that not all persons are talking at the same time explaining the gaps in the localisation results for the directions were the persons are located. Notice further that only few sound events are localised from wrong directions showing the robustness of the algorithm.

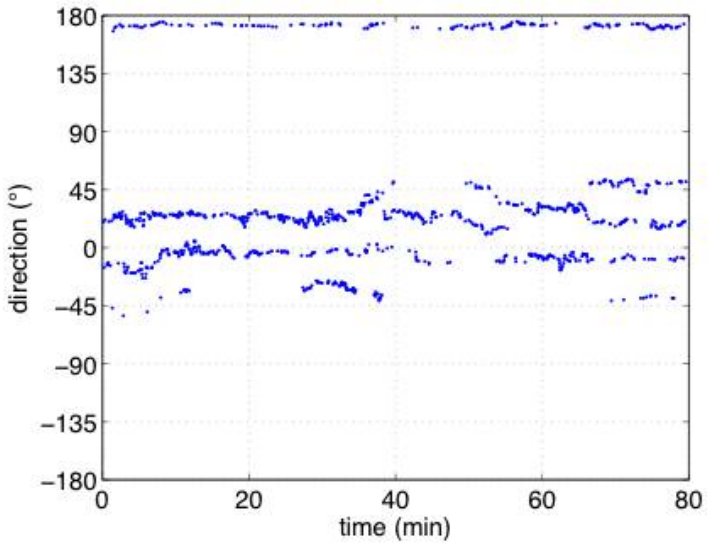

Figure 19: Source localisation results for four simultaneous talkers

To evaluate the performance of directional filtering according to section 2.3, we employed a listening test based measure to approximate the enhancement of speech intelligibility provided by interference suppression. A mere investigation of interference attenuation would not give insight into the amount of disturbing artifacts. During the listening test we measured the signal-to-noise ratio (SNR), at which an average of 50% of the sentence was understood. If we additionally use an interference suppression algorithm (the spatial interference is referred to as noise), it should be possible to choose a lower input SNR to achieve the same level of speech intelligibility. The difference β_{win} of these two SNRs in dB can be interpreted as a kind of intelligibility improvement provided by the interference suppression algorithm. Details about the measure can be found in [14].

Figure 20 shows the aforementioned measure for intelligibility improvement provided by directional filtering simulation. The figure shows the mean and the standard deviation over a variety of sentences and listeners. We compared our method to a conventional minimum variance distortionless response (MVDR) beamformer which can be seen as a state-of-the-art approach to signal enhancement using microphone arrays. We varied the reverberation time τ_{60} between 50 ms (dry as in, e.g., a car) and 450 ms (typical office room). The desired source has been positioned at 0° and the interference at 60°, where both sound sources were placed at a distance of 1 m from the microphone array. The microphone array has been designed according to the D-grid discussed in section 2.1 with a spacing of 3.2 cm. As shown in [3] this spacing involves spatial aliasing beyond 7.5 kHz. The MVDR beamformer was designed on the basis of a four microphone line-array in broadside steering. We chose a spacing of 2.3 cm to reach the same aliasing frequency as in case of the D-grid. Note that all room impulse responses (RIRs) were simulated to have better control over the influence of different reverberation times.

As can be seen from Figure 20, a clear advantage using directional filtering compared to the MVDR beamformer can be observed for low and medium reverberation times. At very high reverberation times, the advantageous performance of the directional filtering is still noticeable, but not statistically significant compared to the traditional beamforming approach.

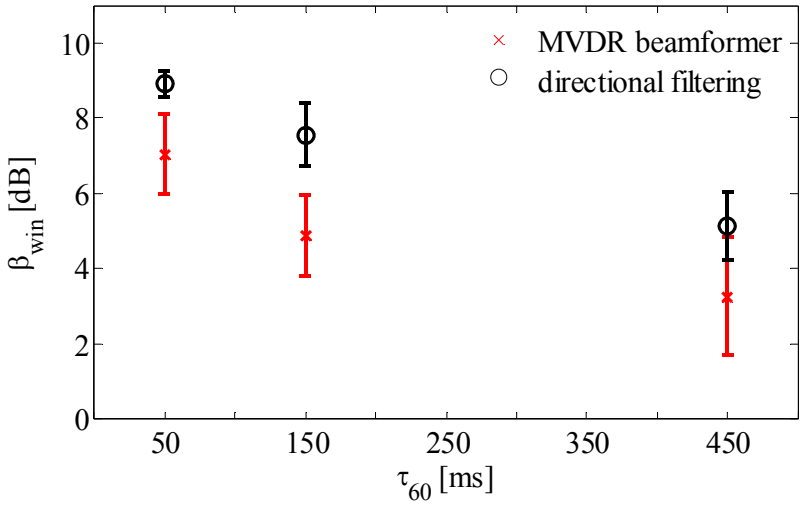

Figure 20: Speech intelligibility improvement provided by directional filtering compared to a conventional MVDR beamformer

Informal listening confirms the desired dereverberation effect: the speech sounds less reverberant and is perceived closer as in case of the unprocessed microphone signal. A comparison to standard beamforming methods for dereverberation using objective measures have been presented in [11].

Objective evaluation of dereverberation approaches is not as straightforward as evaluating processing units to remove additive disturbances like echoes or spatial interferences. The reverberant signal can be modelled to result from the convolution of a dry signal and a room impulse response (RIR). The early part of the RIR, i.e., the part, which models the direct propagation from the source to the microphone as well as some early reflections, is known to support speech intelligibility and, thus, have to be considered as desired signal components. The late part of the RIR generates diffuse reverberation, which acts as a disturbance on the desired signal. In the following evaluation, we use simulated RIR to be able to control the acoustic characteristics which are not accessible in recordings. We have simulated various RIRs and rated the late part after 25 ms after the direct part in the RIR as the undesired part. The convolution with this part produces a disturbing signal whereas convolution with the early part generates the desired reference signal.

For evaluation we have chosen the *perceptual evaluation of speech quality* (PESQ) measure, which was reported to be well suited to judge dereverberation [15]. PESQ quantifies the "perceptual distance" between two speech signals. Therefore, we performed a test to analyse the distance between dereverberated speech signals and a synthetic desired signal, in which we have decreased the energy of the late part of the RIR by 10 dB. The motivation for choosing this specific reference signals is that the goal is to obtain an enhanced signal which still contains early reflection while exhibiting a decreased amount of late reverberation. It should be noted that entirely removing the late reverberation components in a recorded speech signal leads to unnatural sounding output. The reference should be designed in a way that a very good enhancement algorithm can approximate it. Note that PESQ measures range between -0.5 (large differences between reference and sample under investigation) and 4.5 (no difference). We compared our dereverberation approach to a time-invariant MVDR beamformer which represents a widely used approach to enhance speech in reverberant environments. A time-invariant filter-and-sum beamformer, such as the MVDR beamformer, is well-known to significantly contribute to the attenuation of reverberant sound. However, in contrast to our proposed dereverberation approach, the beamformer has to be steered to a certain look-direction, i.e., the location of the desired talker. Both enhancement algorithms were based on a microphone array using the D-grid design with a spacing of 4.4 cm of opposing microphones.

Figure 21 shows the PESQ measure between the reference signal described above and dereverberated signals using the two algorithms. In addition, we investigated an unprocessed microphone signal.

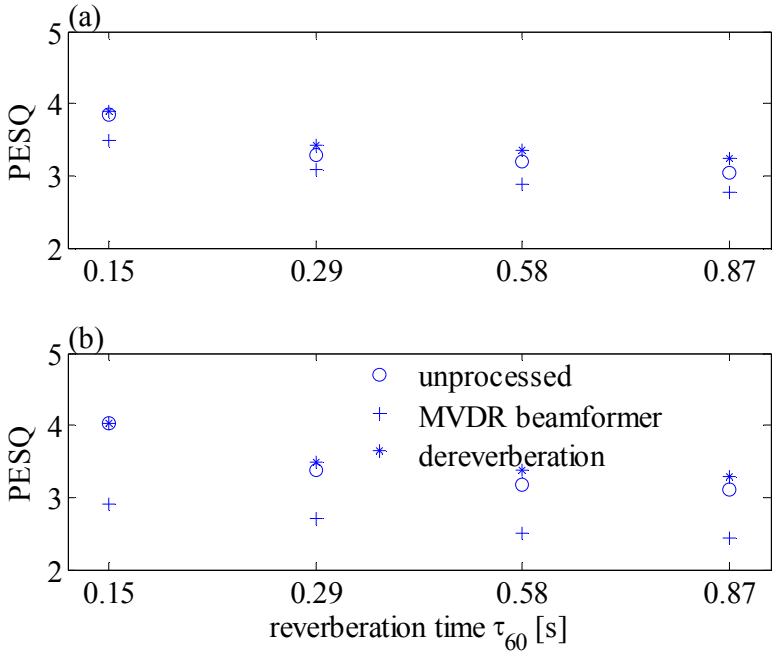

Figure 21: PESQ measure between a processed and a desired speech signal at various reverberation times:
(a) there was one source signal, to which the beamformer was steered,
(b) there was a double talk situation, where a second speaker was located at 60° outside of the beamformer's steering direction

For plot (a) we chose one active speech signal, where the beamformer was steered towards the source position. We can observe only minor improvements in terms of PESQ by our dereverberation procedure compared to an unprocessed microphone signal. Despite these slight objectively measured improvements, informal listening showed an improved sound quality compared the unprocessed microphone signal. In case of the MVDR beamformer, the speech sounded also less reverberant, but artefacts such as inherent low-pass filtering are observed, which is also reflected in the decreased PESQ measures.

One major advantage of the novel approach is its independence of any steering information. This is illustrated in plot (b): we added a second speaker at 60° relative to the beamformer's steering direction. Both sources had a distance of 0.7 m to the array. (In a living room such short distance can be achieved by embedding a microphone array in arc-lamp as shown in Figure 3.) We can still observe an enhancement by our dereverberation technique, whereas the MVDR beamformer is not capable of enhancing multiple spatially distributed sources simultaneously, but rather causes distortions compared to the reference signal.

Further, we exemplify the performance of the multi-channel acoustic echo control in a spatial audio communication scenario.

The general simulation configuration is shown Figure 22. For the playback of spatial audio, each location has five loudspeaker arranged in accordance to a conventional 5.1 surround set-up. The recording of the sound field is simulated using a planar microphone array of four omnidirectional capsules as described in section 2.1 with a spacing of 2 cm. The microphone arrays are illustrated in each of the two locations in Figure 22 by the four dots in a circle. In the scenario, two talkers are present in location A at ± 45°, whereas only one talker is participating in room B, located at 0°.

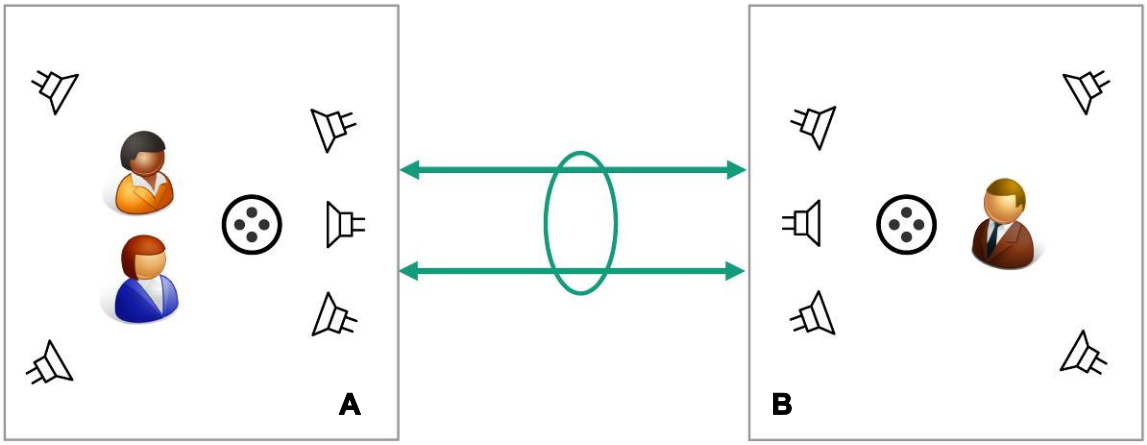

Figure 22: Source localisation results

The following conversation sequence has been used: In room A, the first talker is active for the first four seconds, followed by about 2.5 seconds' speech of talker B. In the next period of approximately 7 seconds, the talkers located in room A are silent, whereas the talker in room B is active. During the last 7 seconds of the conversation sequence a double-talk situation occurs, where the first talker in room A and the talker in the room B are active at the same time.

Let us now consider the resulting loudspeaker signals determined for playback in room A without acoustic echo control. In other words, the microphone signals in room B are kept unmodified and directly processed by the DirAC analysis and synthesis module. Figure 23 shows the loudspeaker signals computed for spatial audio reproduction at room A.

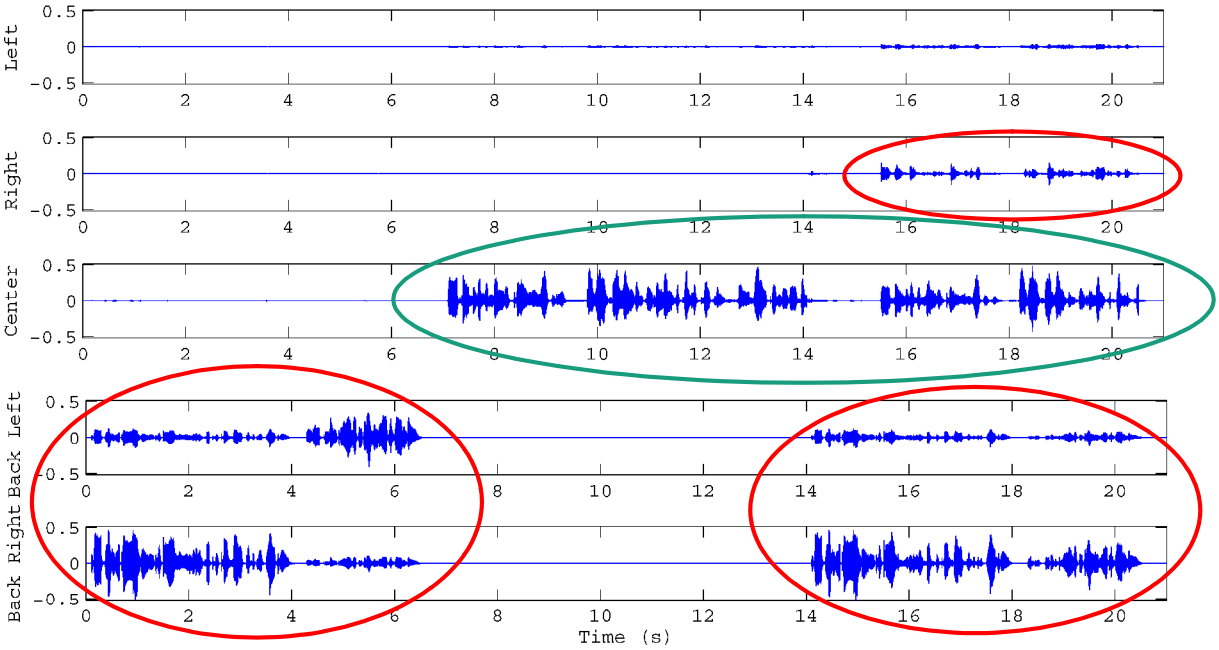

Figure 23: Source localisation results

The desired local signal in room B is marked by the green curve, whereas the undesired echo components are marked by the red curves. Note that the desired signal is played back mainly from the center loudspeaker, as expected from the recording setup in room. The echo signals are reproduced mainly from the loudspeakers in the rear, as this corresponds to the location of the loudspeakers as analysed by DirAC in room B.

In Figure 24, the corresponding signals as in Figure 23 are shown, but now with the acoustic echo control enabled in room B. More precisely, the signals captured by the microphone array are input to the echo control module together with the five loudspeaker signals as a reference for the echo removal. Then, the echo components are estimated and attenuated in the microphone signals. The echo controlled microphone signals are then processed by the DirAC module to compute the enhanced loudspeaker signals for spatial sound reproduction in room A.

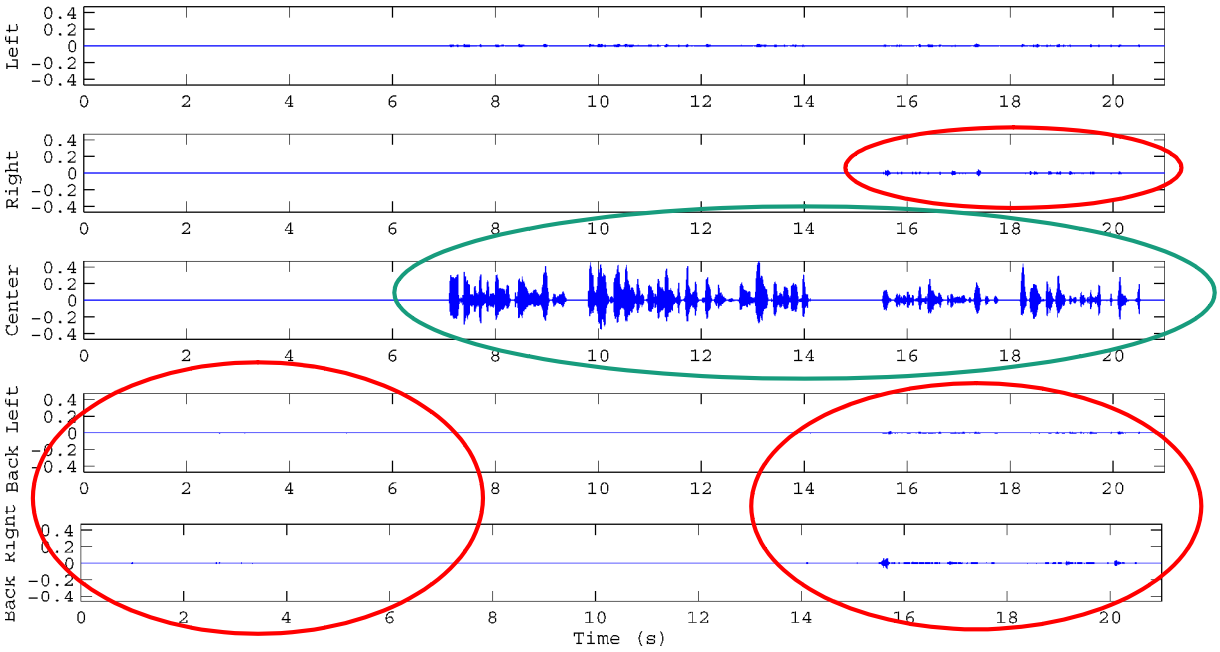

Figure 24: Source localisation results

In order to facilitate this informal evaluation, the curves indicating desired and undesired signal components are included in Figure 24 too. Comparing Figure 23 and Figure 24, it can be noticed that the echo components have been completely removed in the first part of the sequence, i.e., during speech activity in room A only. During the double-talk part in the end of the sequence, some very small residual echoes remain. However, these residual echoes are not audible, as they are masked by the simultaneous speech of the desired talker. It can also be noticed that during the double-talk period, the desired signal of the local talker in room B is also slightly attenuated. Although this results in audible distortion of the speech, the talker in room B can still be understood correctly. In the period between 7 and 14 seconds, only the talker in room B is active. The results shown in Figure 23 and Figure 24 illustrate that the corresponding desired speech signal is kept unmodified.

2.7 Conclusion

In this chapter, the different functionalities covered by the intelligent audio capturing module have been described. A central part is represented by a directional analysis of the observed sound field extracting parametric information such as the direction of arrival (DOA) of sound and the diffuseness of the sound field in frequency subbands. These parameters are then used together with the microphone input signals to perform different signal processing tasks such as:

- spatial audio capturing,
- localisation of multiple sound sources,
- directional filtering for one or more sound sources,
- dereverberation.

As each of these tasks is based on the same directional parameters, they can be efficiently integrated into a more general framework.

The directional information is not limited to be used only for spatial audio recording, but can be exploited as an additional technique for a higher-level analysis of the observed sound scene. This opportunity was evaluated as well and the corresponding results are included in section 3.12. Additionally, directional information can then be exploited for beamforming by directional filtering or for an efficient and interactive coding of spatial audio objects [10]. The information about the observed sound field (in combination with input data other than audio) can further be exploited to give a user an intelligent feedback and/or to derive semantic information.

The sound source localisation algorithm provides a high robustness against reverberation inside a room. Moreover, the number of sound sources which can be localised can be larger than the number of sensors of the microphone arrays.

In addition to the spatial audio related techniques, methods to enable hands-free communication in multi-channel environments have been developed. This includes the removal of distracting acoustic echoes as well as the reduction of undesired background noise.

Experimental results have been presented to illustrate the performance of the techniques developed within the TA2 project.

2.8 References

[1] V. Pulkki, "Spatial sound reproduction with directional audio coding", Journal of the Audio Engineering Society, vol. 55, no. 6, pp. 503–516, 2007.

[2] V. Pulkki and M. Karjalainen, "Localization of amplitude-panned virtual sources I: stereophonic panning", Journal of the Audio Engineering Society, vol. 49, no. 9, 2001.

[3] M. Kallinger, F. Kuech, R. Schultz-Amling, G. Del Galdo, J. Ahonen, and V. Pulkki, "Analysis and adjustment of planar microphone arrays for application in directional audio coding", in 124th AES Convention, paper 7374, Amsterdam, Netherlands, 2008.

[4] O. Thiergart, M. Kratschmer, M. Kallinger and G. D. Galdo, "Parameter estimation in directional audio coding using linear microphone arrays", in 130th AES Convention, London, UK, 2011.

[5] R. Roy and T. Kailath, "ESPRIT – estimation of signal parameters via rotational invariance techniques", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, no. 7, 1989.

[6] O. Thiergart, R. Schultz-Amling, G. Del Galdo, D. Mahne and F. Kuech, "Localization of sound sources in reverberant environments based on directional audio coding parameters", in 127th AES Convention, New York, USA, 2009.

[7] M. Kallinger, H. Ochsenfeld, G. Del Galdo, F. Kuech, D. Mahne, R. Schultz-Amling, and O. Thiergart, "A spatial filtering approach for directional audio coding", in 126th AES Convention, paper 7653, Munich, Germany, 2009.

[8] G. Del Galdo, V. Pulkki, F. Kuech, M.-V. Laitinen, R. Schultz-Amling, M. Kallinger, "Efficient methods for high quality merging of spatial audio streams in directional audio coding", in 126th AES Convention, preprint 7733, Munich, Germany, 2009.

[9] J. Breebaart, J. Engdegård, C. Falch, O. Hellmuth, J. Hilpert, A. Hoelzer, J. Koppens, W. Oomen, B. Resch, E. Schuijers, L. Terentiev, "Spatial audio object coding (SAOC) – the upcoming MPEG standard on parametric object based audio coding", in 124th AES Convention, preprint 7377, Amsterdam, Netherlands, 2008.

[10] J. Herre, C. Falch, D. Mahne, G. Del Galdo, M. Kallinger, O. Thiergart, "Interactive teleconferencing combining spatial audio object coding and DirAC technology", preprint 8098, in 129th AES Convention, London, UK, 2010.

[11] M. Kallinger, G. Del Galdo, F. Kuech, O. Thiergart, "Dereverberation in the spatial audio coding domain", in 130th AES Convention, London, UK, 2011.

[12] A. Favrot, et al., "Acoustic echo control based on temporal fluctuations of short-time spectra", in Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC), Seattle, USA, 2008.

[13] F. Kuech, et al., "Acoustic echo suppression based on separation of stationary and non-stationary echo components", in Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC), Seattle, USA, 2008.

[14] M. Kallinger, H. Ochsenfeld, and A. Schlüter, "A novel listening test-based measure of intelligibility enhancement", in 127th AES Convention, paper 7822, New York, USA, 2009.

[15] S. Goetze, E. Albertin, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "Quality assessment for listening-room compensation algorithms", in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 2450–2453, Dallas, Texas, USA, 2010.

3 Data Analysis for Synchronous Scenarios

In this chapter, we describe a low delay real-time multimodal cue detection engine for a living room environment (Figure 25). The system is designed to be used in open, unconstrained environments to allow multiple people to enter, interact and leave the observable world with no constraints. It allows the detection and tracking of multiple faces, identification and recognition of multiple persons, estimation of head poses, depth information and visual focus of attention, detection and localisation of verbal and paralinguistic events, detection of some hand gestures (pointing and wiping), as well as the detection, recognition and tracking of specified patterns (on a sheet of paper).

The extracted semantic information is exploited by the orchestration engine [1] to produce then an orchestrated video chat by choosing at each point in time the perspective that best represents the social interaction. Some of semantic information is exploited as well by the game engine as additional means of user controls.

The system takes inputs from spatially separated sensors. By placing the sensors at their individually optimal locations, we clearly obtain a better performance of low-level semantic information.

In the context, TA2 presents several challenges: the results need to be computed in real-time with low affordable delay from spatially separated sensors (as opposed to other systems, such as [2], [3], relying on collocated sensors) in open, unconstrained environment. Furthermore, the results are supposed to be localised in the image space to allow for a dynamic and seamless orchestrated video chat and interactions with the system.

Figure 25: Illustration of a family environment setup

3.1 A real-time architecture

The multimodal cue detection engine (analysis engine) architecture is built around several modules as illustrated in Figure 26 comprising a face detector, a multiple face tracker, multiple person identification, head pose and visual focus of attention estimation, an audio real-time framework with spatial localisation, a large vocabulary continuous speech recogniser and keyword spotter, depth estimation, pointing and wiping detectors, pattern detector and tracker, multimodal association and fusion.

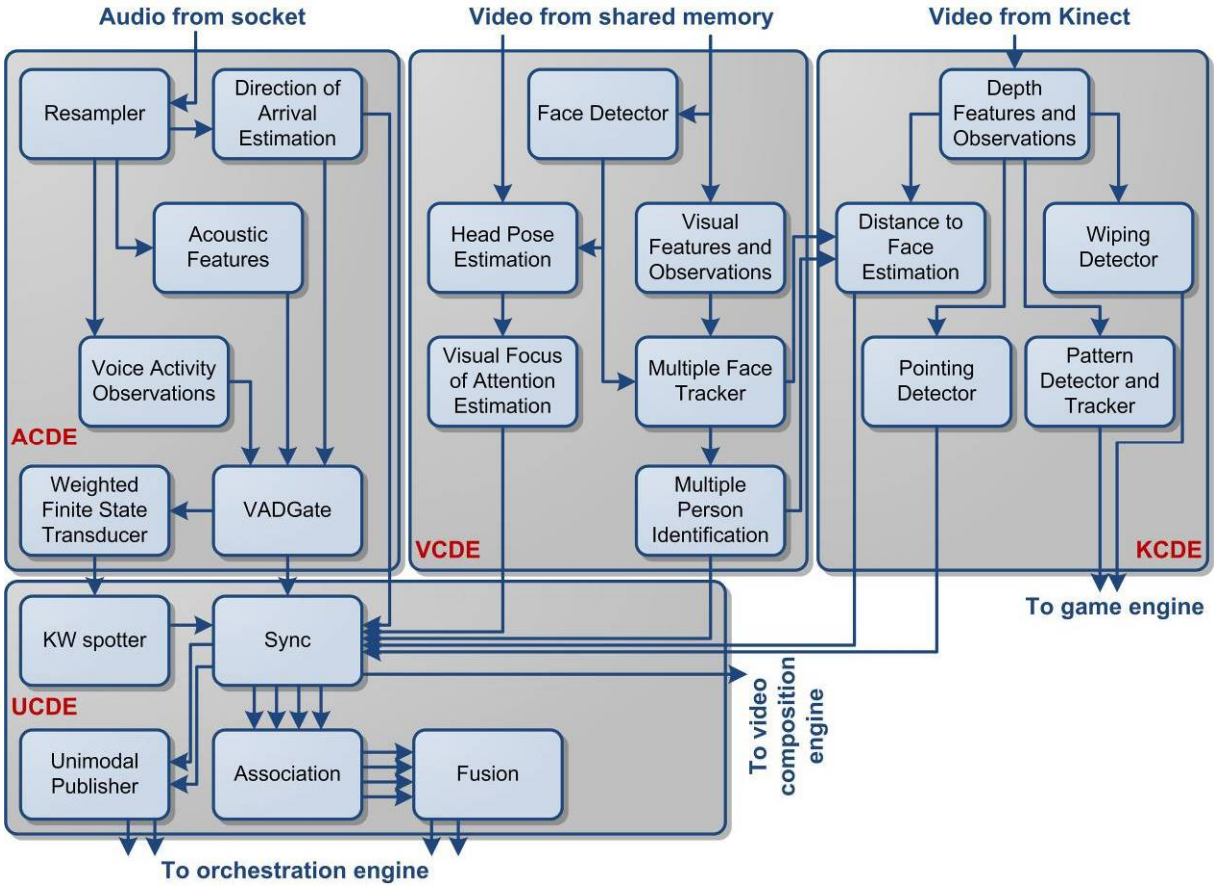

Figure 26: Real-time architecture of analysis engine

Audio input to and semantic output from analysis engine are done via sockets. Video input is done via shared memory for both video sensors (Sony EVI-HD1 and Microsoft Kinect). Note that most of involved algorithms rely on standard video sensor to ensure the best alignment of semantic information with teleconference video streams, while Kinect sensor is used for the algorithms which rely on depth map from Kinect depth sensor. The shared memory interface is used to distribute a full-resolution frontal camera view (Sony EVI-HD1), depth map and colour image from Kinect and their transformations between more than 6 separate analysis processes. Detected semantic cues are propagated through communication manager to the rest of the system and exploited within orchestration engine, composition engine and game engine.

The shared memory interface for distributing visual data between the analysis processes was made available as an open source C++ library[1]. The interface is intended primarily for real-time processing and supports multiple named image queues of unconstrained length. The images are passed through the interface in an efficient way: they are directly allocated in the shared memory, not copied to it. The analysis processes access the images directly in the shared memory. There are no constraints in access to the image queues – multiple processes can write or read the same queue and any process can access any queue. A process can also indicate which queues it needs as an input at a certain point of time allowing for dynamic allocation of computational resources. The interface uses the Boost library for inter-process communication, and it is compatible with OpenCV image structures.

In addition to the shared memory interface, a pre-processing module for image transformations was developed to allow common transformations to be computed only once and with high efficiency which is achieved by employing the Intel Integrated Performance Primitives library. The module supports scaling, cropping, colour-space conversions and other image processing operations.

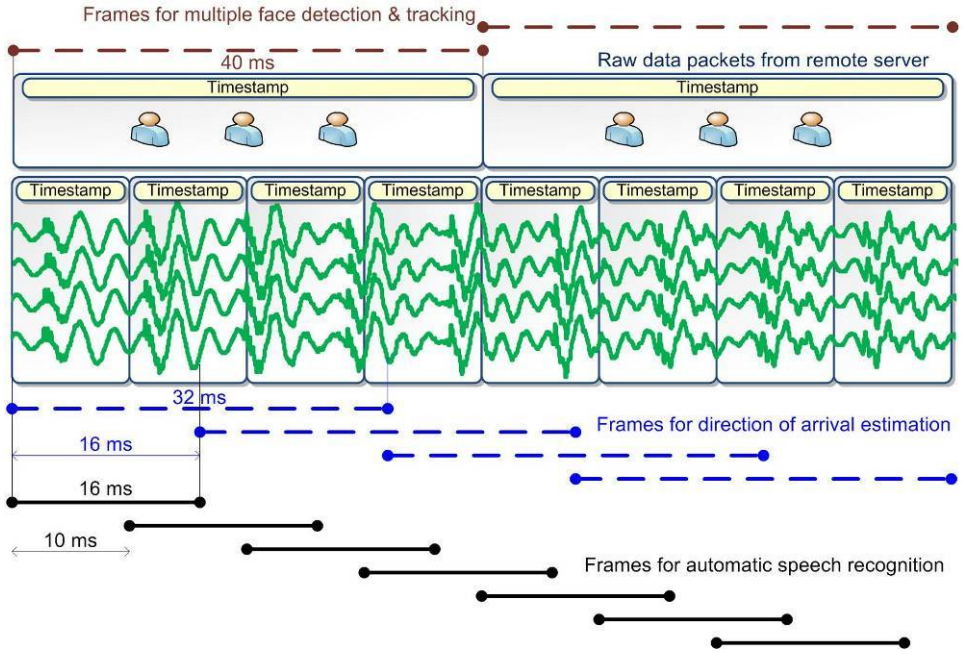

Figure 27: Framing for online processing

The multimodal processing operates in multi-framing mode (Figure 27) with non-overlapping video frames at variable frame rate for video processing, overlapping audio frames of 16 ms in step of 10 ms for voice activity detection and automatic speech recognition, and overlapping audio frames of 32-128 ms in step of 16-64 ms for direction of arrival estimation.

[1] http://www.fit.vutbr.cz/research/prod/index.php.en?id=209 (or contact Michal Hradiš ihradis@fit.vutbr.cz)

3.2 Long-term multiple face tracking and person identification

A multiple face tracking algorithm is automatically initialised and updated using outputs from a standard face detector [4]. The scenario of interest raises a number of challenges for online multiple face tracking:

1) Faces may not be detected for longer periods of time when persons focus on the table or touch screen in front of them (for example when playing a distributed game).
2) When more than two persons are present, they tend to occlude each other more often (see Figure 28), leading thus to more frequent track interruptions.
3) The lighting conditions and scene dynamics are less controlled in a living room environment (than for example in a meeting room).
4) The assignment of consistent IDs to persons is important for further reasoning and automatic stream editing.
5) The processing has to be in real-time and with a low delay.

However, in the TA2 scenario it is necessary to know at each time instant where the people are in the video scene.

Figure 28: An example of difficult to detect head poses and partial occlusions [5]

The solution employed in this component is based on a multi-target tracking algorithm using Markov Chain Monte Carlo (MCMC) sampling, similar to [6]. This is a Bayesian tracking framework using particles to approximate the current state distribution of all visible targets. At each time step, targets are added and removed using the output of an additional probabilistic framework that takes into account the output of the face detector as well as long-term observations from the tracker and image [7].

The main contributions in this regard are the following:

1) A state-of-the-art online multiple face tracker in terms of precision and recall over time.
2) A probabilistic framework for track creation and removal that takes into account long-term observations to cope with false positive and false negative detections [6].
3) A robust and efficient person re-identification method.

The proposed tracking algorithm relies on a face detector [4] with models for frontal and profile views. For efficiency reasons, the detector is applied only every 10 frames (i.e. around 2 times per second). Also, to improve execution speed and reduce false detections, the detector is only scanning image regions with skin-like colours using the discrete model from [8] as a prior and adapting it over time by using the face bounding boxes from the tracker output.

As face detections are intermittent and sometimes rather rare, a tracking algorithm is required. Its goal is to associate detections with tracked objects, to associate tracked objects with persons (person IDs), and to compute an estimate of the number and position of visible faces at each point in time. We tackle the tracking problem using a recursive Bayesian framework. The state estimation is implemented using a particle filter with a Markov Chain Monte Carlo (MCMC) sampling scheme [6]. The essential components of the particle filter are described in the following. For more details about the MCMC implementation refer to [7].

The *state space* is the concatenation of the states of all visible faces, where the state of each single face is a rectangle described by the 2D position in the image plane, a scale factor and the eccentricity (height/width ratio).

The overall *state dynamics* is defined as the product of an interaction prior and of the dynamics of each individual visible face. Note that the creation and deletion of targets are defined outside the filtering step (see below). The dynamics of visible faces are described by a first-order auto-regressive model for the translation components and a zero-th order model with steady-state for the scale and eccentricity parameters. The interaction prior prevents targets to become too close to each other.

As a trade-off between robustness and computational complexity, we employ relatively simple but effective *observation likelihood* for tracking based on multi-level quantised colour histograms in the HSV colour space (see [7] for more details). The overall observation likelihood is defined as the product of likelihoods of each individual visible face.

Target candidates are potentially added and removed at each tracking iteration. Classically, face detectors have been used to initialise new targets and targets are removed when the respective likelihood drops. However, face detectors can produce false detections, and, in our scenario, faces may remain undetected for a longer time due to non-frontal head poses over extended periods. Therefore, we use long-term observations and a probabilistic framework [7] including two hidden Markov models (HMM), one helping to decide on track creation and one to decide on removal.

Whenever the track of a person is lost and re-initialised later, or when a person leaves the scene and then comes back, we would want to assign the same identifier (ID) to that person. This is not done directly “inside” the tracking algorithm but on a higher level, taking into account longer-term visual appearance observations. More specifically, the person model is composed of two colour histograms: a face colour histogram and a shirt colour histogram, as well as a long-term history of previous face positions in the image. The structure of the histogram models are similar to the one used for the observation likelihood in the tracking algorithm.

If a target is added to the tracker and there is no existing person model that is un-associated then a new person model is initialised immediately and associated to the target. Otherwise, the face and shirt colour histograms of the new target are computed recursively over successive frames and stored. Then, after this period, we calculate the likelihood of each stored person model given an unidentified candidate. The probability is a distribution over possible identities at the candidate position. This distribution is updated linearly at each time step and for each image position according to the history of tracked target positions.

A given person is then identified by simply determining the person model with the maximum likelihood. If not, a new person model is created and added to the stored list. All associated person models are updated at each iteration with a small factor. The candidate models are updated with a higher factor. The evaluation results of proposed tracking method are summarised in section 3.12.

3.3 Head pose and visual focus of attention estimation

Based on the output of the face tracker, the head pose (i.e. rotation in 3 dimensions) of an individual is estimated. The purpose of computing head pose is the estimation of a person's visual focus of attention, which within the context of this work is constrained to being one of the video conferencing screen, the touch sensitive table, or any other person in the room.

Head pose is computed using visual features derived from the 2-dimensional image of a tracked person's head. The features used here are gradient histograms [9] and colour segmentation histograms. The colour segmentation features are estimated from an adaptive Gaussian skin colour model which is used to classify each pixel around the head region as either skin or background. More advanced colour segmentation has also been explored, where four classes are used: skin, hair, clothing or background. The posterior class probability of each pixel is estimated using spatial priors as well as colour priors which are adapted to each individual being tracked [8].

To compensate the variability in the output of the face tracker, the 2-dimensional face location is re-estimated by the head pose tracker. This serves to normalise the bounding box around the face as well as possible, while simultaneously using the visual features mentioned above to estimate a pose. This joint estimation of head location and pose improves the overall pose accuracy [10].

Given the estimated belief (probability distribution) over head pose, the visual focus of attention target is estimated. The range of angles that correspond to each target is modelled using a Gaussian likelihood. This likelihood is derived from the known spatial locations of the targets within the room. The posterior belief over each target is computed with Bayes' rule using the method of [11].

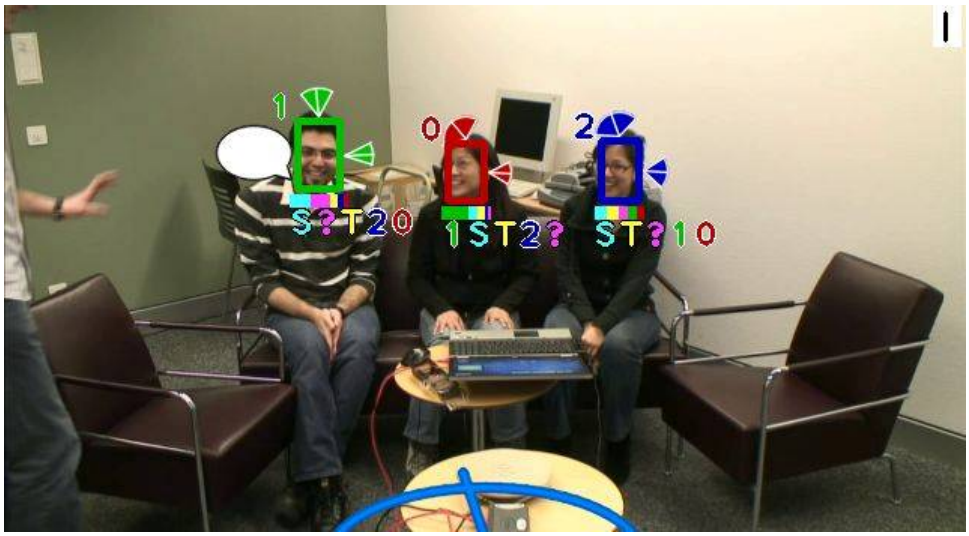

Figure 29: Multimodal cue visualisation information:

- **ID (at the top-left of the face bounding box) of each person,**
- **the head orientation estimation, i.e. pan and tilt, with a variance indication (on the top and right side of the box) for each person,**
- **the estimated distribution over targets where the person is looking at (at the bottom of the box), where the left-most target is the most likely one (the letter "S" means "screen", "T" means "table", "?" means "unknown", and the numbers correspond to the IDs of the other persons),**
- **the blue line in the bottom of the image is estimated direction of arrival of sound,**
- **the speech bubble indicates that a person is speaking,**
- **the output of the keyword spotting in the top-right of the image, here the word "I"**

3.4 Head motion estimation

Many existing works proposed to use visual cues for speaker detection in videos or other audio-related tasks (e.g. [12], [13], [14], [15]). Most of these works, try to detect people's lip motion. Naturally, this is indeed likely to be an informative visual cue for determining if a person is speaking or not. However, there are several drawbacks with this approach:

- Lip motion estimation requires a relatively precise localisation of the mouth region. This is a challenging task when lighting conditions are not controlled, when head pose varies largely, and when the (face) image resolution is low. In some scenarios, the mouth region might not even be visible because of occlusion (e.g. by the hands) or extreme head pose (e.g. looking down).
- The robust and precise detection of lips in an image is computationally complex in a multi-person, real-time scenario.

To overcome these drawbacks, we make use of the fact that when people speak they move or behave in a different way. Generally speaking, people who speak move their body more. Therefore, a relatively simple and efficient visual cue based on the amount of head motion can be used. Here, we leverage the fact that face tracking (described in section 3.2) provides us with the face regions of the visible persons. From these regions, it is straightforward to efficiently and reliably extract the overall head motion.

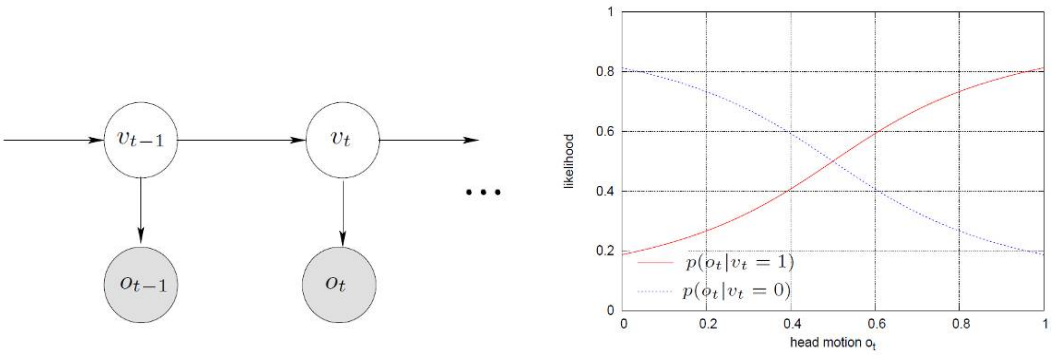

Figure 30: (left) The HMM used for each person to estimate voice activity from visual cues; the hidden, binary variable v_t indicates if the person is speaking or not; the probability of v_t is estimated recursively using the previous estimate v_{t-1} and the current observation o_t; (right) sigmoid functions defining the observation likelihood of head motion for v_t=0 and v_t=1

In order to incorporate visual observations over a more extended period of time, i.e. not frame-by-frame, we propose a simple hidden Markov model (HMM) that estimates the probability of a hidden, binary variable with the states "a person speaks" / "a person does not speak" at each time moment. Figure 30 (left) illustrates this model. We deliberately modelled this binary variable for each person independently because we don't want to impose any constraints regarding the interaction of persons at this stage but rather at the audio-visual processing level. The observation is the estimated head motion amount for a given person that is the mean motion magnitude inside the face region, based on the displaced frame difference between the pixel intensities in two successive frames. The observation likelihood is defined by two symmetric sigmoid functions with the parameters from separate training data (illustrated in Figure 30 (right)).

Finally, the posterior probabilities of each person and at each time step constitute the visual part of the features that is used in the subsequent multi-modal classification step. Note that, for simplicity and general applicability, we currently do not train this model for specific persons nor adapt it online. This could improve the overall results, but might also lead to over-fitting and drift.

3.5 Direction of arrival estimation

The speaker localisation is performed by the direction of arrival module (Figure 26). Typically, speaker localisation can either be done in the audio modality, video modality or multimodality. The first one implies a microphone array usage, while the second one is based on movement detection. Multimodal localisation allows results to be less affected by noise in the audio modality, although it increases significantly the CPU load. In this section we describe employed audio modality based direction of arrival algorithm, while multimodal approaches are discussed in section 3.12.

To achieve seamless low delay real-time performance the audio spatio-temporal fingerprint processing algorithm (presented in [16]) was implemented and evaluated as a plug-in for the data-flow architecture Tracter [17]. Data-flow is a well established signal processing technique that represents individual processing elements (plug-ins) as vertices in a directed graph. The data is propagated through the graph using a "pull" mechanism, instigated by the sink.

The instantaneous spatial fingerprints [16] are defined as bit patterns of overlapping sector-based acoustic activity measures, where each sector is represented by 1 bit of information. The corresponding instances in time refer to processing frames of 32-128 ms length.

Each sector (bit) is defined as a 36° wide and 60° high (from the horizontal plane) connected volume of physical space around the microphone array. The sectors are taken in the horizontal plane in steps of 6°. This results in a total of 60 sectors. Wider sectors in smaller steps allow to avoid jittering of acoustic directions and smooth acoustic tracking of dynamic sources.

The sector activity measure [18] is defined as integrated within the sector point-based steered response power with phase transform weighting (SRP-PHAT). SRP-PHAT [19] in turn is defined as the sum of generalized cross correlations with phase transform weighting (GCC-PHAT [20]) for each microphone pair. Generalized cross correlation with maximum likelihood weighting (GCC-ML) is theoretically more optimal in the presence of uncorrelated noise, nevertheless its performance degrades with increasing reverberation [21]. In addition, it requires the spectral information of the noise from the preceding noise-only frames. GCC-PHAT is more robust against reverberation [22] due to the whitening of the microphone array signals. Also, it does not require any information about precedent noise levels. Therefore, it is more suitable for TA2 scenarios. Further, a sparsity assumption is applied for each frequency bin via minimisation of phase error and the sector activity measures are normalised by the volume of the sector.

Each sector activity measure is threshold to keep a binary decision, which gives 60 bits of data for 360° spatial representation per each instance in time. This information is stored as one 64 bit integer value.

Finally, the spatial fingerprint is multiplied by the predefined "zone of interest" mask. This multiplication results in directional filtering of the predefined areas of interest, elimination of unnecessary post calculations and outlier removal. It can be very helpful in the case of interconnected environments, where audiovisual links do not have an echo suppression mechanism. For example, remote parties can be shown on a TV screen (Figure 25) while the corresponding TV zone is out-of-interest for local diarization (distributed diarization can be driven as the superposition of local diarizations from all interconnected environments).

The spatio-temporal fingerprint representation is defined as an array of temporally connected spatial fingerprints taken in steps of 16-64 ms. This results in a 2D bit pattern (Figure 31) with a total of 62.5 columns per second and the low bit rate of 500 bytes/second (62.5 long integer values of 64 bits each). The spatio-temporal fingerprints are defined as subsets of the spatio-temporal fingerprint representation (the length depends on the application and can vary from 32 ms to several seconds).

The intersection fingerprint is defined as an intersection in the time domain of all elements within a spatio-temporal fingerprint. Similarly, the union fingerprint is defined as a union in the time domain of all elements within a spatio-temporal fingerprint. The resulting intersection and union fingerprints are normalised at each time instance by keeping one centralised bit per active source.

Due to the compact fingerprint representation we can benefit from the exploitation of multiple (60) data streams against a single instruction stream to perform operations, which may be naturally parallelized. This approach is widely used in many areas of Information and Communication Technologies (ICT) nowadays and is often referred as single instruction, multiple data (SIMD) streams according to Flynn's taxonomy [23]. In this section the SIMD approach is exploited for most of the bitwise operations (e.g., intersection, union and normalisation operations are represented via computationally efficient bitwise AND, OR and XOR operators).

The intersection fingerprints are used for continuous tracking of acoustic sources.

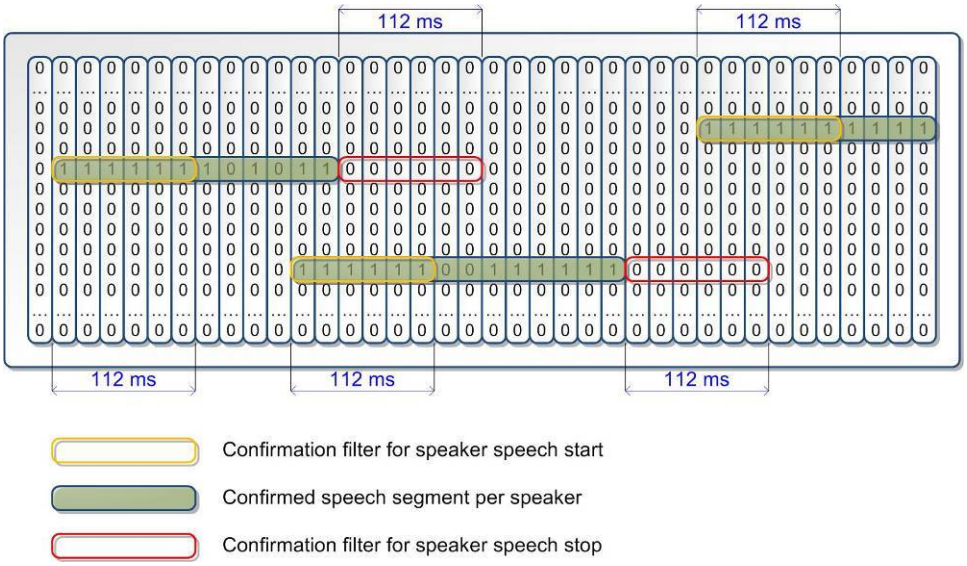

Figure 31: Spatio-temporal fingerprint processing

To allow low delay real-time and seamless diarization, transitions between states are used as additional triggers for voice activity events. The transition into a state with a lower number of acoustic sources is performed based on a union fingerprint, while the transition into a state with a higher number of acoustic sources is performed based on an intersection fingerprint.

The number of simultaneous acoustic sources is estimated as the Hamming distance between the last confirmed fingerprint and 0. The corresponding spatial locations of the active sources are computed as bit positions inside the confirmed intersection fingerprint multiplied by 6°. Taking into account that acoustic sources are not static in general, we cannot apply the Hamming distance between two intersection fingerprints to segment voice activity in respect to the turn taken. The voice activity is segmented in accordance with turn taken only in the case that a shift of the active source bit position from the previous confirmed state is higher than the predefined threshold. Otherwise the state is updated with the newer location without segmenting voice activity.

3.6 Voice activity detection and keyword spotting

The voice activity detection (VAD) covers both verbal and paralinguistic activities and is implemented as a gate. Downstream from the gate, the automatic speech recognition (ASR) is unaware that VAD is happening. It just receives segmented data in the same manner as if it was read from a sequence of pre-segmented utterances. Upstream from the gate, however, the data is actually one continuous stream. The gate segments the input stream in accordance to directional and voice activity / silence information from an algorithm based on silence models [24] or trained multi-layer perceptrons (MLP) using traditional ASR features. More precisely, in spite of the use of trained multi-layer perceptrons, which represents the favoured algorithm in ASR applications [25], [26], the current implementation uses adaptively thresholded energy coefficients and directions of arrival to perform localised voice detection. The algorithm works similarly to the traditional VAD module standardized by ETSI for speech coding (AMR1 and AMR2 techniques [27]) and benefits from low complexity and relatively small delay in comparison to more complex VAD techniques, e.g. a MLP based VAD system [28]. The directional information is used to additionally segment voice activity based on a spatial change between active sources and to filter out the acoustic events, coming from out-of-interest zones. The voice activity information is augmented by a confidence estimate (a probability of an audio segment being classified as voice segment) based on a distance from the decision threshold.

Similar to the detection of speech from audio input, we also considered to exploit visual information provided by head motion algorithm (see section 3.4). The corresponding confidence value represents the amount of motion at a given time instant and can be used in two ways: (a) replace the confidence of an appropriate speech segment detected in audio stream; (b) provide video-based speech detection, where speech segments and related confidence values are extracted using visual motion algorithm. The visual-based VAD can perform reasonably well in noisy audio conditions.

Finally, an approach considering both sources of information extracted from audio as well as video streams was employed. More specifically, several ways to efficiently combine audio and video-based VADs in a real-time scenario were explored. In one case only confidence scores were considered, in another case both confidence scores together with time-boundaries of speech segments were considered to be fused into audio-visual VAD.

The design is not necessarily the most efficient possible because the VAD logic must confirm that speech has begun, typically by waiting for a minimum period of time, before the VAD gate will issue the event and let the appropriate audio frames downstream towards ASR. However, the design is otherwise very flexible and allows the decoder to be developed completely offline and independently.

In [29], a data driven approach has been proposed to perform robust speech/non-speech detection in noisy environments. The optimal weights obtained using linear discriminant analysis (LDA) are used to weigh the temporal context of the signal energy. The discriminant obtained using LDA is interpreted as a filter in the modulation spectral domain. Experimental results obtained for various SNR levels show that the proposed method achieves improvements over ITU G.729B [30], ETSI AMR1, AMR2 systems [27] as well as the state-of-the-art MLP-based system [28].

The ASR component performed by ACDE enables speaker-independent large vocabulary based voice commands and keywords spotting. The spotting is performed based on the predefined list of participants and keywords relevant to the given scenario (e.g., orchestrated video chat). In a strict sense, ASR performs the conversion of a speech waveform (as the acoustic realisation of a linguistic expression) into words (as a best-decoded sequence of linguistic units). More specifically, the core of TA2 ASR system is represented by the weighed finite state transducer (WFST) based token passing decoder known as Juicer [25]. It is an HTK compatible WFST based token passing decoder and architecturally a Tracter sink. This means that any Tracter graph can be used seamlessly for feature acquisition. The decoder is also capable of continuous decoding. The WFST composition process has been overhauled to allow composition of very large transducers. The result is a faster and more capable decoder that can operate directly on high-order language models in real-time.

In general, ASR decoder is more computationally expensive component compared to the front-end or feature-extraction components. Whilst the decoder is based on a request-driven architecture, the analogues to digital converters (ADCs) are generally interrupt driven. That is, they write data into a buffer interrupting the host CPU when enough data is available. This is fundamentally a data-driven architecture.

Analysis data flow framework [17] is, in its simplest form, an interface between the decoder's pull architecture and the ADC's push architecture. This framework allows for a directed graph of components that are all pull driven. In an ASR system, this allows the design to be decoder-centric. Following the decoder-centric design, the decoder is unaware that VAD is running. From the decoding point of view, data simply starts and stops, following on from which the front-end is reset and data starts again.

Due to the real-time constraints required by the TA2 system, the spotting of keywords is currently performed on 1-best output obtained from the ASR decoder. However, together with improving the efficiency of the current approach, we have experimented in parallel with other approaches to more accurately detect predefined keywords. Although these algorithms are currently implemented to run in offline mode, their use for real-time detection is feasible.

3.6.1 Keyword spotting based on LVCSR/STD system

One of the most accurate algorithms used to detect large number of keywords (in this context, the task is referred to as spoken term detection (STD)) employs a large vocabulary continuous speech recognition (LVCSR) system. In fact, the spoken terms (keywords) are usually not known a priori. A common approach is to split the task into two stages. Firstly, a LVCSR system is used to generate a word or phone lattice. Secondly, a lattice search is performed to determine likely occurrences of the search terms. STD systems based on word lattices provide significantly better performance than those based on phoneme lattices (e.g., [31]). The word lattices can be associated with a confidence measure for each word. Traditionally, forward-backward re-estimation is used to calculate a confidence measure using word posterior probability conditioned on the entire utterance. Although such an STD system does not deal with out of vocabulary words, taking into account phone recognition lattices generated by the same LVCSR system can solve the problem. More precisely, the LVCSR system can be used to generate word (or phoneme) recognition lattices. The word lattices are then converted into a candidate term index accompanied with time and detection scores. The detection scores are represented by the word posterior probabilities, estimated from the lattices using the forward-backward re-estimation algorithm.

In our experimental work, the robust hypotheses outputs were achieved using a 3-pass LVCSR system, employed, based on the conversational telephone speech (CTS) system, derived from AMIDA LVCSR [32]. 250 hours of Switchboard data was used for training hidden Markov models (HMMs). The decoding was done in three passes, always with a simple bigram Katz back-off language model. In the first pass, perceptual linear prediction features (accompanied with delta coefficients) were used and processed by heteroscedastic linear discriminant analysis to perform a robust data-driven dimension reduction. HMMs were trained using a minimum phone error procedure. In the second pass, vocal tract length normalization was employed on similar features to the first pass. In addition to heteroscedastic linear discriminant analysis, a minimum phone error and speaker adaptive training were applied. Finally, the third pass was similar to the second pass, except input perceptual linear prediction features were replaced by posterior-based features estimated using a neural network system. The neural network processes 300 ms long temporal trajectories of mel filter bank energies. The neural network is represented by a multi-layer perceptron with one hidden layer (500 neurons). The LVCSR system reaches a word error rate of 2.9% on the Wall Street Journal (WSJ1) Hub2 test set from November 1992 (2.5 hours, with 5K dictionary and a trigram language model).

The LVCSR word lattice based STD system was evaluated on three hours of two channel CTS English development database distributed by NIST for the 2006 Spoken Term Detection (STD06) task [33]. The speech recordings were first segmented into shorter sub-segments using a speech/silence segmentation algorithm which removed around 50% of the data. Then, word lattices were generated using the previously described LVCSR system with a dictionary containing 50K words. The generated bigram lattices were subsequently expanded with a trigram language model. One-half of the 1107 English search terms were randomly selected from the list defined for the dry run set distributed for the STD06 evaluation. False alarm probabilities and miss probabilities in the STD task were evaluated. Figure 32 shows performance using standard detection error trade-off (DET) curves.

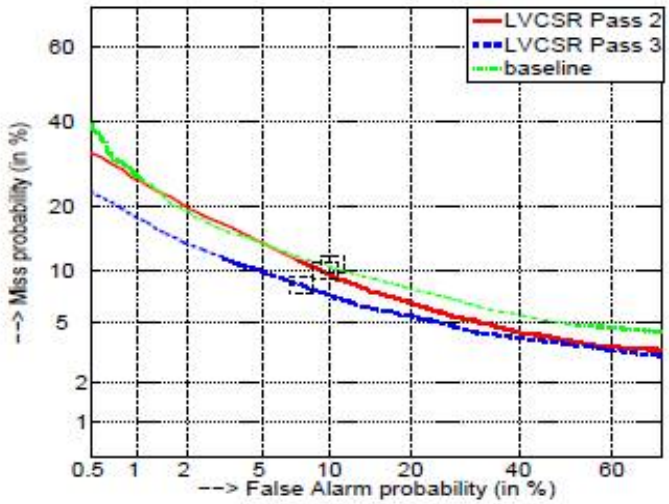

Figure 32: Performance of STD system (the boxes highlight EER – operating points)

In addition, we present equal error rates (EERs), a one number metric, mainly used to optimise the system performance. One can see that word lattices generated in the third pass provide significantly better performance than those in the second pass. STD performance is also compared to the baseline system described in [34]. The baseline system achieves EER of about 10.1%. The STD built on 3-pass LVCSR gives EER about 8% (20% relative improvement).

Besides EER and DET, we use a term-weighted value evaluation measure defined by NIST STD06 [33], which is also a one number metric. The term-weighted value is estimated first by computing the miss and false alarm probabilities for each term separately, then using these and a pre-determined prior probability to compute term-specific values, and finally averaging these term-specific values over all terms to produce an overall system term-weighted value. In particular, we use maximum term-weighted value computed over the range of all possible values. The maximum term-weighted value ranges from 0 to 1. The achieved results together with EERs are summarised in the table below.

Evaluation	Equal Error Rate	Maximum Term-Weighted Value
Baseline	10.13%	0.358
2-pass LVCSR	9.66%	0.478
3-pass LVCSR	8.04%	0.565

3.6.2 Exploiting out-of-language detection module in STD system

Many spontaneous conversations often contain short sentences uttered in different languages. The spoken term detection performance dramatically decreases when the system is employed on "inappropriate" speech input, such as speech pronounced in a different language or the out of vocabulary words. This introduces many difficulties for LVCSR, which is designed to recognize spontaneous speech pronounced in one language, including a higher number of false alarms. One solution consists in modifying the detection threshold (represented by the operating point given by the application) in order to reduce false alarms introduced by "inappropriate" input speech segments. However, this has a direct effect as an increase of missed spoken terms.

The out-of-language detection module [35] is an extension of out-of-vocabulary detection and is based on a confidence measure. It is able to detect speech segments that are not uttered in the same language for which the LVCSR system was designed. By exploiting an out-of-language detection module in a word lattice based spoken term detection system, "inappropriate" speech segments can be detected and thus the number of false alarms can be significantly reduced. The out-of-language detection module is based on processing word and phone lattices obtained by LVCSR. It can also be used as an out-of-vocabulary detector, i.e. to detect words that do not appear in the dictionary.

The study was carried on a database that contains discussions in English, Czech and German languages uttered by native Czech and German speakers. Combination of out-of-language detection module with the spoken term detection system to detect English spoken terms improved the performances by more than 2% absolute (7% relative).

3.6.3 Keyword spotting using neural networks

One of the promising approaches is based on neural network. Neural network based keyword spotting systems can be seen as a phoneme based system, where phoneme posteriors are estimated by the neural network. A keyword is then modelled by conventional hidden Markov model (HMM) in order to carry out a temporal context during detection. To evaluate the final likelihood of a detected keyword, a universal model of speech (background model) is used.

Although neural network based keyword spotting performs generally worse than the one using LVCSR output, such the system is beneficial from several points of view:

- It usually has much less missed keywords (although much more false alarms).
- It can detect any kind of keywords (it is not restricted by the dictionary).
- It can run much faster than other systems.

The study was carried out on 16 kHz real unconstrained speech recorded using close-talk microphones in a fairly clean environment (SNR 20 dB). In total, about 70 minutes of recordings pronounced in English by non-native (male/female) speakers were used. 740 occurrences of predefined keywords composed of various phone lengths (i.e., 3 to 8 phones) appear in the experimental data and their time positions are precisely annotated. A subset (70 minutes) of 16 kHz audio lectures annotated for ASR as well as keyword spotting tasks was employed for a training dataset.

As an acoustic keyword spotting, one of HMM neural network based phone ASR systems was employed in our experimental work [34]. More specifically, the phone recogniser exploits context-independent phone models, which are represented by phone posteriors estimated using neural networks and temporal pattern (TRAP) features. Outputs of neural networks, trained separately on left and right context TRAP parts, are merged using another neural network to produce 3-state phone posterior estimates for beginning, center and end of a phone. The word models of searched keywords are created from corresponding phone models. Parallelly concatenated keyword models are then accompanied by filler and background models (represented by simple phone loops) to create a decoding network, as shown in Figure 33.

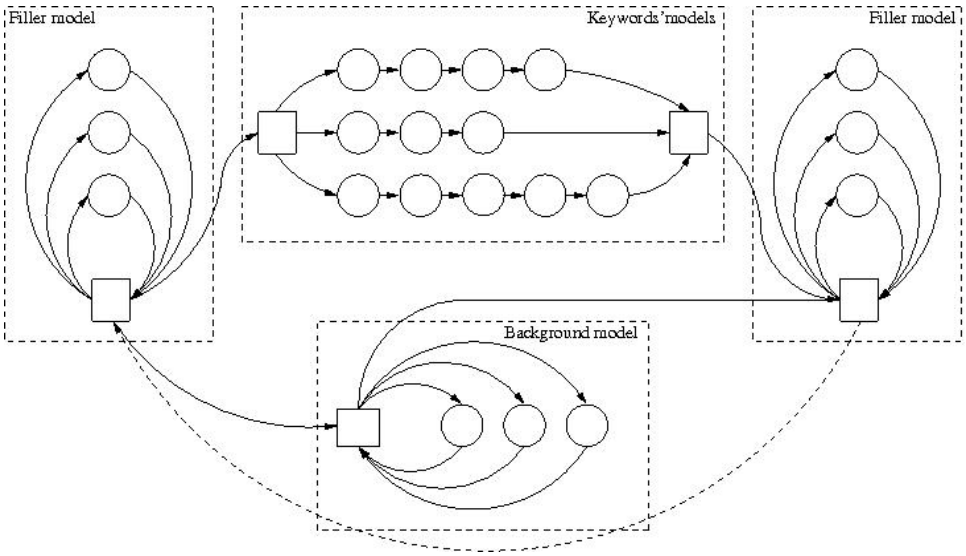

Figure 33: General scheme of acoustic keyword spotting

Likelihoods of the detected keywords are taken from the last state of each keyword model (computed using Viterbi decoder) and compared with the likelihood obtained from the background model. Confidence score of each detected keyword is then given as a log-likelihood ratio between these two likelihoods. The acoustic keyword spotting is denoted as **KWS-1xRT$_{acoust}$** and is able to run much faster than LVCSR keyword spotting.

Figure 34 compares DET curves of the keyword spotting detection on a test dataset using different individual systems. The plot indicates that in case of low false-alarm probabilities, **KWS-20xRT$_{lvcsr}$** significantly outperforms other keyword spotting systems. However, it yields insufficient performances for low miss probabilities. This is caused by the fact that some occurrences of keywords are not found even in weakly pruned word recognition lattices generated in the third pass. Other LVCSR keyword spotting systems perform worse and yield similar negative properties for low miss probability. Acoustic **KWS-1xRT$_{acoust}$** reaches significantly worse detection performances for low false alarm probability, but can operate up to low values of low miss probability.

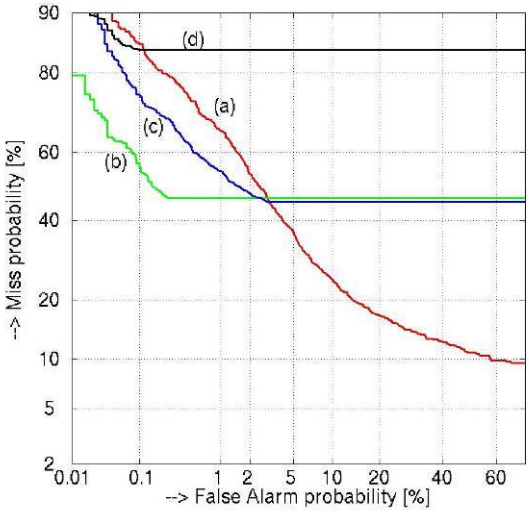

Figure 34: Keyword spotting performance:

(a) KWS-1xRT$_{acoust}$, (b) KWS-20xRT$_{lvcsr}$, (c) KWS-5xRT$_{lvcsr}$, (d) KWS-20xRT$_{lvcsr\text{-}phone\ lattice}$

3.6.4 Combining acoustic and LVCSR based keyword spotting systems

The acoustic keyword spotting can operate on much larger scale of DET curve (Figure 35) than LVCSR keyword spotting system. However, the detection performance is significantly worse, especially due to high number of false alarms.

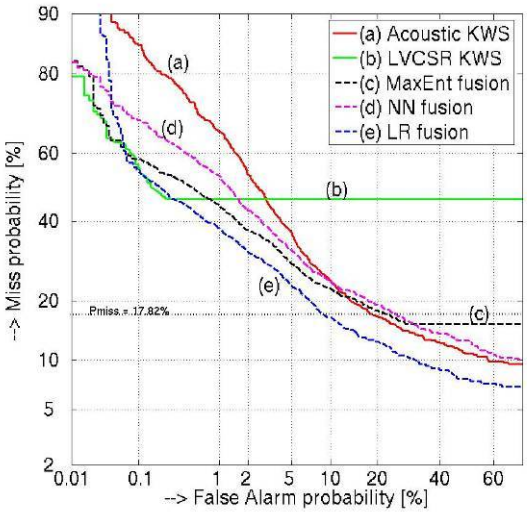

Figure 35: Performance of combined keyword spotting systems

Several conventional techniques were exploited to combine the acoustic and the LVCSR keyword spotting systems to improve keyword spotting performance:

- **NN -** A feed-forward back propagation neural network with one hidden layer. A hidden layer comprises 20 nodes with tangent sigmoid as a transfer function. Input is represented by confidence scores (log based) obtained by individual keyword spotting systems. Output of the neural network is trained to discriminate between 0/1 depending on the true/false occurrence of a given keyword in the transcription.
- **MaxEnt -** Maximum entropy criterion uses conditional maximum entropy models which have been shown to provide good performance in speech and language processing.
- **LR -** Linear regression. The resulting confidence score is given by linear combination of individual confidence scores weighted by a constant. In order to avoid problems with negative infinity values, the linear regression approach uses posterior probabilities for the combination.

Achieved DET performances are given in Figure 35. Equal error rates and relative number of false alarms (computed for the operating point given by EER of the acoustic keyword spotting where miss probability is 17.82%) are given in the table below. Combined confidence scores obtained using neural network and maximum entropy classifiers perform better for lower false alarm probability and worse for lower miss probability than acoustic keyword spotting. The linear regression classifier significantly improves detection performances overall operating points of the DET curve.

System	Equal Error Rate	False Alarms
Acoustic keyword spotting	17.82%	100%
Maximum entropy	18.54%	120%
Neural network	19.32%	134%
Linear regression	14.46%	46%

3.7 Distance to face estimation

The Kinect is a depth sensor originally developed by Primesense for Microsoft. Among others, it provides 11 bit single channel images in resolution of 640x480 pixels containing depth information of captured scene. In general, the depth information can be used as an input in most computer vision applications instead of camera image while providing direct information about object surfaces in a scene. Having object surface information makes tasks as object tracking and pose estimation much easier. In the TA2 project, the depth information is used to gain knowledge about participant positions, actions and poses. The depth information lets us to better distinguish between participants and to get their actual 3D positions or the location of detected events.

The presence of the Kinect sensor enables to enrich the tracked face positions (3.2) with the real 3D face/person position relative to the camera. However, reliability of this information depends on proper calibration – the relative position of the Kinect sensor and main camera must be known together with their projection (intrinsic) parameters.

The calibration is done using the well known Zhang's method [36] which is based on the capturing planar chessboard pattern. The pattern is simultaneously captured with the main camera and Kinect sensor. The chessboard is repeatedly located in both views and the intrinsic and extrinsic parameters are estimated and saved. The calibration can be performed during system installation; however, then the main camera and Kinect sensor positions and/or orientations cannot be changed without redoing the calibration process.

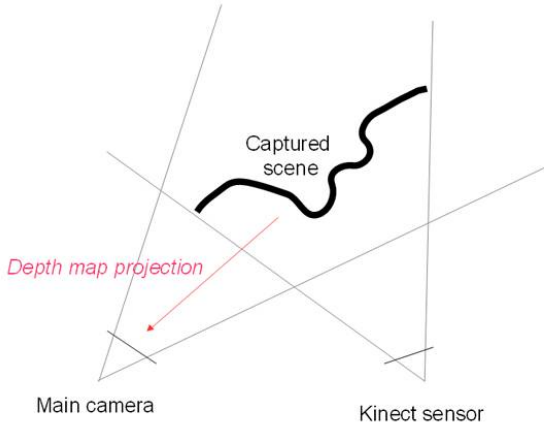

Figure 36: Schematics of the depth map projection
(the Kinect and main camera projection parameters and their mutual position must be known)

With the proper calibration the Kinect depth map, represented as a point cloud, can be projected according to the projection matrix of the main camera, creating depth map of the scene surfaces captured by Kinect from the main camera point of view (see Figure 36). This gives us correspondence of the camera pixels to the depth values. The depth map contains minor errors and even holes of unknown values due to depth discontinuities and occlusions. The distance from the face to the camera can be estimated by weighted average of its pixels. In the current implementation, the weighted average is computed on randomly selected subset of facial pixels to reduce the computational cost.

The projection of the depth map into the main camera view is the most computationally expensive part of the described approach. The depth map in native resolution (640x480) is represented by approximately 300’000 point-sprites and it takes around 50 ms to project them using single CPU core. The computational time can be reduced by sub-sampling the depth map at the expense of slightly lower accuracy. Naturally, the projection can be accelerated by using GPU. The GPU implementation achieves the task in less than 5 ms on mainstream GPUs.

3.8 Wiping, steering and pointing detection

The TA2 family game scenario brings the experience of a casual family game (such as a board game, Figure 25), an experience which normally relies very much on people's interaction within one room, to separated households. The essentials of a family game are interaction, communication, sharing a common experience and above all having fun together. Additionally, the current design of the TA2 family game includes so called minigames. These minigames are short collaborative physical activities during the course of the family game. Currently, there are two modules of analysis for minigames based on the Kinect sensor input supporting wiping and steering. The wiping module detects hand-waving gestures which can be used for example for interaction with objects or destruction of obstacles. The steering module is used to provide the user with a virtual steering wheel that can be used for steering various virtual vehicles. In addition to the minigame interfaces, Kinect is used to detect pointing and its direction which is passed as an additional cue to the orchestration module.

The steering and wiping modules share the same pre-processing scheme in which the user hands are located. The input information obtained from the Kinect sensor via OpenNI API is the depth map of the scene and segmentation of the users (user map) with assigned IDs. The user closest to the sensor is marked as the active user. The pre-processing works as follows: For the active user, the algorithm masks the depth map with the relevant user area which gives the user silhouette enriched with the depth information. The silhouette depth values are then recalculated with respect to the user center of mass, which suggests the possible hand locations which are usually extended from the body, as shown in Figure 37. To locate the hands, a non-maximal suppression algorithm, which detects extreme regions in the depth map, is used.

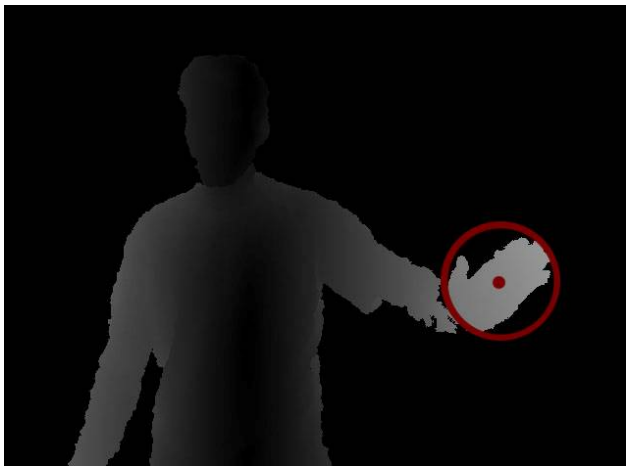

Figure 37: Hand position detection based on the depth to center distance map transformation

In the wiping module, the hand location and its local neighbourhood are tracked. This enables to detect common types of hand waving – hand movement with large spatial extent and fast repeated local palm movements or their combinations. The large movements are captured by differentiation of hand position and local movement is captured by frame difference. The resulting movement values are normalized and are passed via shared memory interface sent to xmlRPC server which makes it available to minigames.

The steering detection, by contrast, observes the mutual position of both user hands. If the user stretches his arms a bit in front of the body, the detection is triggered. As depicted in Figure 38, a virtual line segment is stretched between 3D locations of the hands. Consequently, size and orientation of the line segment within the 3D space is calculated. The resulting information contains values of four degrees of freedom that can be used arbitrary within the minigame. The values represent in-plane rotation, out-of-plane rotation, stretching and distance to the body.

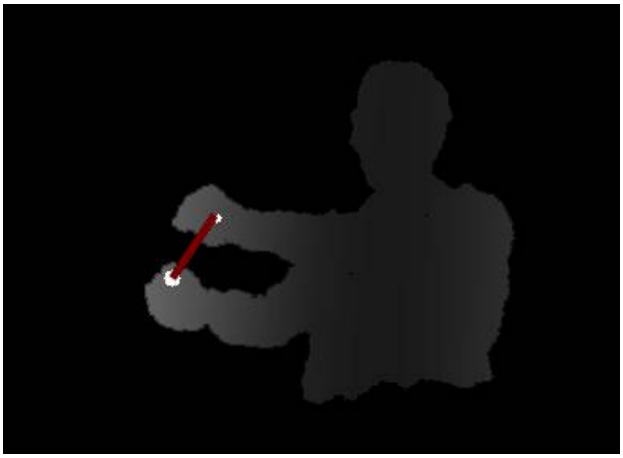

Figure 38: Steering detection
(a virtual line segment is stretched between 3D locations of the hands)

The pointing event and direction cannot be successfully estimated without more detailed information about user arm poses. So the processing scheme must involve so-called partial skeletonization step that fits a virtual skeleton modelling bones and joints to the user depth mask. This provides the estimated location of the main arm joints.

We have developed state-less skeletonization algorithm that is capable of locating shoulders, elbows and hand locations without the need of initialisation and tracking. The only assumption for skeletonization is that the user must be observed in more or less upright position. The algorithm processes the user depth silhouette. First, the user head is located, which serves as the main cue for estimation of the user body part dimensions. The shoulders are located as the second step by finding the best fit for a head-shoulders triangle. Using the shoulder and hand locations, it is possible to locate the elbow joint positions based on the assumption of the human arm size. The algorithm tries to fit the elbow joint by pruned state space search. The best position is selected based on the correspondences between depth silhouette and the virtual arm bones. The example of the skeletonization output is captured on Figure 39.

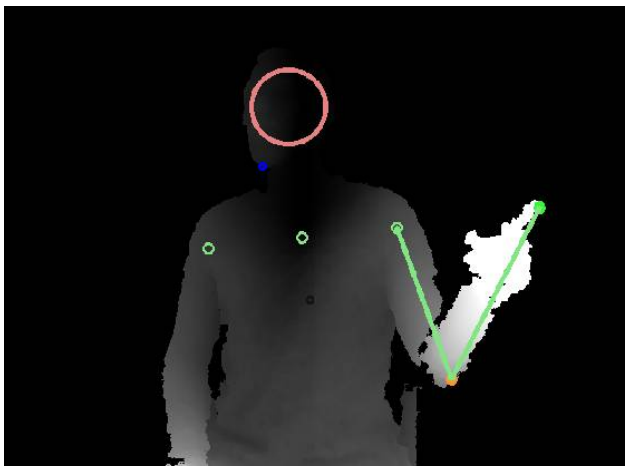

Figure 39: Example of the partial user skeletonization

The pointing direction can be then estimated directly from the arm pose, or in the case of stretched arm as the link of the hand and head. The main problem of the pointing detection is that it is hard to distinguish between an actual pointing event and a simple arm stretching. The best cue for this would be locating of the index finger; however, it is not possible to detect it in the current TA2 scenarios with current depth sensor because of limited sensor resolution.

3.9 Planar pattern detection and tracking

A known planar object can be detected in a camera image and its six degree of freedom can be reliably estimated [37], [38]. The methods rely on detection and description of invariant local image features to find correspondences between the searched pattern and image taken from the camera. From a set of estimated point correspondences, homography transformation of the object can be estimated by a robust estimator. RANSAC (RANdom SAmple Consensus) and its variants are used for this purpose. The six degree of freedom position can be estimated from the known homography transformation when camera parameters are known.

A large number of methods for detection of planar patterns exist. These methods differ in the type of local feature detector, descriptor, method for correspondence search and the particular RANSAC variant that is used. For the TA2 scenario several specific considerations have to be taken into account when implementing solution for this task. In the expected home setup, the participants sit relatively far from the main camera and a reasonably sized card in their hands covers just small fraction of the full HD image captured. To be able to detect such small patterns, a large number of small local image features have to be detected. Implementations of highly invariant detectors such as SIFT and SURF [37] cannot achieve real-time performance in these conditions.

The method [38] uses combination of only moderately invariant but very fast local features detection and description, and semi-naive Bayesian model learned to reliably match the features. Although the method can use any feature detector, the original work uses Harris corner detector, which only detects position of the features and does not give any information about scale or rotation. Small image neighbourhood of the detected features is described by comparison of pixel values at random locations pairs. Each comparison gives one bit of information and these are concatenated into a code called fern. Reasonable length of ferns is 9-11 bits. An image feature is described by several fern codes. A semi-naive Bayesian classifier, whose features are the ferns, is used to compute probability that a detected local feature corresponds to a feature of the pattern. The classifier is trained during initialisation step by applying random geometric transformation and noise to the pattern, and collecting statistics of the fern descriptors for detected features. The approach is illustrated in Figure 40.

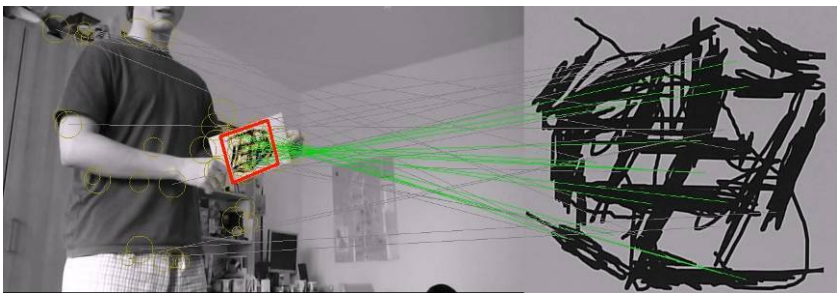

Figure 40: Visualization of planar pattern detection approach: yellow circles are local features that probably correspond to some part of the searched pattern; red is the estimated object position; green lines are correspondences consistent with the estimated position

In addition, we implemented our own version, which scales better with number of detected patterns and image resolution. The better scalability is achieved by solving correspondences for all patterns in a single semi-naive Bayesian classifier and by using a version of RANSAC, which selects the random samples for estimation of homography such that their distance is lower than some threshold. Setting this threshold to the expected size of the pattern in the camera image significantly improves the chance that consistent correspondences are selected, which improves RANSAC convergence especially for high-resolution images. Accuracy of both implementations is sufficient for the intended applications in the TA2 system. In the TA2 system, they are used as an alternative solution for the steering minigame user interface (see section 3.8), and for alternative login into the system where a distinct pattern is used to associate user ID to person in the main camera during the session initialisation.

3.10 Action detection

The possibility of using space-time interest points and statistical classifiers to detect a fixed set of user actions from the main camera view has been evaluated. The action detection approach is based on extraction of space-time interest points [39] from video. These interest points are extracted from the position of participants based on head position tracking and they describe visual changes in the video. The space time-interest points sample the 3D space-time (image coordinates and time) space at positions with high information content. These positions are described by a fixed length descriptor vectors. The descriptors from a short video segment are converted to a bag of words representation using codebook transform. The bag of words vectors are then classified by support vector machine classifiers to estimate if any of the action of interest is present.

The two different approaches to space-time interest point extraction were evaluated. The method of Laptev and Lindeberg [39] is built upon the well known Harris interest point detector, which detects corner-like structures in 2D image. The space-time interest point detector extends the Harris detector to 3D finding well-localisable local events. The second approach is dense sampling, which samples the video volume homogenously on a regular grid aiming to preserve the most information from the video. The dense sampling has higher computational cost as it usually provides higher number of regions.

The local descriptor used in the experiments was inspired by Dollar et al. [40]. This descriptor describes an axis-aligned cuboid centred at the interest point position. To obtain illumination invariant description, the pixel values are locally normalised, brightness gradient and optical flow is computed. The brightness gradient is calculated at each spatio-temporal location. Optical flow is computed by the Lucas-Kanade algorithm [41]. The resulting description vector is created by serialising histograms of the gradients and the optical flow. Several histograms are computed on a 3D grid. By using the histograms this way, the descriptors become insensitive to small geometrical transformations and imprecision in interest point localisation while retaining enough structural information.

To create feature vectors suitable for classification, codebook transform was used to translate the sets of local descriptors to a bag of words representations. Generally, codebook transform assigns objects to a set of prototypes and computes occurrence frequency histograms of the prototypes. The prototypes are commonly called codewords and a set of prototypes is called a codebook. In our case, the codebooks were created by k-means algorithm with Euclidean distance.

When assigning local features to codewords by hard mapping, quantization errors occur and some information is lost. This is especially significant in high-dimensional spaces, as is the case of the local volume descriptors, where the distances to several nearest codewords tend to be very similar. In the context of image classification, this issue was discussed for example by Gambert et al. [42], who propose to distribute local patches to close codewords according to codeword uncertainty. We used the uncertainty soft keyword assignment in our experiments.

In the experiments, the support vector machine classifiers were used and their optimal hyper-parameters were estimated by grid search with 5-fold cross-validation. We evaluated the action detection system on action classes of waiving and clapping. For the evaluation purposes a special dataset was created, which contained sitting people facing the camera and performing actions of interest and other arbitrary movements, which were used as distracters in learning. The optimal time-window length and other parameters were estimated during the experiments. The average precision for the best parameter setting was 65% for clapping and 72% for waiving.

The results seem reasonable and the approach is performing relatively well. However, the considered actions are very rare in the TA2 scenarios. They happen only several times per communication session and the detection system still gives several false alarms per minute at a setting, where it is able to detect 75% of actions of interest. The conclusion is that the detection accuracy useful for orchestration cannot achieved by using the space-time interest points, a bag of words and support vector machine classifiers.

3.11 Multimodal calibration, association and fusion

The fusion can be performed at different levels, based on type of input information available. It can be at sensor level, feature level, score level, rank level or decision level. First two levels can be considered as pre-classification category, while others can be considered as post-classification category [43]. The feature-level multimodal approach is normally done via transformation of the data in such a way that a correlation between the audio and a specific location in the video is found [44], [45]. In our work we concentrate mainly on score level fusion and propose a technique [46], which relies on information derived from spatially separated sensors. By placing the sensors at their individually optimal locations, we clearly obtain a better performance of low-level semantic information. This in turn results in better performance of the complete system. Other score-level multimodal techniques rely on estimation of the mutual information between the average acoustic energy and the pixel value [47], probability densities estimation [48] or a trained joint probability density function [49].

The association and fusion of acoustic and visual events is not a trivial task, because at each time instant there might be some events that are more reliable than others. The combined model has to be able to compute a confidence measure of the different modalities and weighs them accordingly. In addition, the sensors capturing the audio and video signals are spatially separated. Due to the real-time requirements, the association and fusion of person IDs from the video identification with voice activity cannot be postponed until the voice activity is over. The fused events have to be available within a timeframe of two hundred milliseconds to keep the feeling of being instantaneous. The low delay temporal association and fusion scheme is depicted in Figure 41.

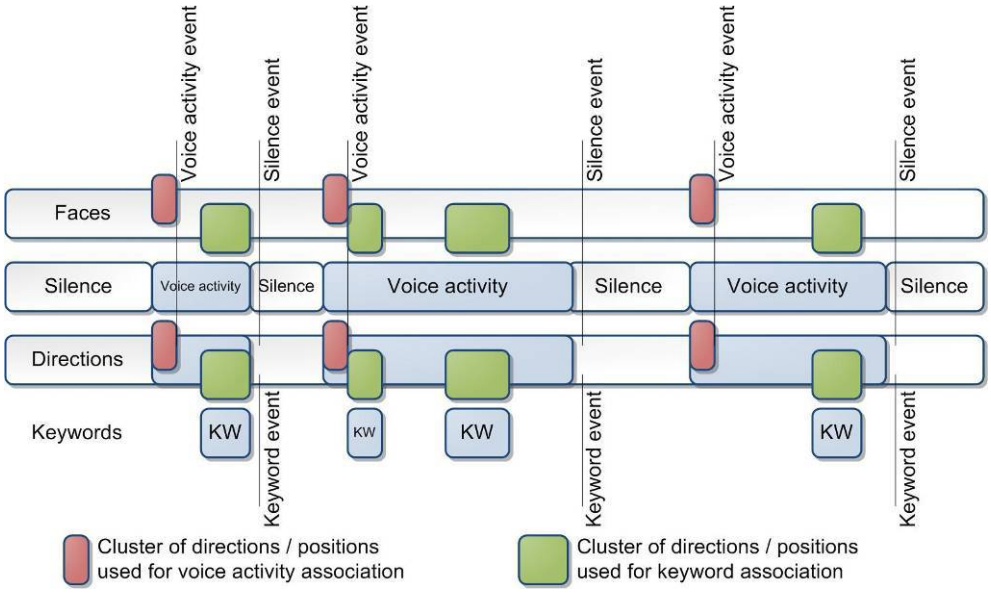

Figure 41: Low delay association and fusion

Since the position of people does not significantly change within a few hundred milliseconds, predictive temporal association is employed for video modality to remove possible lags during the capturing of the video stream by hardware and video grabber. Further, audiovisual association is performed between acoustic short-term directional clusters and the positions of detected faces from the video modality. This involves a mapping estimation between microphone array coordinates (acoustic directional clusters w.r.t. the microphone array centre) and the coordinates of the image plane, which are defined by the field of view of the camera (Figure 42).

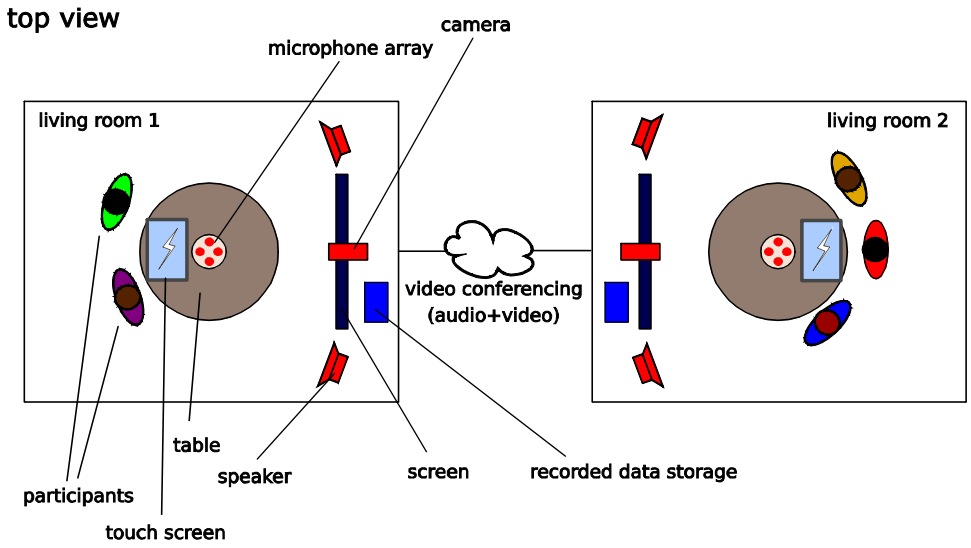

Figure 42: TA2 setup, view from top [5]

Since the participants do not sit at predefined positions in the room, it can often cause ambiguities in the association and fusion. Clearly, the same acoustic short-term directional cluster can correspond to different positions in the image and vice-versa. Therefore, the location of a detected face within the image can be mapped to many different sound directions. However, since the participants are mainly located around a coffee table, such ambiguities occur rarely. Therefore an association can be computed [46] via distance minimisation between the positions of the detected people, the position of the microphone array and projections of the mean angles of the acoustical directional clusters.

Automatic calibration of the association and fusion parameters is shown on Figure 43 for Datasets 1 and 2 (see the section 3.12 for datasets description). More specifically, the horizontal position of the microphone array and projection weight for the acoustical directional clusters were considered as free parameters (C2 and C1 respectively) and estimated to provide the best accuracy in the mapping between various face locations and given sound direction. The automatic calibration of the parameters was performed on the first few minutes (~5%) of the corresponding datasets. The highest achieved accuracies are plotted with bold lines. As it can be seen in Figure 43, the optimal estimated parameters differ over different datasets. On the other side, reasonably a priori selected parameters, i.e., without employing the process of calibration result in good performances as well. This is especially true for Dataset 1 (see Figure 43 a-b), where variations of calibration parameters do not have strong effect on the final accuracy of audio-visual association. This is however not completely true for Dataset 2 (see Figure 43 c-d), where the effect of optimally selected parameters on the final accuracy is more important. The best performances are plotted with bold lines.

In addition to the fusion of acoustic and visual events, the correspondence between data from the main camera and Kinect sensor has to be established. This is possible by their mutual calibration. That is, their relative position and orientation is estimated together with their viewing parameters. With this information, it is possible to align the detected cues, or add spatial information to the events detected in the main camera. Currently, an initial offline calibration procedure is used (see the section 3.7). However, the fusion of the main camera and Kinect sensor has inherent limitations. Due to the different fields of view of the devices, the captured areas of both devices differ. Due to this, the depth information may not be available for some parts of the main camera view. The both devices have to be positioned properly to maximize the overlapping region.

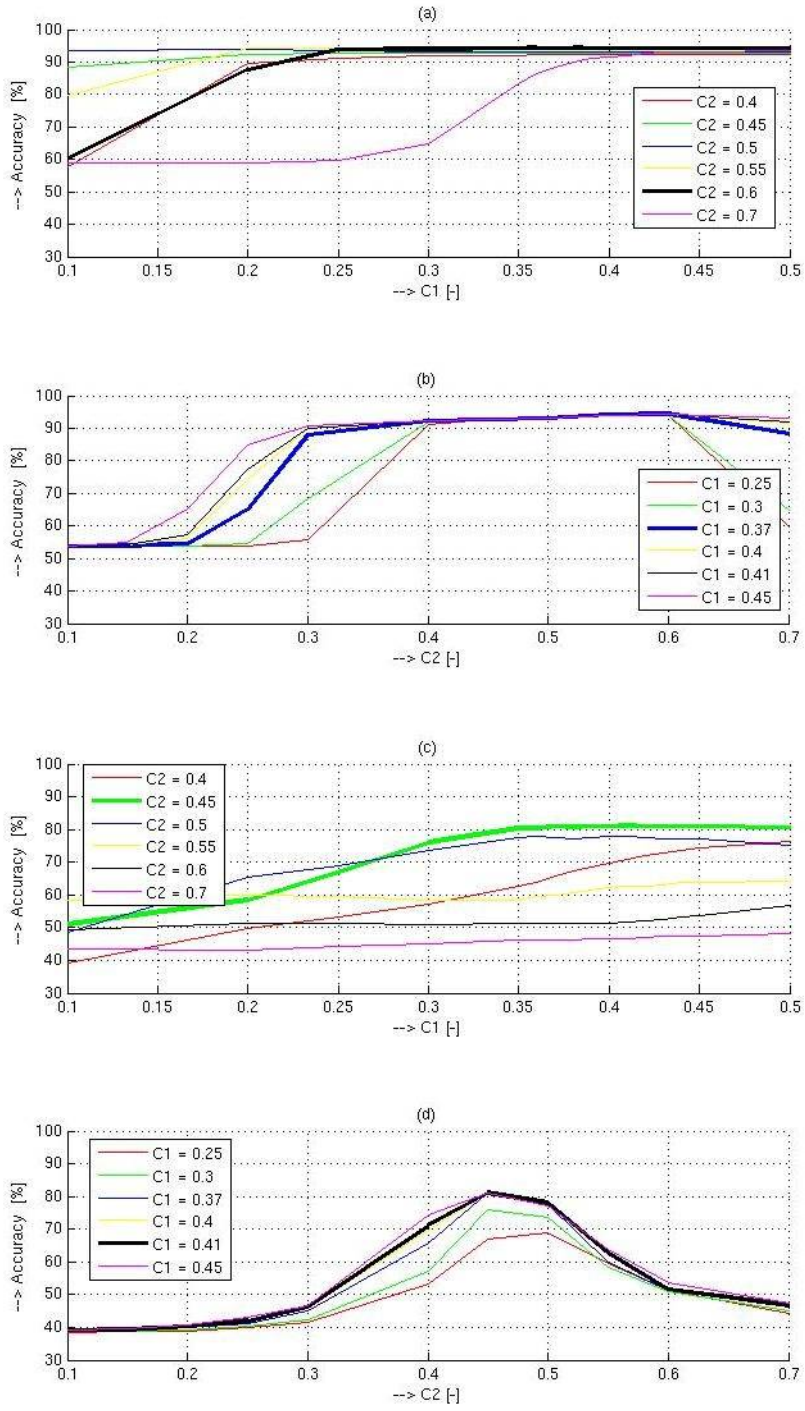

Figure 43: Accuracy of the audio-visual association for Datasets 1 and 2:
(a) accuracy = f(C1), for various C2, Dataset 1;
(b) accuracy = f(C2), for various C1, Dataset 1;
(c) accuracy = f(C1), for various C2, Dataset 2;
(d) accuracy = f(C2), for various C1, Dataset 2

3.12 Evaluations

The experiments for objective evaluations were performed on real life hand-labelled datasets (3 h 50 min for Dataset 1 with enabled echo suppression [50]; 1 h 20 min for Dataset 2 [5] with disabled echo suppression, lower SNR and fewer frontal face views). The Dataset 2 [5] was made publically available. The datasets follow the systematic description presented in [5] and contain recorded gaming sessions with enabled video chat of socially connected but spatially separated people. Each room was recorded and analysed separately and contained up to 4 people. The annotated voice activity events from Dataset 2 are illustrated in Figure 44. One can see that there are a lot of short utterances, and speakers change quite frequently.

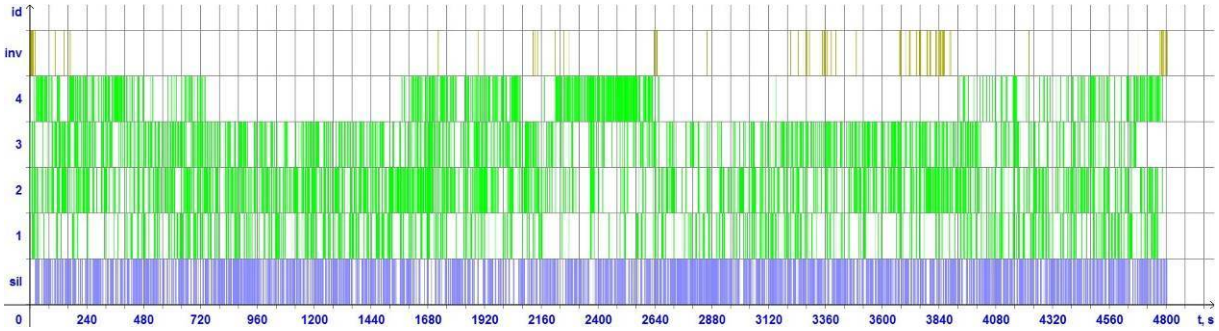

Figure 44: Annotated voice activities over time for room 1 of Dataset 2 [5] (the first row (inv) shows voice activity of persons not visible from the camera, the four following rows show the voice activity of the four different persons visible in the video (id 1 to 4), and the last row shows silence (sil))

The achieved results at different steps of processing are summarised in Figure 45. The results from the blocks, marked with a tick, are propagated further to the TA2 system, while the results from other blocks are intermediate or comparative. Precision is defined as the number / the total time of true positive test events (test events correctly detected as belonging to the positive class) divided by the total number / the total time of test events detected as belonging to the positive class (the sum of true positive and false positive test segments). Recall is defined as the number / the total time of true positives test events divided by the total number / the total time of test events that actually belongs to the positive class (the sum of true positive and false negative test events).

The block “Face Detection” shows precision and recall of a standard face detector [4] applied on single frames of the video stream. It represents mean value over all people. The block “Multiple Face Tracking” shows the results of the face tracking algorithm, which improves the overall accuracy of the video processing. The recall (i.e. detection rate) is increased by absolute 2.3% while the false positive rate (i.e. the proportion of falsely detected faces) is decreased by absolute 35.7% compared to a face detection algorithm [4]. More extensive multiple face tracking evaluation is presented in [7], where we have shown that the recall is increased by relative 7.8% while the false positive rate is decreased by relative 38.3% compared to a state-of-the-art multiple target tracking algorithm.

The block “Acoustic Voice Activity” and derivative blocks show precision and recall on the output of the local far-field voice activity detection (6000+ observations). Although only Dataset 1 is echo-cancelled we were able to achieve reasonably good precision/recall levels for Dataset 2 after application of the directional filtering on semantic level within voice activity detector (the difference of 20.2% precision can be seen between corresponding blocks). The sector of interest in the final system for directional filtering was defined as [-110°, 110°] with respect to the reference direction of 0°, defined as an imaginary arrow intersecting the camera and the centre of the microphone array, facing the participants. This allows us to eliminate remote parties in case of disabled echo suppression (Dataset 2) and few echo-cancelation artefacts in case of enabled echo suppression (Dataset 1).

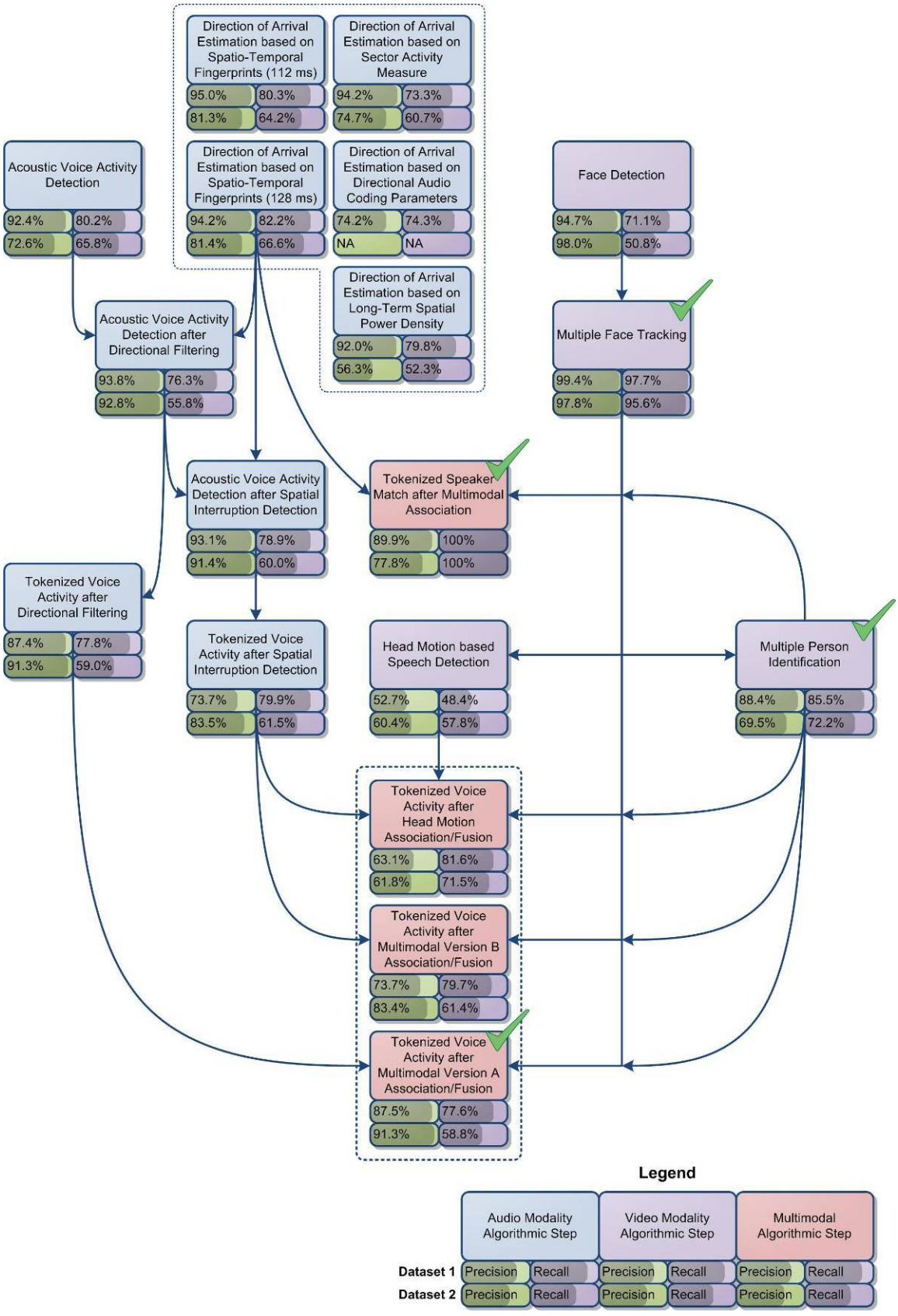

Figure 45: Evaluations at different steps of processing

The blocks "Acoustic/Tokenized Voice Activity Detection after Directional Filtering" show precision and recall values for the case barge-in events are treated by temporal interruption detector, while the blocks "Acoustic/Tokenized Voice Activity Detection after Spatial Interruption Detection" show precision and recall values for the case barge-in events are treated by spatial interruption detection. While the approach with spatial interruption detection shows slightly better performance (using temporal weighted scoring), surprisingly we have found that in case of tokenized events (using event based scoring) the spatial approach has a "dropped" performance. We presume this is due to some of false alarms have been fragmented into shorter ones.

The performance of far-field voice activity detection based on fusion of multimodal information is given in the blocks "Tokenized Voice Activity after Multimodal Version A/B Association/Fusion" and "Tokenized Voice Activity after Head Motion Association/Fusion". One can see that the performance varies due to assigning the voice activity to video tracked person and fusion with head motion estimation. The block "Tokenized Speaker Match after Multimodal Association" expresses a person identity association of detected voice activity, i.e. how well we can assign previously detected voice activity to a local person based on AV information. A more complex model based on full-body movements or hand gestures could be considered in the future. However, this could possibly increase the delay for voice activity detection, and induce further challenges, e.g. in the given scenario, people also move their hands while manipulating the touch screen.

Since voice activity detection (VAD) is a detection task, performance can be characterised by detection error trade-off (DET) curves of miss versus false-alarm probabilities. These probabilities are estimated using absolute number of targets (i.e., number of speech segments comprised in the transcription) as well as non-targets (i.e., number of potential speech segments not comprised in the transcription but appearing in the detection output). The resulting DET curves are normalised in such a way that the number of targets and non-targets is set to be equal. For each operating point in DET curve, precision and recall values can be estimated. Thus depending on a potential application, VAD can easily be tuned by considering different thresholds applied on confidence scores associated with each speech segment.

Figure 46 and Figure 47 show DET characteristics for detection of voice activity on Datasets 1 and 2. More specifically, 5 different audio-visual VAD systems were considered based on the input audio and a visual motion extracted from video stream:

- Audio: the speech is detected purely from the audio signal in the block of ACDE, together with confidence scores, estimated for each detected speech segment.
- Video: the speech is detected purely from the video, using a visual motion algorithm (from the section 3.4).
- Audio + Video #1: the speech segments are detected from both modalities and are merged in case of their overlap, the confidence scores from audio and video are linearly weighted.
- Audio + Video #2: the speech segments are detected from audio only, however the corresponding confidence scores are estimated using visual motion algorithm.
- Audio + Video #3: the speech segments are detected from audio only, however the assigned confidences are given by combination of acoustic and visual confidence scores.

Graphical outputs presented by DET plots in Figure 46 and Figure 47 show that the VAD based on both audio and video modalities (Audio + Video #1) outperforms audio-only VAD. In more detailed view, the highest improvement was obtained for Audio + Video #1 VAD system, where the speech segments are first detected independently from the both modalities and then merged into one stream of speech segments. In case of simple scenario provided by Dataset 1, where the audio signals from the remote rooms were well separated using echo cancellation and the audio have relatively high SNR, Audio + Video combination did not improve over audio only VAD. In case of Dataset 2, where the audio is not echo cancelled, the Audio + Video combination offers better detection results over the

whole DET curve (especially for low miss probabilities). In case of moderately difficult Dataset 2, audio-based VAD outperforms video-based VAD, however Audio + Video combination is able to enlarge a potential set of operating points (especially when low number of false-alarms is expected).

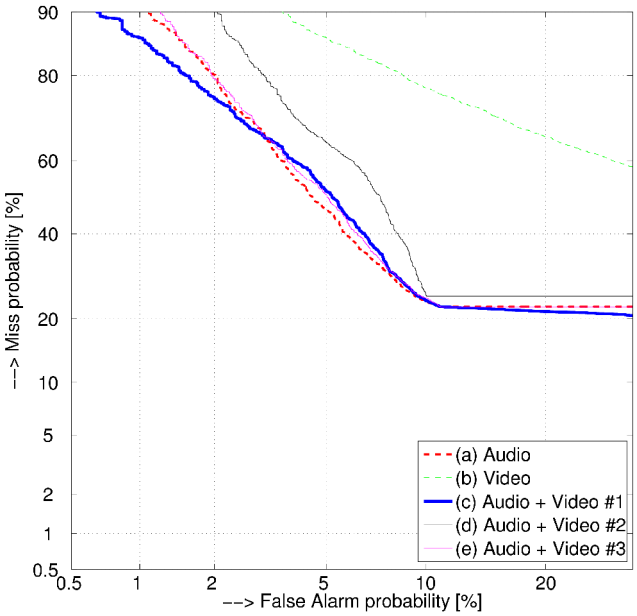

Figure 46: DET plot of voice activity detection performance for Dataset 1

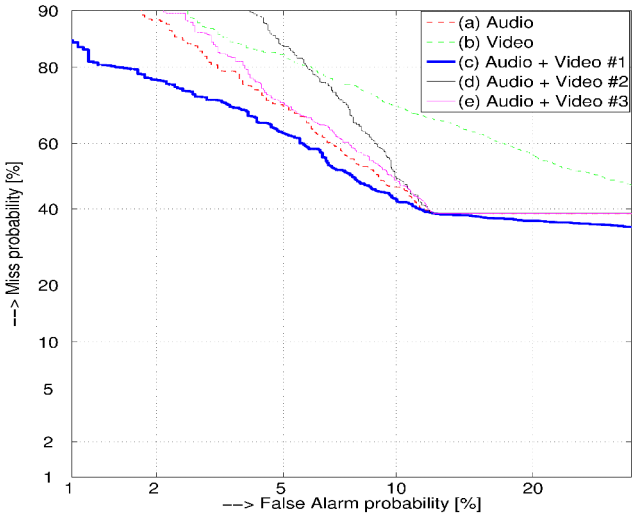

Figure 47: DET plot of voice activity detection performance for Dataset 2

Now let's take a closer look to acoustic localisation. All presented in Figure 45 approaches have latency within 128 ms. In case of the approach based on spatio-temporal fingerprints the dependency between precision and recall values can be estimated via application of different thresholds at the step of packing data into the spatio-temporal fingerprint representation. In Figure 48 this dependency is illustrated for speaker match. This dependency is calculated based on continuous data representation rather than discrete event count.

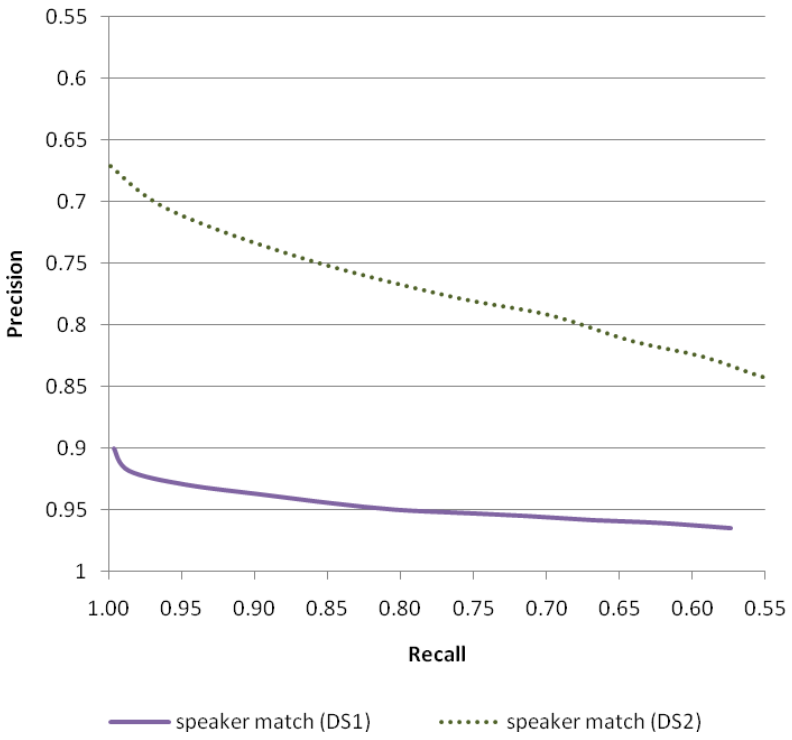

Figure 48: Precision versus recall for speaker match (Dataset 1 (DS1) and Dataset 2 (DS2)); Dataset 1 shows much better performance due to there are only two people within a sector of 130°, while in case of Dataset 2 there are 4 people within a sector of 90°, which is going beyond the spatial resolution of the used microphone array

It is clearly visible, that for Dataset 1 the speaker match (solid line) shows much better performance than for Dataset 2 (dot line). This is due to higher people density in Dataset 2. Depending on the subsequent application, the precision and recall priorities can be different.

Another parameter, which has a strong impact on the performance of the system, is the length of spatio-temporal fingerprints. This parameter defines as well the algorithmic delay of the proposed approach. The best results were achieved for spatio-temporal fingerprints within [112 ms, 192 ms]. We were able to achieve 95.0% precision and 80.3% recall for the speaker match with algorithmic latency of 112 ms for Dataset 1 (81.3% precision and 64.2% recall for Dataset 2).

Known meeting-wise speaker error rates for CPU-intensive state-of-the-art techniques [51] are as low as 7.0% for realigned MFCC+TDOA combination of the HMM/GMM system with optimal weights and for Kullback-Leibler based realigned MFCC+TDOA combination of the information bottleneck system with optimal weights. In the case of automatic weights, overall speaker error rates are 13.6% and 9.9% correspondingly. These state-of-the-art estimates are given only as an overview and cannot be used for direct comparison as the data, hardware and scenario used in our experiments differ from the data, hardware and scenario used in [51]. In addition the state-of-the-art systems have a latency of 500 ms and a state of minimum 3 seconds duration, while we were able to achieve reasonably good results with an algorithmic delay and minimum state duration as low as 112-128 ms, which is more crucial for TA2 scenarios. We should note that the algorithmic delay does not include capturing delay, which in turn can result in additional 10-20 ms. Surely there is a trade-off between lower latency and better precision/recall values, and systems, not requiring lowest possible delay, can potentially achieve higher precision/recall values.

According to the state-of-the-art [52], [53], [54], the expected performance of the large vocabulary ASR system is less robust than for "simple" ASR systems dealing with isolated words from a small vocabulary in laboratory conditions (the performance (word accuracy) of such an "ideal" system is almost 100%): the implications of "a real life environment" means a drop in accuracy of 30% or even more (depending on the ratio of speech energy level to background noise energy level). The implications of low latency and real-time execution mean partial exclusion of syntactic-semantic models and a drop in accuracy of ~12%.

The wide evaluations [55] of state-of-the-art algorithms are performed by NIST (National Institute of Standards and Technology) each year from 1986 in different setups (Figure 49).

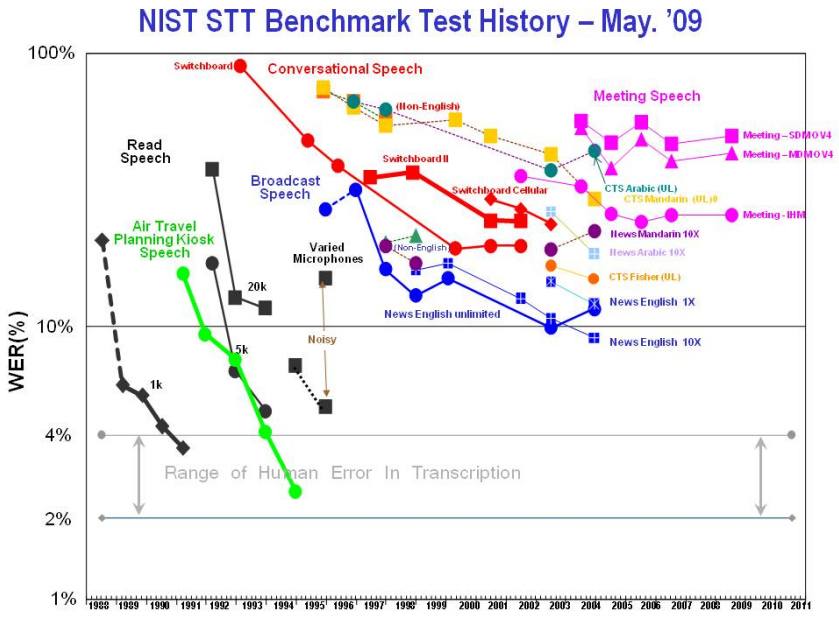

Figure 49: The history of automatic speech recognition evaluations at NIST [55]

For comparison, we also give the (multi-pass) AMI result on the more recent RT09 evaluation [56], which can be taken as a lower boundary on the achievable error rate for individual headset microphone (IHM) and multiple distant microphone (MDM). Corresponding performance for used approach is summarised in the table below. The numbers represent word accuracies.

Evaluation	**Performance (word accuracy)**	
	Individual Headset Microphone	Multiple Distant Microphone
RT07	63%	59%
RT09	76.5%	66.8%

Accuracy for commercial systems is paramount, and commercial solutions typically achieves this by having the user "train" the software during setup and by adapting more closely to the user's speech patterns over time. Automated real-time speech services that interact with multiple speakers in the same setup do not allow for speaker training because they must be usable instantly by any user. To cope with the lower accuracy, they either handle only a small vocabulary or strongly restrict the words or patterns that users can say.

In TA2 scenarios it is not possible to adapt a speech-recognition system to individual speakers. The ASR system has to cope with a relatively high level of reverberation, acoustic background "noise" and speaker-independency. Co-articulation, inter-speaker variability, language variability, noise and reverberation are known to be the main challenges for these systems. These factors and usage of multiple distant microphone gives performance levels sufficient for secondary subsidiary cues, nevertheless it cannot yet be exploited in the strict rules, which require 100% reliability. We experimented in parallel with other approaches to more accurately detect predefined keywords such as spoken term detection and neural networks. Another technique, which can be employed prior to the speech recognition system to improve the detection of keywords in case of multilingual conversations, is out-of-language detection. Although these algorithms are currently implemented to run in offline mode, their use for real-time detection is feasible. Corresponding performance of these methods is presented in sections 3.6.1, 3.6.2, 3.6.3 and 3.6.4.

Due to the Kinect sensor was available on the market after the Datasets 1 and 2 were recorded and annotated, the same datasets cannot be used for objective evaluation of Kinect-based algorithms, nevertheless during subjective evaluation the performance of estimation of depth information, detection of some hand gestures (pointing and wiping), as well as the detection, recognition and tracking of specified patterns (on a sheet of paper) was sufficient for the intended TA2 applications.

During subjective evaluations of earlier version [46] of analysis engine, several bottlenecks for further evolution of the system were experienced. To overcome these bottlenecks, several architectural and algorithmic changes had to be applied.

First of all, while the socket interface was allowing for a flexible software solution, the experienced latency for uncompressed video signal transmission from remote video grabber was resulting in additional latency of 30-300 ms. This tangible lag was successfully removed by switching to a shared memory interface for video input stream. While a shared memory interface could be potentially used for audio input stream as well, experienced latency of 12-20 ms for audio transmission was on acceptable level and kept as before.

To reduce the latency of audio processing we have decided to reduce the algorithmic delays of both direction of arrival estimation and voice activity detection. The algorithmic latency of both components were reduced from 200 ms down to 128 ms. This is due to the replacement of the previous implementation based on a short-term clustering approach [57] by the computationally more efficient spatio-temporal fingerprints processing [16] and the reduction of corresponding temporal filters.

Exact clock synchronisation between separated audio and video grabbers was seen as another source of potential problems and during subjective evaluations we have found that use of local timestamps results in more consistent multimodal association and fusion. Moreover since the position of people does not significantly change within a few hundred milliseconds, predictive temporal association was finally employed within the system to remove possible lags during the capturing of the video stream by hardware and video grabber.

Since the participants do not sit at predefined positions in the room, theoretically it can cause ambiguities in the association and fusion. Clearly, the same acoustic directional cluster can correspond to different positions in the image and vice-versa. However, since the participants in TA2 scenarios are mainly located around a coffee table, such ambiguities occur rarely during evaluations.

Finally, head pose and visual focus of attention estimation have been identified as important semantic cues for orchestration engine and have been successfully integrated into the analysis engine. Head pose estimation is to be used for better selection of frontal/side views with respect to aesthetic and cinematic rules, while visual focus of attention can be beneficial for better modelling of social interactions (e.g. predictive turn estimation during grant-floor moments) and can have a direct impact on temporal filters within the aesthetic and cinematic rules.

3.13 Conclusion

In this chapter we have described and evaluated different components of a low delay real-time multimodal cue detection engine for open, unconstrained environments with spatially separated multimodal sensors.

The main contributions in this regard are the following:

- A state-of-the-art online multiple face tracker in terms of precision and recall over time.
- A probabilistic framework for track creation and removal that takes into account long-term observations to cope with false positive and false negative detections.
- A robust and efficient person re-identification method.
- A real-time framework for head pose and visual focus of attention estimation.
- A robust and efficient localisation method.
- A low delay real-time framework for detection, association and fusion of spatial voice activity and keywords with detected and identified people.
- Comparison of several keyword spotting approaches.
- A distance to face estimator to derive 3D face position from 2D face position.
- A real-time wiping, steering and pointing detectors.
- A real-time planar pattern detector and tracker.
- Evaluations of the involved techniques.
- The dataset [5] and the shared memory interface for distributing visual data were made publically available.

We have described applied architectural and algorithmic findings to improve the performance of different approaches on challenging data, reduce an overall latency down to 130 ms and fulfil real-time processing requirements.

The presented performance levels (achieved to date on hand-labelled datasets) illustrate the performance of the techniques developed within the TA2 project.

3.14 References

[1] M. Falelakis et al., "Reasoning for video-mediated group communication", In Proc. IEEE International Conference on Multimedia & Expo (ICME), Barcelona, Spain, 2011.

[2] D. Bohus and E. Horvitz, "Dialog in the open world: platform and applications", in Proc. of ICMI, Cambridge, USA, 2009.

[3] K. Otsuka, et al., "A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization", in Proc. of ICMI, Chania, Greece, 2008.

[4] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features", in Proc. of CVPR, Hawaii, USA, 2001.

[5] S. Duffner, P. Motlicek, and D. Korchagin, "The TA2 database: a multi-modal database from home entertainment", in Proc. of Signal Acquisition and Processing, Singapore, 2011.

[6] Z. Khan" MCMC-based particle filtering for tracking a variable number of interacting targets", in IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, pp. 1805–1918, 2005.

[7] S. Duffner, J.-M. Odobez, "Exploiting long-term observations for track creation and deletion in online multi-face tracking", IEEE Conference on Automatic Face & Gesture Recognition, 2011.

[8] C. Scheffler and J.-M. Odobez, "Joint adaptive colour modelling and skin, hair and clothing segmentation using coherent probabilistic index maps", in Proc. of the British Machine Vision Conference, 2011.

[9] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection", in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[10] S. O. Ba, and J.-M. Odobez, "A probabilistic framework for joint head tracking and pose estimation", in Proc. of the International Conference on Pattern Recognition, 2004.

[11] S. O. Ba, and J.-M. Odobez, "Recognizing visual focus of attention from head pose in natural meetings", in IEEE Transactions on System, Man and Cybernetics, 39 (1):16–33, 2009.

[12] D. Sodoyer, B. Rivet, L. Girin, J. L. Schwartz, and C. Jutten, "An analysis of visual speech information applied to voice activity detection", in Proceedings of ICASSP, 2006.

[13] G. Friedland, H. Hung, and C. Yeo, "Multi-modal speaker diarization of real-world meetings using compressed-domain video features", in Proceedings of ICASSP, 2009.

[14] A. J. Aubrey, Y. A. Hicks, and J. A. Chambers, "Visual voice activity detection with optical flow", IET Image Processing, 4(6):463, 2010.

[15] H. Hung, and S. O. Ba, "Speech/non-speech detection in meetings from automatically extracted low resolution visual features", in Proceedings of ICASSP, Dallas, USA, 2010.

[16] D. Korchagin, "Audio spatio-temporal fingerprints for cloudless real-time hands-free diarization on mobile devices", in Proceedings of the 3rd Joint Workshop on Hands-Free Speech Communication and Microphone Arrays (HSCMA), pp. 25–30, Edinburgh, UK, 2011.

[17] P. N. Garner, J. Dines, "Tracter: a lightweight dataflow framework", in Proc. of Interspeech, Makuhari, Japan, 2010.

[18] G. Lathoud and I. A. McCowan, "A sector-based approach for localization of multiple speakers with microphone arrays", in Proc. of SAPA, Jeju, Korea, 2004.

[19] C. H. Knapp and G. C. Carter , "The generalized correlation method for estimation of time delay", in IEEE Trans. on Acoustics, Speech and Signal Processing, vol. 24(4), pp. 320–327, 1976.

[20] J. DiBiase, H. Silverman, and M. Brandstein, "Robust localization in reverberant rooms", in Microphone Arrays, M. Brandstein and D. Ward, Eds. Springe, ch. 8, 2001.

[21] B. Champagne et al., "Performance of time-delay estimation in the presence of room reverberation", IEEE Trans. on Speech Audio Processing, vol. 4(2), pp. 148–152, 1996.

[22] C. Zhang, D. Florelncio, and Z. Zhang, "Why does PHAT work well in low noise, reverberative environments?", Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2565–2568, 2008.

[23] M. Flynn, "Some computer organizations and their effectiveness", in IEEE Trans. Comput., C-21: 948, 1972.

[24] P. N. Garner, "Silence models in weighted finite-state transducers", in Proceedings of Interspeech, Brisbane, Australia, 2008.

[25] P. N. Garner et al., "Real-time ASR from meetings", Proceedings of Interspeech, pp. 2119–2122, Brighton, UK, 2009.

[26] D. Moore, J. Dines, M. Magimai-Doss, J. Vepa, O. Cheng and T. Hain, "Juicer: a weighted finite-state transducer speech decoder", 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI), 2006.

[27] Digital Cellular Telecommunications System (Phase 2+), "Voice activity detector (VAD) for adaptive multi rate (AMR) Speech Traffic Channels", 1999.

[28] J. Dines, J. Vepa, T. Hain, "The segmentation of multi-channel meeting recordings for automatic speech recognition", in Int. Conf. on Spoken Language Processing (Interspeech ICSLP), pp. 1213–1216, 2006.

[29] S. Hari Krishnan Parthasarathi, P. Motlicek and H. Hermansky, "Exploiting contextual information for speech/non-speech detection", in Text, Speech and Dialogue, Brno, Czech Republic, pp. 451–459, Springer-Verlag Berlin, Heidelberg, 2008.

[30] A. Benyassine, E. Shlomot, H. Su, ITU Recommendation G.729 Annex B, "A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications", IEEE Comm. Mag., pp. 64–73, 1997.

[31] D. Vergyri et al., "The SRI/OGI 2006 spoken term detection system", in Proc. of Interspeech, pp. 2393–2396, Belgium, 2007.

[32] T. Hain, et al, "The AMI system for the transcription of speech in meetings", in Proc. of ICASSP, pp. 357–360, Hawaii, USA, 2007.

[33] NIST Spoken Term Detection (STD) 2006 Evaluation Plan, http://www.itl.nist.gov/iad/mig//tests/std/2006/docs/std06-evalplan-v10.pdf

[34] I. Szoke et al., "BUT system for NIST spoken term detection 2006 – English", in Proc. of NIST Spoken Term Detection (STD) Workshop, pp. 15, Washington D.C., USA, 2006.

[35] P. Motlicek, "Automatic out-of-language detection based on confidence measures derived from LVCSR word and phone lattices", in Proc. of Interspeech, Brighton, England, 2009.

[36] Z. Zhengyou, "A flexible new technique for camera calibration", in IEEE Transaction on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

[37] D. G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece, pp. 1150–1157, 1999.

[38] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns", in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

[39] I. Laptev, T. Lindeberg, "Space-time interest points", in ICCV, 2003.

[40] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, "Behavior recognition via sparse spatio-temporal features", in VS-PETS, 2005.

[41] B. D. Lukas, T. Kanade, "An iterative image registration technique with an application to stereo vision", in IJCAI, pp. 674–679, 1981.

[42] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual word ambiguity", in PAMI, vol. 32, no. 7, pp. 1271–1283, 2010.

[43] C. Sanderson and K. K. Paliwal, "Information fusion and person verification using speech and face information", Idiap Research Report IDIAP-RR 02-33, 2002.

[44] M. Slaney, and M. Covell, "Facesync: a linear operator for measuring synchronization of video facial images and audio tracks", in Proc. of Neural Information Processing Systems, pp. 814–820, 2000.

[45] G. Monaci, O. D. Escoda, and P. Vandergheynst, "Analysis of multimodal sequences using geometric video representations", in Signal Processing, vol. 86, pp. 3534–3548, 2006.

[46] D. Korchagin, P. Motlicek, S. Duffner, and H. Bourlard, "Just-in-time multimodal association and fusion from home entertainment", in Proc. IEEE International Conference on Multimedia & Expo (ICME), Barcelona, Spain, 2011.

[47] J. Hershey and J. Movellan, "Audio vision: using audio-visual synchrony to locate sounds", in Proc. of Neural Information Processing Systems, pp. 813–819, 1999.

[48] H. Nock, G. Iyengar, and C. Neti, "Speaker localisation using audio-visual synchrony: an empirical study", in Proc. of CIVR, Urbana-Champaign, USA, 2003.

[49] M. Gurban and J. Thiran, "Multimodal speaker localization in a probabilistic framework", in Proc. of EUSIPCO, Florence, Italy, 2006.

[50] F. Kuech, et al., "Acoustic echo suppression based on separation of stationary and non-stationary echo components", in Proc. of Acoustic Echo and Noise Control, Seattle, USA, 2008.

[51] D. Vijayasenan, F. Valente and H. Bourlard, "An information theoretic combination of MFCC and TDOA features for speaker diarization", in IEEE Trans. on Audio Speech and Language Processing, 19(2), 2011.

[52] T. Hain et al., "The AMI meeting transcription system: progress and performance", in Proceedings of NIST RT06 Spring workshop, 2006.

[53] T. Hain et al., "The 2007 AMI(DA) system for meeting transcription", in Lecture Notes in Computer Science, ISSN 0302-9743 (Print) 1611-3349 (Online), vol. 4625/2008, Multimodal Technologies for Perception of Humans, pp. 414–428, 2008.

[54] T. Hain, V. Wan, L. Burget, M. Karafiat, J. Dines, J. Vepa, G. Garau, M. Lincoln, "The AMI system for the transcription of speech in meetings", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, pp. 357–360, 2007.

[55] The history of automatic speech recognition evaluations at NIST, http://www.itl.nist.gov/iad/mig/publications/ASRhistory/

[56] T. Hain et al., "The AMIDA 2009 meeting transcription system", in Proceedings of Interspeech, Makuhari, Japan, 2010.

[57] D. Korchagin, P. N. Garner, and P. Motlicek "Hands free audio analysis from home entertainment", in Proc. of Interspeech, Makuhari, Japan, 2010.

4 Multimedia Structuring and Content Extraction

This chapter reports on several novel techniques devised to enhance the content annotation stage in the MyVideos concept demonstrator. The MyVideos demonstrator explores asynchronous social sharing and storytelling. Using a school concert as the guiding theme, the demonstrator allows short video clips filmed by audience members to be compiled automatically into personalised stories.

The demonstrator aims at providing new ways to interactively explore a media collection, using visual representations of annotations or by interacting with a narrative in real time. Additionally, a collaborative tool for authoring and sharing personal stories is explored. It allows parents to easily author and share videos of school performances of their children through a common repository. Essentially, it allows people who are apart to share video stories about an event in which they all took part, extending the habit of talking about such events when together.

Several techniques were devised to enhance the content annotation stage:

- a content synchronisation (section 4.1), a crucial step in the MyVideos workflow;
- techniques to facilitate the annotation process and improve the quality of annotations (section 4.2), including an update on the detection of severe unusual material that has the same appearance as shot boundaries – like very rapid camera movements, heavy unsteadiness or people crossing the picture close to the camera – and generates a set of annotations, using MPEG-7 standard;
- mechanisms to extract higher-level semantic annotations (section 4.3), such as face detection and tracking for MyVideos;
- MyVideos database structure (section 4.4) to store all the annotations and other relational data.

Following a careful analysis of the results obtained from the MyVideos evaluations, a new set of use cases has been proposed to address the key challenge of helping to build social relationships through the collaborative authoring and sharing of media which people care about. As a result, three new user-facing software components have been proposed and developed: interactive narrative exploration, visual vault exploration and authoring/editing. Figure 50 shows these key components envisaged for MyVideos and how they are anticipated to fit within a simple user workflow. The inputs to and outputs from stages are shown in red.

The first stage, shown at the top, provides the mechanism by which digital media is captured during an event, and subsequently processed to create a collection of media clips which are accompanied by structured annotations. The structure and semantics of these annotations are dictated by the requirements of the subsequent stages. Sections 4.1, 4.2 and 4.3 describe in detail how the capture and annotation workflow has been enhanced in MyVideos.

The exploration stage, shown in the bottom left, includes two new components, which will be used by triallists, and aim to meet elements of all three chosen use cases by providing enhanced ways to access a vault of digital media, while making use of personalised narratives and collaborating with others. Interactive narrative exploration focuses on automatically compiled narratives, which can be influenced during playback by user interactions and which can be stored, subsequently edited and adjusted in collaboration with others. Similarly, user interactions can be stored as an indication of the changing nature of their "profile" while watching the narrative. In Figure 50 this is represented by the stylised graph showing how a user's interest in keywords "Julia" and "saxophone" may change over time. The visual vault exploration focuses on a highly structured, visual representation of the digital media available, which helps the user to explore by using personalised recommendations. As with interactive narrative exploration, user interactions can be used to indicate a "profile", but these are more likely to be captured as individual values rather than changing over time.

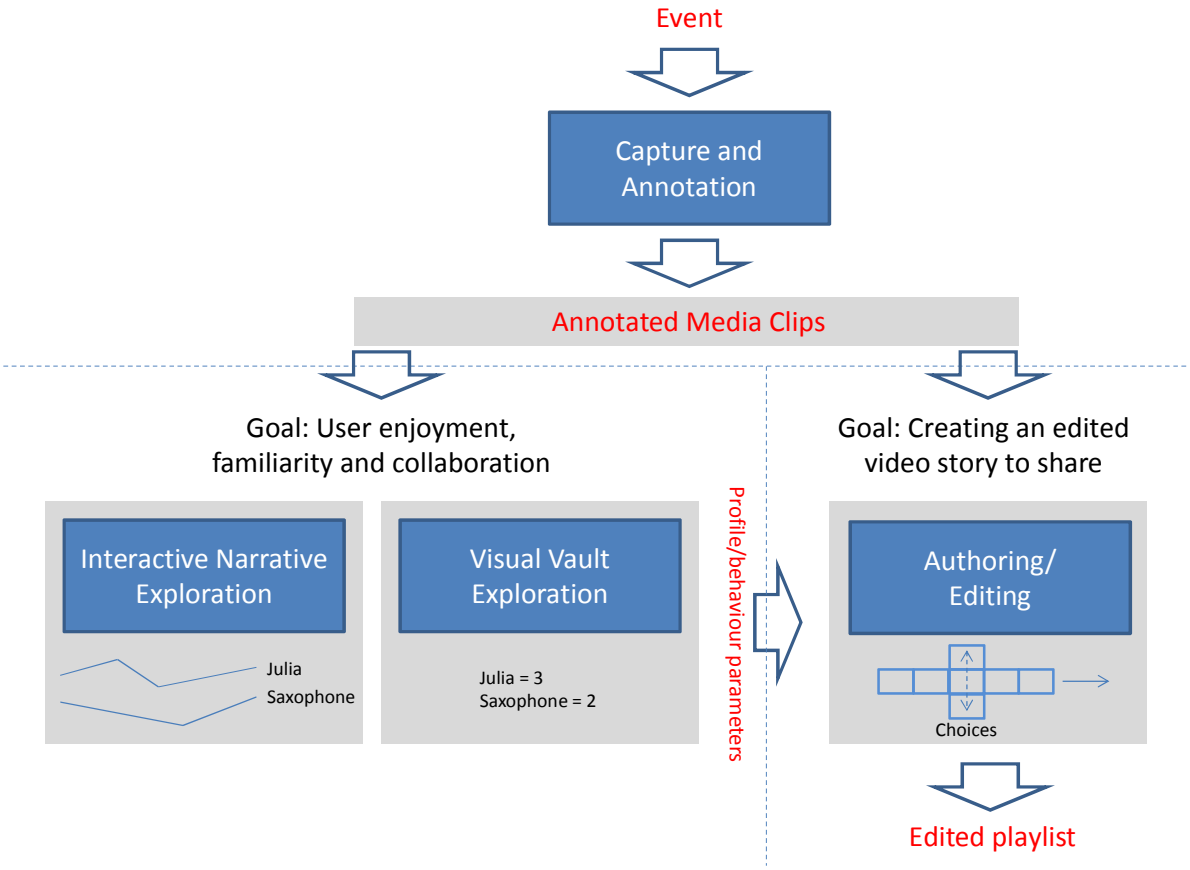

Figure 50: MyVideos components and anticipated workflow

Users of the MyVideos system may only choose to make use of the exploration components to discover and gain familiarity with recorded media clips. However, another key lesson learned from the MyVideos evaluations related to the opportunity to collaboratively create and share, and the editing stage, shown in the bottom right, is focused on this aspect. The authoring/editing component can make use of knowledge gained from the exploration components, in the form of a user's profile, to present a personalised narrative, which can be edited and customised on a per clip basis. The component ambitiously brings together both automatic and manual processes, so that narrative segments can be recompiled, adjusted and re-edited successively. The resulting compositions can be shared and modified by others at any point in this process if desired.

The capture and annotation workflow is explained in more detail in the reminder of this chapter. For more details of the achievements of the MyVideos concept demonstrator throughout the TA2 project, please also refer to D3.5 "Summary report – application design and implementation".

4.1 Content synchronisation

In a professional scenario, one might expect to be able to use multiple capture devices, and for them all to be synchronised via a common clock or similar [1]. Consumer level devices, however, do not normally provide such capabilities and are turned on and off at the will of their users. This leaves us with the metadata and audiovisual signals to infer synchronisation information. The available camera time and recording time in the metadata are based on the personal capturing devices and are most likely to be different across the devices. Another intuitive and simple approach would be to compare the corresponding audiovisual signals. However, recordings captured at the same time by different cameras may look and sound different because of camera locations (e.g. different lighting ambience, noisy surrounding), camera settings (e.g. white point balance, audio gain), quality of the camera components (e.g. sensor, lens, microphones). Therefore, raw audiovisual signals are not suitable for synchronisation purpose. The solution would be to automatically synchronise the multisource recordings by detecting and matching audio and/or video features extracted from the content.

Early studies on video-based synchronisation techniques [2], [3] relied on assumptions of static cameras and homographic images. In [4] a usage of tracking a line feature in multiple videos with limited camera motion is used, though the method implies identical frame rate on all cameras. In [5] moving features are computed that best relate with the pre-computed camera geometries, nevertheless the method depends on sufficient texture for tracking and other constraints. In [6] authors propose a synchronisation based on flash sequences, which is suitable only for particular type of events. Other state-of-the-art video-based synchronisation techniques [7], [8], [9], [10] also impose controlled environments. Therefore, if the devices are hand-held and environments are unconstrained, we cannot rely in any predictable sense on the video signal. This leaves us with the audio signal from which to infer synchronisation information.

One of audio-based solutions is the use of audio onsets [11], which are the perceived starting points in an auditory event. Many other solutions rely on audio fingerprinting techniques [12], [13], [14], which result in fairly good but not perfect synchronisation of the recordings. In our studies [15], [16] we have shown that the auxiliary signals can be synchronised with the reference signal reliably based on audio perceptual features typical of ASR applications.

Consider a music performance. The duration of the corresponding event can easily be of the order of a small number of hours. It is normal in such situations to decrease the search space, retaining only useful information for synchronisation. In our study [15] we have shown that multiple recordings can be synchronised to an acceptable accuracy and corresponding confidence can be reliably estimated. For recordings longer than 15 s we were able to achieve 100% precision on 100 recording dataset for time-quefrency signatures without excitation frequency versus 98% for fast cross correlation. For recordings shorter than 5 s the precision levels were lower due to limited length of the signatures and the real world variability of the data (noise, reverberation, non-stationarity of cameras, etc). The estimated precision dependency on the length of test signals in respect to enlarged dataset (997 test recordings) is shown in Figure 51.

We define time-quefrency signatures as time-quefrency matrices based on normalised truncated mel-cepstral vectors in steps of 10 ms. A 256 point discrete Fourier transform (DFT) is performed on overlapping audio frames of 16 ms in steps of 10 ms and squared to give the power spectrum. The resulting 129 unique bins are then decimated using a filter-bank of 23 overlapping triangular filters equally spaced on the mel-scale. The mel-scale corresponds roughly to the response of the human ear. A logarithm and DFT then yield the mel-cepstrum [17]. Lower 13 dimensions retain the energy and general spectral shape, while higher dimensions retain excitation frequency [18], which is normally truncated. We keep higher mel-cepstrum coefficients, related to excitation frequency, to estimate corresponding impact on short-term recording synchronisation and confidence estimation. The energy is truncated for proposed approach, though kept for a subset of other considered signatures. Next,

Cepstral Mean Normalisation (CMN) is performed by subtracting from each cepstral vector the mean of the vectors of the preceding (approximately) half second. This has the effect of removing convolutional channel effects. Finally, if the norm of a vector of the mean normalised cepstral coefficients is higher than 1, then the vector is normalised in Euclidean space. This gives us the reduced variance of the search distance space. Synchronisation, based on the above time-quefrency signatures, is performed by searching for a best distance [15] in n-dimensional Euclidean space between the time-quefrency representations.

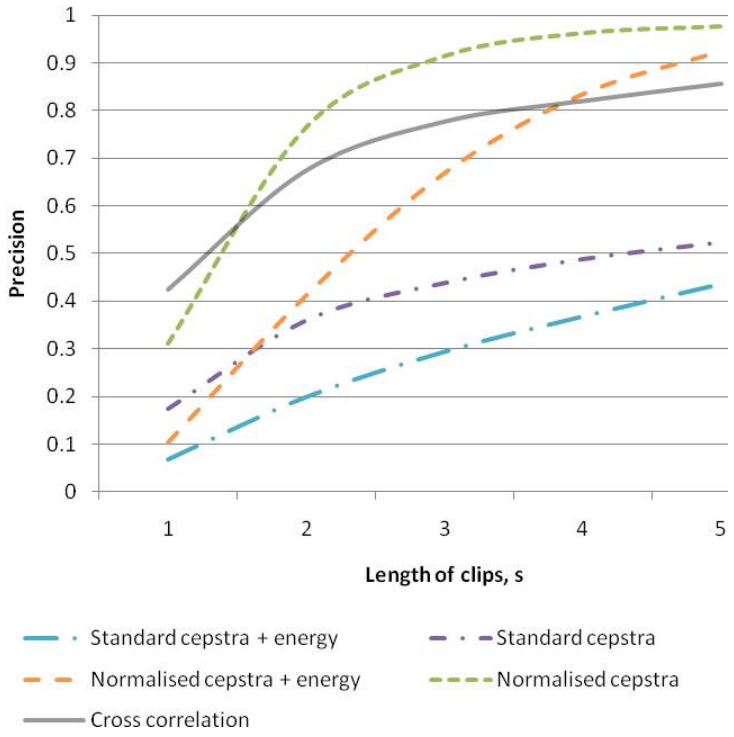

Figure 51: Precision versus test signal length

The confidence of the above techniques can be estimated as a measure of relative variance of the search space via standard deviation. For time-quefrency signature based technique, the standard deviation can be replaced by the maximum distance [15]. Thus the synchronisation confidence can be estimated by searching for a confidence corresponding to a best distance in n-dimensional Euclidean space between time-quefrency representation of test and reference signals.

It is worth mentioning that the use of standard cross correlation instead of fast cross correlation is not feasible as it is computationally onerous (several days per test signal instead of few minutes on an Intel Core 2 CPU 6700 2.66GHz).

The results presented in this section were achieved on a real life dataset of 1010 recordings, which includes 13 reference signals (total length – 13 h 31 min) and 997 test signals. The reference signal contents consist of musical concerts/rehearsals with multiple events/replays one after the other. All corresponding audio tracks were extracted and converted to 16 kHz mono PCM files with FFMPEG software [19].

Experiments were conducted on a closed set (i.e. we did not consider test signals that did not correspond to the reference signal). Nevertheless according to our previous study on a rejection mechanism [20], the proposed approach can be successfully extended to an open set.

To avoid possible inaccuracy associated with manual annotation (the ear is insensitive to delays below 160 ms) and limited speed of sound (each 10 m distance from the object results in 1 frame lag) the precision was calculated as the number of correctly (within ±5 frames) synchronised clips divided by the total number of test clips. This is a bit wider range than ITU-R recommendation [21], proposing the range between -125 ms and +45 ms as a requirement for editing multiple recordings without losing lip synchronisation. While theoretically it is feasible to reduce our experimental range for scoring to fit ITU-R recommendation, it would require a lot of additional work to update annotations in respect to required ITU-R accuracy.

In Figure 48 we illustrate how the dimensionality of the feature vector including excitation frequency range influences precision of short-term recording synchronisation.

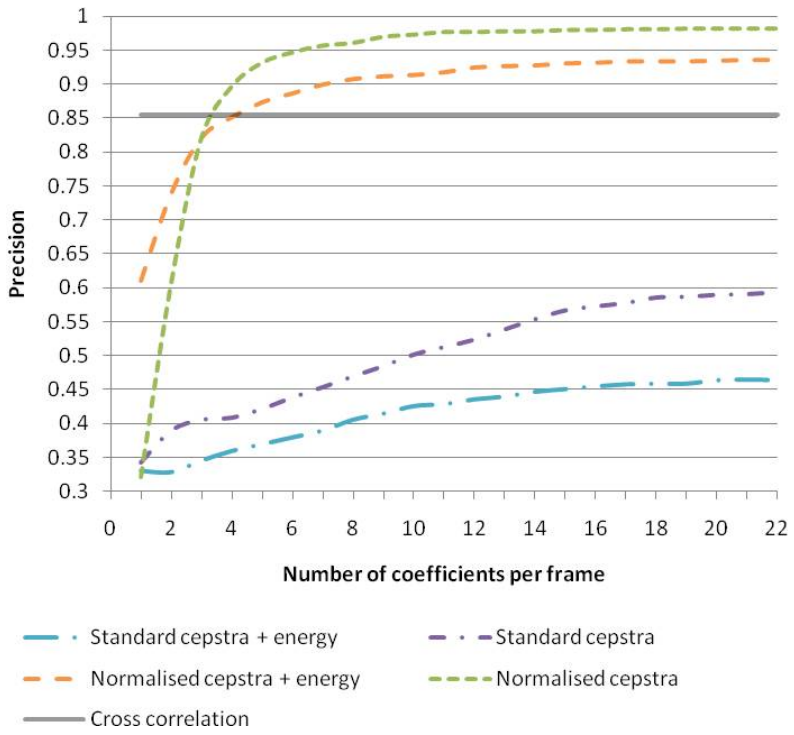

Figure 52: Precision versus number of coefficients

It is clearly visible that the precision improves with increasing cepstral analysis order. Precision for lower 12 dimensions, corresponding to the general spectral shape, results in 97.69%, while additional 7-10 coefficients, corresponding to excitation frequency range, allows to increase precision level of synchronisation up to 98.19%. I.e. we observe absolute improvement by 0.5% in the case of excitation frequency use. This in turn corresponds to 21.6% relative improvement in respect to error rate achieved on described dataset and based on the technique from our previous study [15] (from 2.31% to 1.81%). However, precision is lower when the energy is considered or normalisation in Euclidean space is excluded. We hypothesise this is due to the increased variance of the search distance space.

In Figure 53 we illustrate how the dimensionality of the feature vector including excitation frequency range influences confidence estimation distribution for recordings 5 s length. Here we consider only the case when energy is excluded and normalisation in Euclidean space is applied. The graph contains in total 21’934 confidence estimates for both positive (green dots) and negative (red dots) classes of synchronisation. The positive class is defined as set of test signals, properly synchronised with the reference signal. The negative class is defined as set of test signals, misaligned with the reference signal. While we observe positive impact of excitation frequency on reducing negative class, we have

to state that corresponding negative class is becoming wider and sparser. Also an optimal separation of positive and negative classes for short-term recordings is much trickier than if we would have the recordings of 30+ s. One of generic solution for this two class classification problem would be the use of machine learning approach, e.g. the support vector machine [22]. Nevertheless, depending on subsequent application, the weights for corresponding classes can be different. This is why, it is important to know not only confidence estimates distribution, but the dependency between precision and recall values.

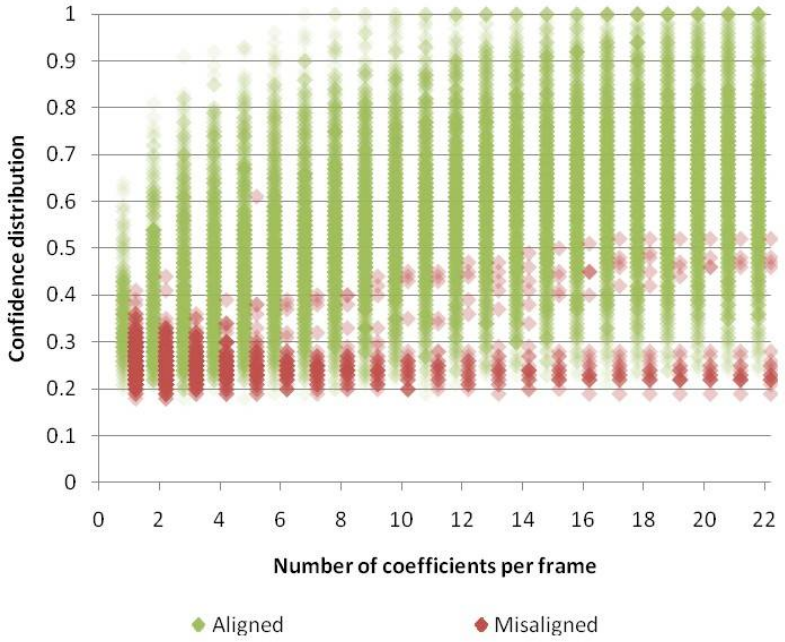

Figure 53: Confidence distribution versus number of coefficients

Dependency between precision and recall values can be estimated experimentally via application different confidence threshold values. In Figure 54 this dependency is illustrated for 9 selected cases. Precision is defined as the number of true positive test signals (test signals correctly detected as belonging to the positive class) divided by the total number of test signals detected as belonging to the positive class (the sum of true positive and false positive test segments). Recall is defined as the number of true positives test signals divided by the total number of test signals that actually belongs to the positive class (the sum of true positive and false negative test signals). Prefix “standard” means no normalisation in Euclidean space is performed. Prefix “normalized” denotes normalisation in Euclidean space is performed. Signatures with lower 12 dimensions, corresponding to the general spectral shape, are marked as “cepstra”. Signatures with lower 22 dimensions, corresponding to the general spectral shape and excitation frequency, are marked as “cepstra + excitation”. Signatures with lower 12 dimensions and energy are marked as “cepstra + energy”. Signatures with lower 22 dimensions and energy are marked as “cepstra + energy + excitation”. To allow better positioning with other techniques we present the results for well-known fast cross correlation method as well.

It is clearly visible, that 4 out of 8 time-quefrency signature based techniques for confidence estimation perform better than confidence estimation based on fast cross correlation. The best result belongs to the case when the general spectral shape is combined with excitation frequency and normalised in Euclidian space (double square dot green line). We were able to achieve 99.08% precision (versus 97.79% for the general spectral shape without excitation frequency) in the case of 100% recall and 76.00% recall (versus 75.46% for the general spectral shape without excitation frequency) in the case of 100% precision for confidence estimation.

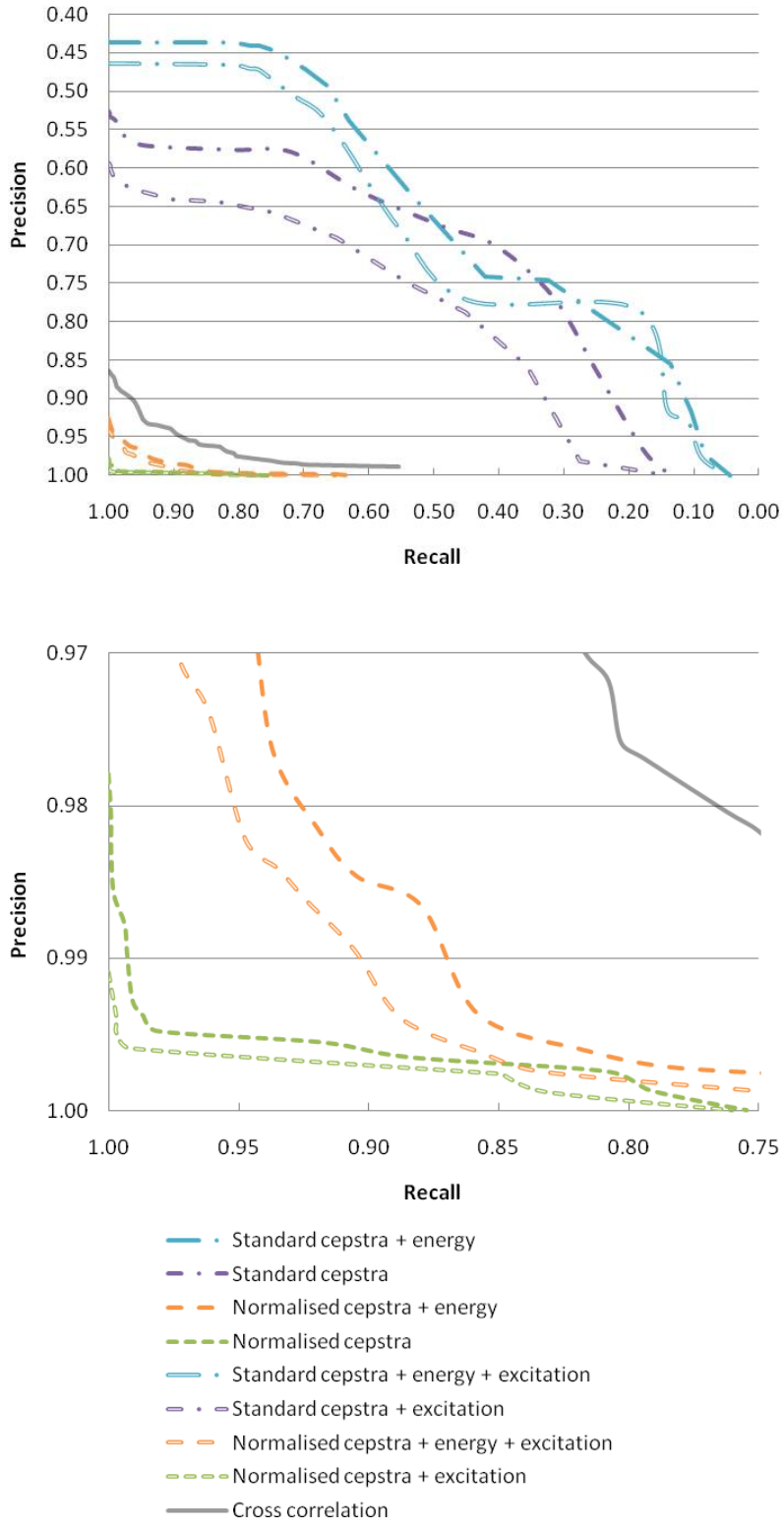

Figure 54: Precision versus recall for confidence estimation

Processing time (on an Intel Core 2 CPU 6700 2.66GHz) for the implemented algorithm without multi-core optimisation was 25 seconds for automatic synchronisation of a 5 second test signal over the 51 min reference signal using the general spectral shape and excitation frequency, 14 seconds for the same test signal using the general spectral shape only, and 70 seconds for the same test signal using fast cross correlation technique. It is directly proportional to the length of the test signal, to the length of the reference signal and to the feature vector dimensionality. Thus we can conclude that computational efficiency of proposed approach is even better than fast cross correlation and memory requirement is about 28% of the size of reference signal (28 MB versus 3 GB for fast cross-correlation). There is clearly a trade-off between desirable precision/recall levels and execution time / memory requirements. By lowering the cepstral order we can surely reduce execution time, memory requirements, and precision/recall levels.

In addition to synchronisation, we have shown [23] that time skew can be successfully detected and corrected as well and the implemented synchronisation algorithm is sustainable to time skew issue. Time skew detection corresponds to estimation whether all recordings in the same session have the same absolute time velocity or not (see Figure 55). Therefore it is enough to answer the question whether the relative time velocity ratio between the recordings from the same session equals to 1.0 or not. Nevertheless to be able to perform time skew correction the relative time velocity ratio has to be estimated precisely. The relative time velocity ratio is estimated as a relation of time distances between gravity points from different recordings.

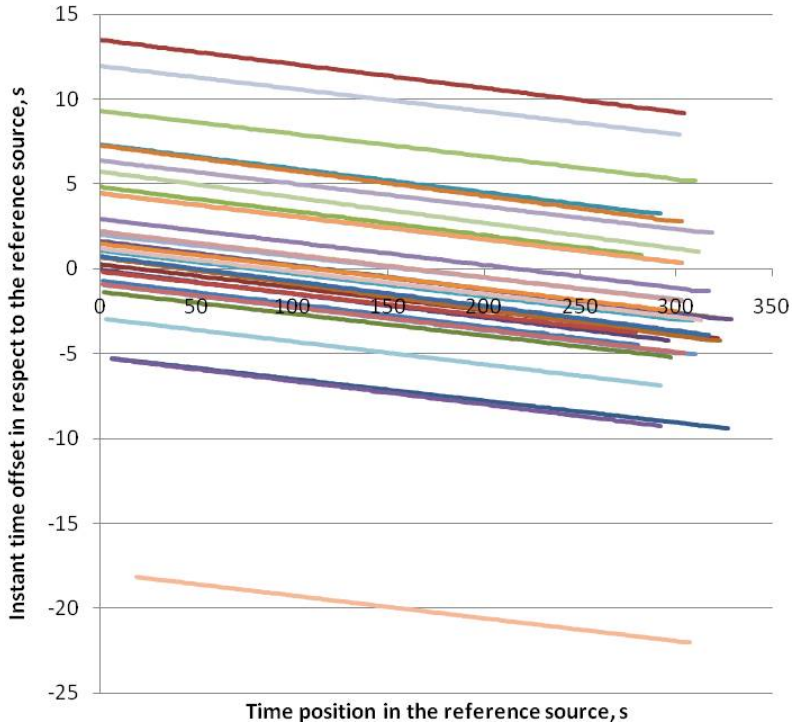

Figure 55: Estimated time skew trajectories for sessions with time skew issue

According to the definition of time-quefrency signatures we observe very good precision due to information taken from both domains: temporal and cepstral. Due the presence of the time skew issue the temporal information does not match precisely anymore. Therefore the longer signatures after a certain point should results in lower precision and, accordingly, in lower confidence. In Figure 56 we illustrate how the length of the signature influences the confidence measure in case of time skew presence.

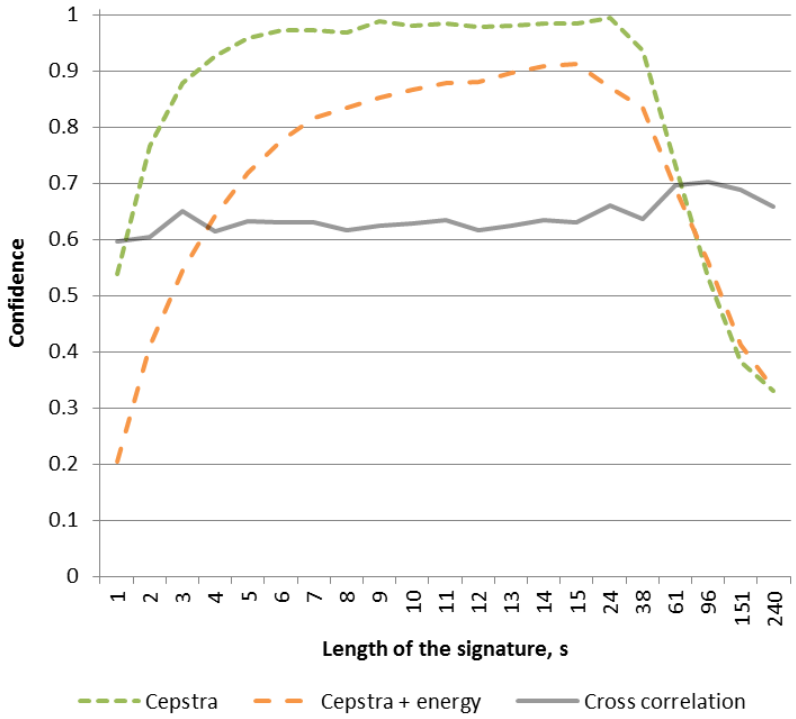

Figure 56: Confidence versus signature length in case of time skew presence

It is clearly visible that the confidence measure of defined time-quefrency signature increases with increasing the length of the signature for the first 9 seconds, keeps its average maximum during the following 15 seconds and start to decrease afterwards (dash dot line). However, the confidence measure of time-quefrency signature is lower when the energy is considered (long dash dot line). Further investigations have proved good robustness of time-quefrency based confidence measure in the case of the time skew presence and resulted in 100% of confidence precision for test data with the confidence higher than 50%. In this sense, confidence estimates for time-quefrency signatures were much more reliable than confidence estimates for cross correlation – we were not able to achieve good confidence precision even for much higher cross-correlation confidence thresholds. Further, it was found that longer signatures slightly decrease the smoothness of the estimated instant time skew trajectories. The corresponding standard deviations from expectation are shown in the table below.

Time-quefrency signature length	Standard deviation
10 s	1.3%
20 s	2.1%
30 s	3.2%

From the table we can see that the standard deviation improves with shortening of the signatures (smaller values are better). While on described dataset the optimal estimated length of the signature was 10 s, it is worth mentioning that estimated confidence distribution and standard deviation values are dependent on relative time velocity ratio and could be different across different datasets.

4.2 Content analysis and structuring

One part of the offline video analysis is to detect unusable material and thereby assist the manual creation of subclips. A plug-in was developed for the video analysis software to convert and export shot boundary, visual activity and camera motion information into a CSV file for each video (see Figure 57). In a first step, the user selects those video files in the job list view for which CSV files should be generated. When pressing the "Export" button within the "TA2Mpeg7ExtractionTool" control, a folder browse dialog appears where the user selects the output directory. A single CSV file will be generated for each video file similar to the example below.

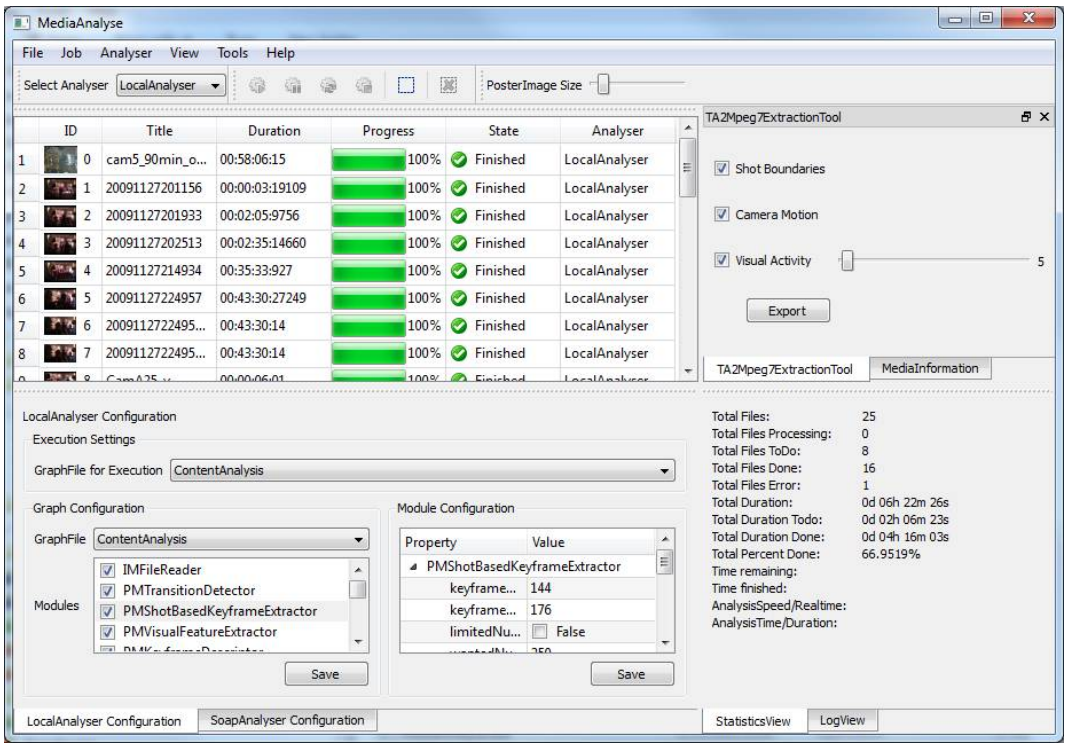

Figure 57: MediaAnalyse user interface

Example of the beginning of an MPEG-7 file transformed into CVS:

```
Shot Boundaries
00:00:49.733
00:06:46.333

Camera Motion Segments
TC In          TC Out         Type                 MeanVal
00:00:01.733   00:00:12.666   ZoomOut              269
00:03:54.400   00:03:56.200   ZoomOut              42
00:05:13.666   00:05:16.000   PanRight             164
00:07:04.666   00:07:06.666   TiltDown, ZoomOut    192

VisualActivity Segments, Threshold:5
TC In          TC Out         Avg Value
```

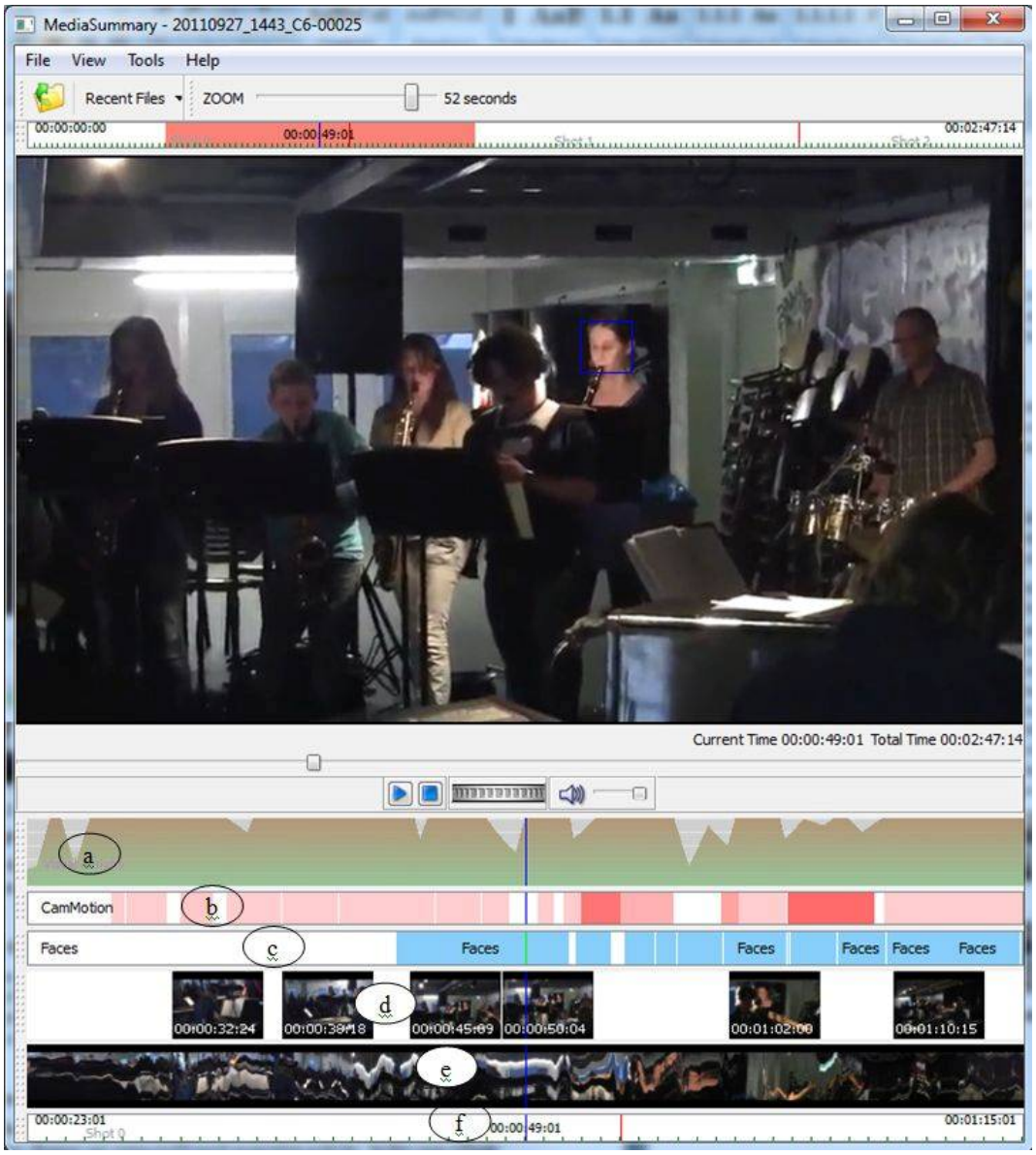

Figure 58: Updated SVAT user interface

The semantic video annotation tool (SVAT) for quickly exploring the video analysis results of a single video was tailored to the needs in the TA2 project. A screenshot is shown in Figure 58.

The following sections describe the individual modules. The face detection algorithm is described in more detail in section 4.3, the detected faces are visible as overlays in the video player, and in the face timeline view (see Figure 58, control (c)).

4.2.1 Shot boundary detection

Modern file-based cameras store each single recording in a separate file which actually contains no hard cuts or transitions. The shot boundary detection algorithm detects severe unusual material that has the same appearance as shot boundaries, like very rapid camera movements, heavy unsteadiness or people crossing the picture close to the camera. Shot boundary information is visualised in Figure 58, control (f).

4.2.2 Key-frame extraction

Key-frames are representative frames of a video. The algorithm extracts at least one representative frame within each shot, further key-frames are extracted based on motion activity. In the case of high levels motion, key-frames are extracted more densely, in case of low motion key-frames are extracted more scarcely. Key-frame information is visualised in Figure 58, control (d).

4.2.3 Stripe images

Stripe images aid the user in quickly exploring a video. Especially shot boundaries, changes in the scene, static scenes, zooms and tilts, and also the setting of the recording can be seen in the stripe images. Stripe images are visualised in Figure 58, control (e).

4.2.4 Visual activity

Visual activity is a continuous measure extracted for every second in a video. The activity may result from moving objects or from camera movements. Segments with very high visual activity can be candidates for unusable material. Visual activity information is visualised in Figure 58, control (a).

4.2.5 Camera motion

The camera motion algorithm calculates the motion of the camera in a video. Camera motion clusters are pan left, pan right, tilt up, tilt down, zoom in, zoom out, roll clockwise and roll anticlockwise. For each camera motion segment a value indicating the strength of the movement is calculated.

The original version of the camera motion algorithm was developed and tested with broadcast videos, recorded with professional cameras by professional cameramen. The type of camera movements of videos taken for the MyVideos scenario, however, differs from those in broadcast videos: zooms are more rapid because of the limitations of home user devices. Also pans are more rapid because users spot highlights and immediately direct their camera there. The camera motion algorithm was adjusted to this type of content in order to better detect more rapid camera movements. Very rapid camera movements are detected by the shot boundary detection module. Camera motion information is visualised in Figure 58, control (b).

4.3 Face detection and tracking for MyVideos

The person-centric use cases of the MyVideos scenario deal with processing of recorded content in order to deliver personalised videos. As a simple example, a concert highlights video generated for a certain viewer should focus on the actors the intended viewer has a strong social relation to, i.e. preferably use shots in which those people appear. To accomplish that, the automatic clip compilation process has to know about the relation between the viewer and the actors and has to know in which content clips the actors appear. The video analysis module described in the following is tackling the latter requirement by attempting to detect faces in the content.

The resulting information is put into the MyVideos database where the clip selection process can directly use it for improving the selection behaviour through queries in the NSL structures. Besides that input to the knowledgebase, the information may be used in a number of different ways – in the visual vault for visualisations, for improvement of keyframe selection working on the hypothesis that keyframes with persons visible are more useful for contribution to the detection of unusable content (visual quality good enough for face detection ensures a certain quality), etc.

For the implementation of the module, the existing MediaAnalyze tool (also called "SVAS" previously) was extended by modules for *face detection* and *face tracking* and a user interface for semiautomatic clustering. It realises a semiautomatic process in 4 steps:

1. Detect faces.
2. Track faces.
3. Cluster faces (semiautomatic or manual).
4. Export MPEG-7 to database.

Semiautomatic means that the automatic clustering is optional – the GUI allows the user to choose if the clustering step should be performed by the algorithm or if all detected face tracks should be put in the same bin for manual classification. The reason behind this is that early evaluations indicated that the precision of the automatic clustering component is questionable due to the limited quality/resolution of the content. Details on the algorithm and the user interface will be discussed in the following sections.

Note that the component described here does not realise a typical *face recognition* process, for which a training set of images of the actors would be required but is not available. It is performing offline analysis and is not dealing with live video streams. So far, we do not have a dataset annotated with ground truth that can be used for quantitative evaluation of the whole component.

4.3.1 Problem discussion

The component was designed to efficiently detect occurrences of faces over time in content of quality specific to the MyVideos scenario. The videos processed in this scenario are typically of poor quality, filmed by unskilled and unsupervised users (user generated content, UGC), mostly using low-resolution mobile devices without tripods, capturing videos that are jiggling/shaking considerably more than professional content.

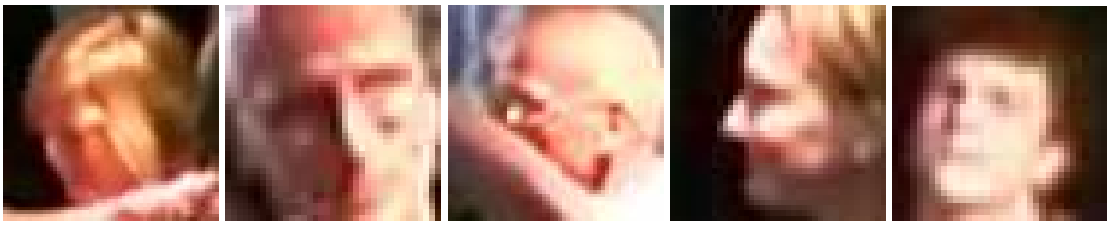

Figure 59: Examples for low quality true positive detections

Figure 60: Examples for false positive detections

The algorithm is able to detect 10+ persons in the same frame – the number of persons detected simultaneously is technically not limited; but, practically, the number of people visible without major occlusions is, though. While the aim is of course to detect all actors, a number of conditions are impeding the algorithm, including (partial) occlusions, persons only visible from behind, etc.

The recorded videos are analysed frame by frame, navigating through the video based on timestamps. For each timestamp, all persons of the video have to be localised and their position has to be tracked over a time interval. For two subsequent tracks with a certain maximum spatial gap within the frame, the algorithm should attempt to connect the tracks to a single one based on rules ensuring a low error rate. The workflow of this approach can be divided into two successive steps. The first step is offline where a set of video files is processed and face track models for each track are extracted. The second step aims to cluster similar models using a probability function, working on the hypothesis that each cluster of similar track models is belonging to one distinct person.

4.3.2 Automatic offline video analysis

The video files selected in the user interface are analysed by a face detector. The detector tries to localise each person's face for every timestamp, or, if configured otherwise, every n^{th} frame to speed up the process. A rate of every 10 frames proved to be useful. Detected faces have to be tracked over their visual appearance in time which is at least up to the next predefined detection timestamp. At this timestamp the face detector output has to overlap the tracker output to get robust face-tracker results.

The library used for the basic face detection is the well-known Viola-Jones detector, which has already been published a decade ago but is good state-of-the-art for this environment. The algorithm is designed to work quite fast (not a big factor for offline analysis anyway) and to detect faces quite *optimistically* – which means that at the cost of false positives, fewer face occurrences will be missed. The used Viola-Jones algorithm is described in [24].

The algorithm aims to detect faces both looking straight at the camera and faces captured in a profile view. This is realised by two specialised sub modules for which the results are merged.

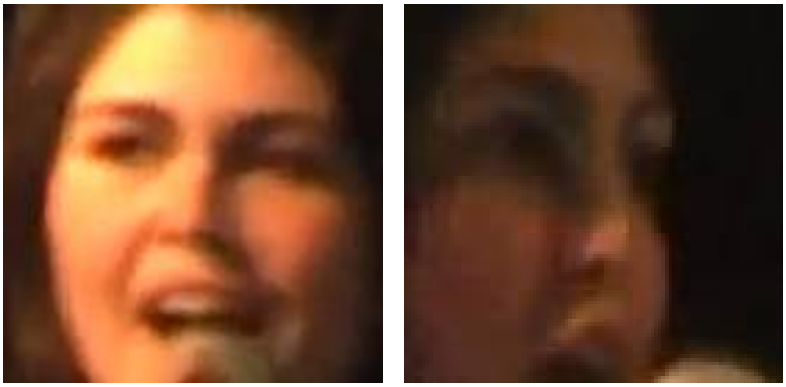

Figure 61: Examples for frontal and profile face occurrences

The component is aware of possible false positive detections and tries to minimize the error rate in the subsequent tracking step. The temporal continuity of tracks is a good indicate; too short tracks are filtered out. For the tracking process itself a pyramid block-matcher is used. Tristen Georgiou (see OpenCVWiki [25]) developed a "fast template matcher" which is a quite robust fast tracker that tends to drift from the foreground face to a background region. This tracker searches for the face in subsequent frames in the surrounding region. This is realised by the best visual match in terms of pixel differences. If there are overlaps between tracked and detected faces then the region of the detected face is used and the tracked face is *merged*.

The drift problem of the tracked region is solved by using the previously mentioned face detections at predefined timestamps as reference. By doing so, it is continuously "guaranteed" that the tracked region is a face region. Another factor for choosing this simple tracker is its robustness. Other trackers like Kalman filter [26] based or particle filter [27] based trackers may perform better on high quality videos but under the circumstances mentioned above we decided to work with this "fast match template" pyramid block-matcher.

According to the OpenCVWiki, the fast match template works as follows [25]:

1. Both target and source image are down sampled numDownPyrs times.
2. cvMatchTemplate function is called on shrunken images (uses CCOEFF_NORMED algorithm, also works with CCORR_NORMED, but we have found better results with the former).
3. The numMaxima best locations are found.
4. For each point, where a maxima was located, original source image is searched at point +/- searchExpansion pixels in both x and y direction.
5. If match score is above matchPercentage then the location and score is saved in the foundPointsList and confidencesList respectively.
6. If findMultipleTargets is true, an attempt will be made to find up to numMaxima targets.
7. (Optional) The targets can be drawn to a colour version of the source image using the DrawFoundTargets function.

The extracted person tracks of the video are described with a series of time stamps and the location of the person in the frame at the timestamp (coordinates). In the next step the algorithm tries to assign each track to a person. In detail, the face tracks are clustered, not classified, so the identity of the person is unknown in contrast to how face recognition algorithms work. In the proposed algorithm the different tracks are combined with a *hierarchical clustering*. A simple minimum distance clustering is used, which is also referred to as *single linkage hierarchical clustering* or *nearest neighbour* clustering. Inputs for this cluster-algorithm are similarities between the tracks.

A range of research papers (e.g. [32], [33], [34]) propose face-recognition algorithms with support vector machines (SVM). The similarities are computed by comparing the first (frontal) appearance of a face track with the (model of the) positive/negative examples (support vectors) of the classifier. For each track the algorithm's SVM classifier decides on the frontal appearances and pre-generated negative faces (extracted out of 50 different faces from different databases). If a first frontal face of a track is more similar to a track in the classifier than to the negative faces. The similarity value is extracted by the SVM classifier between the SVM models (support vectors) and the frontal face coefficients with a RBF [30] (radial basis function) kernel. After that all necessary similarities for the hierarchical clustering between all tracks are calculated.

In detail, offline SVM model extraction for each track and the negative faces consists of (algorithm steps in brackets are optional):

1. All (frontal face) images of a track and negative faces are loaded.
2. Faces are normalised to a defined size. Grey-image calculation with histogram equalisation (extraction of Gabor Wavelet [31] features from grey-images; features are normalised to values between 0 and 255 and saved in a grey-image).
3. Extraction of a predefined number of *eigenobjects* from all grey-images with principle component analysis (PCA) [28] and the extracted average image from the PCA analysis. Normalisation of the images with the following PCA analysis leads to a reduction of the dimension of the feature space proposed in [32] and [35].
4. Calculation of the coefficients for each grey-image by the extracted *eigenobjects*.
5. Normalisation of the coefficients with the min/max coefficient values for each *eigenvalue*.
6. Extraction of the SVM model with the calculated coefficients from the track's object and the negative object.
7. Storing the PCA *eigenobjects* and the average image.
8. Storing the SVM model.

The online step performs similarity calculation for every image:

1. Load PCA *eigenobjects* database and average image.
2. Load SVM classifier.
3. Get first (frontal face) image from track.
4. Normalise face to a defined size. Grey-image calculation with histogram equalization.
5. Calculation of the coefficients for the grey-image by the previous extracted *eigenobjects*.
6. Normalisation of the coefficients with the min/max coefficient values for each *eigenvalue*.
7. Similarity calculation by the SVM classifier from the negative support vectors and the track's support vectors with the previously calculated coefficients. If the SVM prediction results in a higher similarity to the track, the similarity is used for the hierarchical clustering.

The extracted similarities between each track are input for the hierarchical clustering. After the hierarchical clustering the tracks are combined into clusters via a defined similarity value. Ideally, each track cluster would belong to one certain person to get clusters with all appearances of the person over several videos as a result. Up to this point, the entire process is fully automatic. Practically, a manual step is required to correct clustering errors afterwards.

4.3.3 User interface visualisation and assignment

In addition to the automatic face tracking component a graphical user interface has been developed. The GUI provides a possibility to both visualise and correct/re-assign face tracks. For each track, an image from the start of the track is taken for display. This allows humans to visually verify the automatically generated results very quickly.

The GUI is part of the MediaAnalyze tool as discussed before. It is depicted by the screenshot in Figure 62. Below the menu there is the list of video files to be analysed, in that case only one with the title "Dvorak". The interface for face track clustering is below. Besides buttons to trigger the process, it consists of two parts resembling file explorers like the Microsoft Windows Explorer: on the left side there is a list of persons that are either created manually or automatically as described above. At the top of that list, there is an entry named "undefined" which contains all detected tracks, which have not been assigned to a person. By clicking on a person, all tracks assigned to it appear on the right side of the GUI in a grid layout. Tracks can be re-assigned to another person or the "undefined" category using drag & drop (all selected items).

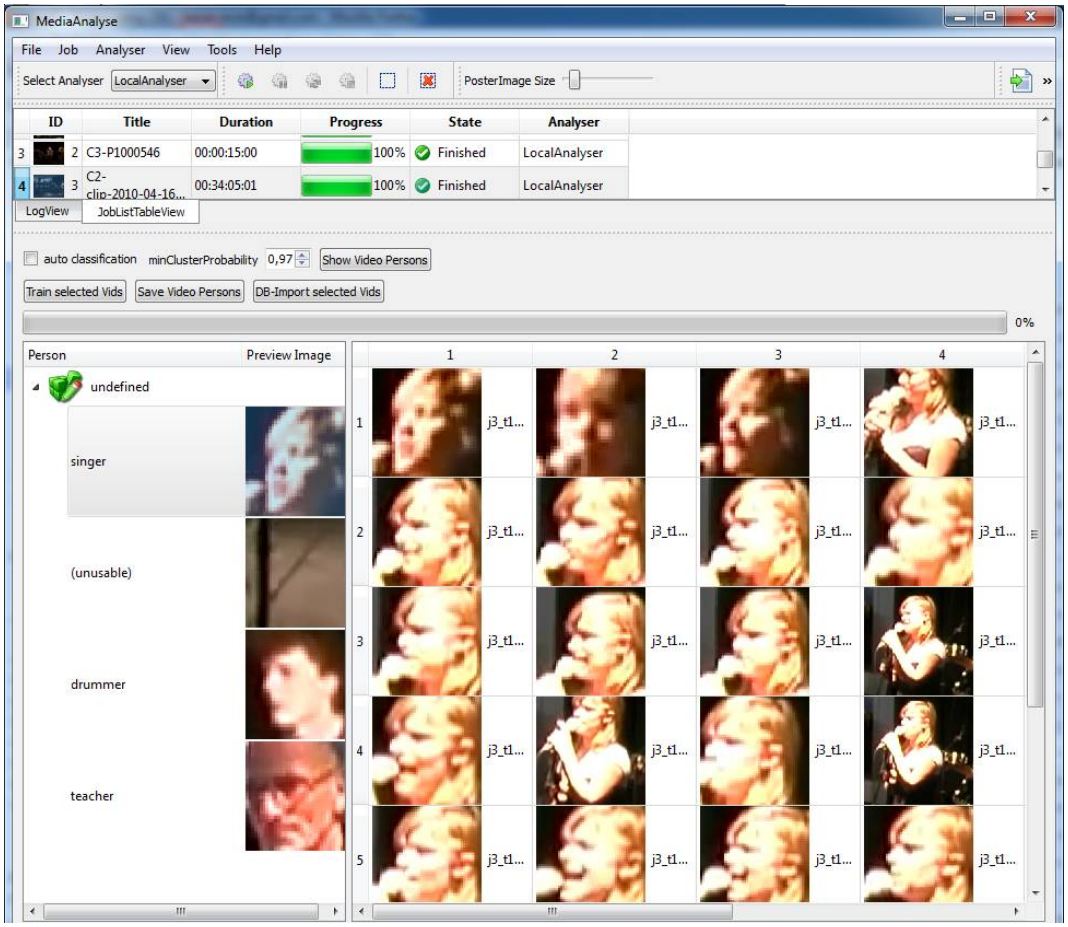

Figure 62: Screenshot of semiautomatic face clustering GUI within MediaAnalyze

The user can choose between the two modes of *semi-automatic* clustering and *fully manual* clustering of the automatically detected tracks using the "auto classification" checkbox. Using the first option, a number of clusters (persons) will be generated and the user's task is to check the results by first removing wrong tracks from clusters and then assigning unassigned tracks to the corresponding person. Wrong track detections are simply left in the "undefined" category. Overall, because of the good visualisation of tracks and the drag & drop functionality, this GUI allows to quite quickly assign face appearances to a person's identity.

4.4 Database schema

The implementation of the MyVideos front-end was envisioned around the 3 main paths mentioned in the beginning of this chapter: interactive narrative exploration of the vault (interaction with the personalised narrative in real time), visual vault exploration (evolution of the vault as a shared media space), and authoring/editing (authoring and editing via automatically generated stories). The decision was to implement each of the paths as platform independent as possible. The visual vault exploration and the editing/authoring interface have been developed as a web-based application targeting users with little technical background. From the user viewpoint this means that they only need access to the public Internet and everything runs within a JavaScript-enabled web browser on their device, which could be a desktop computer, an interactive table or a mobile device. The interactive narrative exploration path requires an improved version of the Ambulant Player as the rendering device, and therefore it is more effective to be implemented as a standalone application, rather than a web-based application. However, by directly interfacing with the player, a more immersive user experience can be achieved.

The high-level architecture of MyVideos is composed of five main components:

- A Mongrel web application server for the Ruby on Rails[2] (RoR) web application, whose role is to present the user interface and manage user interactions (visual vault exploration).
- The narrative engine, a tool which creates and updates personalised narratives on-the-fly.
- A media server that stores the recorded video clips and delivers them through HTTP video streaming.
- An improved version of the Ambulant[3] player for seamlessly rendering video playlists in the client's device (interactive narrative exploration).
- A MySQL Database[4] that stores all relational data concerning MyVideos.

The MyVideos web application makes use of a MySQL relational database to record information about a range of entities related to the event(s) and the collected video clips. Figure 63 shows an updated entity relationship model for the current system (Figure 64 shows its extension). The core entities have been maintained. For instance, video clips are known as media objects, and the temporal alignment annotations are stored in the fields *start_time_milli* and *end_time_milli*. Other annotations are described using a hierarchy of tag classes and tag instances, which form a simple taxonomy. As a reminder, tag classes represent the branches in the tree structure, while tag instances represent the leaf nodes. Annotations are then applied to individual video clips by creating a *tag instance*, which effectively links the clips to a particular tag class. An *annotations* table enables video clips to be effectively linked to tag instances, and provides extra flexibility by allowing a time period for the applicability of the annotation to be defined, along with additional information such as the user who created it. Tag classes can either be created globally to cover common themes within one or multiple events or specifically for individual events. For example the *tag_classes* table typically would hold information such as "Performer", "Instrument" and "Song". The *tag_instances* table would hold specific instances of the tag classes. For example relevant to the tag classes listed above, instances could include "Jan van Helder"; "Keyboard"; "Cry me a river". This schema supports the SQL database queries that are required by the application, such as to return lists of media objects that belong to a specific event, include a specific person, show a specific instrument or instruments, or have some combination of annotations.

[2] http://rubyonrails.org/

[3] http://www.ambulantplayer.org/

[4] http://www.mysql.com/

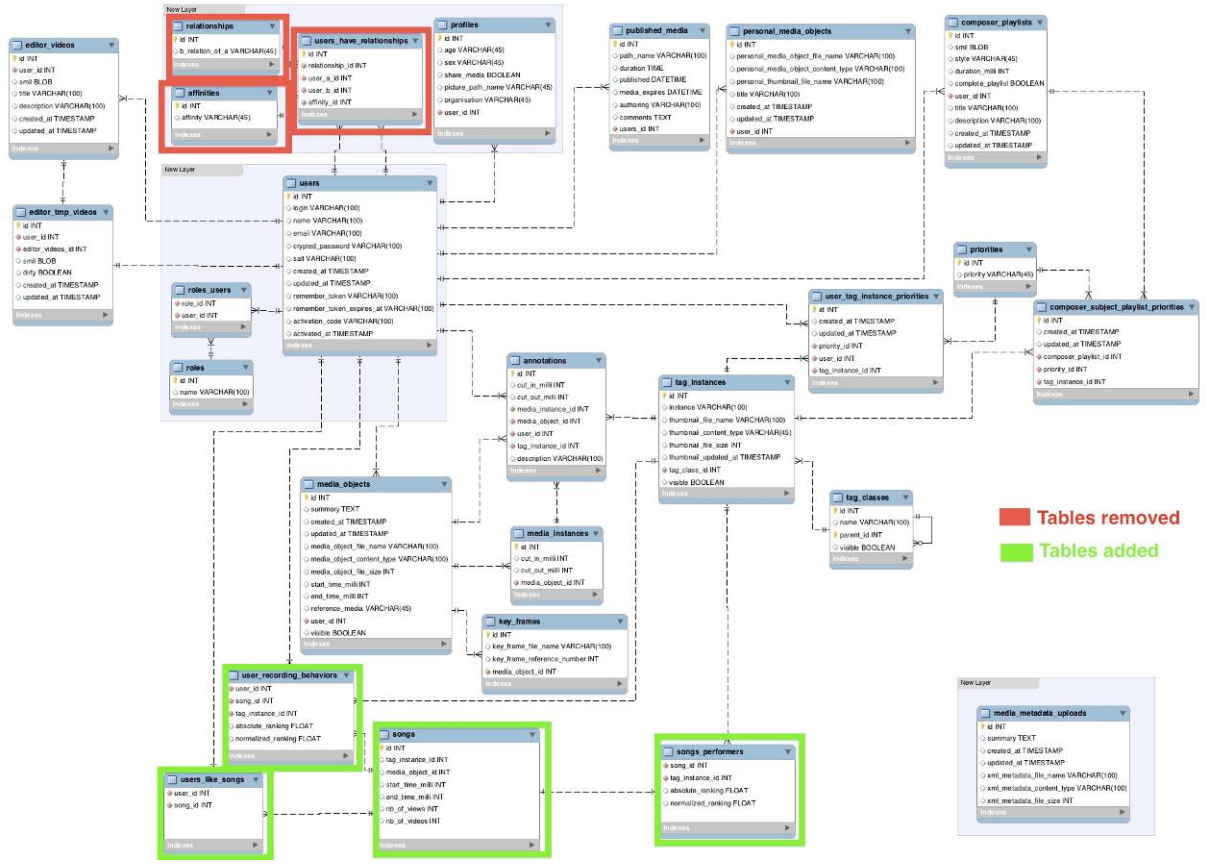

Figure 63 : Entity relationship model for MyVideos implementation

S*ongs* is a table that summarises the high level information about songs in an event. The records in the *songs* table contain the following information: a unique identifier (id), identifier of the respective *tag instance* entity, reference to the song audio file (*media_object*), start time and end time relative to the event timeline, number of users that explored the song, and number of videos in a particular song. In addition, 2 others have been created: *songs_performers* and *users_like_songs*. The first table lists performers that played in a particular song. Each performer has a rank that will be used in the visual vault exploration. The *users_like_songs* gathers user feedback by storing the number of likes for each song.

The *relationships* table has been removed, as well as the *affinities* and the *user_have_relationships* tables. These tables intended to capture the tie strength among users and performers featured in the concert videos, but they were never really used. In the current database schema, the relationships among users and performers come from the user recording behaviour, and for that a new table has been created (*user_recording_behaviors*). This information comes from the annotations created from in each video.

The results of face detection and tracking can be imported into the database according to the schema in Figure 64. For each continuous face annotation over time an entry in the *face_tracks* table is added. References to *media_objects* and *tag_instances* are resolved automatically if possible (e.g. if performer is annotated with the name according to the database entry). The fields *start_time_milli* and *end_time_milli* hold the milliseconds of the face track start and end in relation to the start of the corresponding *media_object* (milliseconds within the media file). The average face size over the whole face track in relation to the image size is stored in the *face_size* field. Values of *face_size* are between zero (exclusive) and one (inclusive).

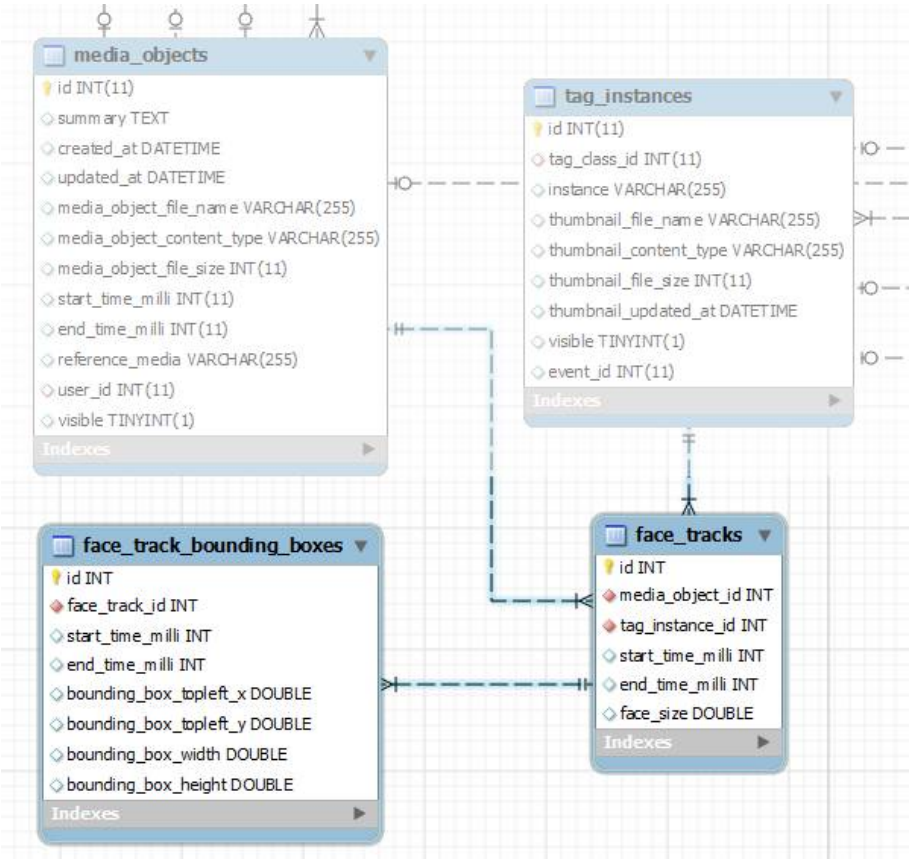

Figure 64 : TA2 database extension for face annotation

The *face_track_bounding_boxes* table is for querying the bounding boxes of a face track for a given time. For sparse annotations a bounding box entry is not added for every frame, but for a period of time if the bounding box stays static. If the bounding box is valid only for a single frame, *start_time_milli* and *end_time_milli* are equal. The bounding box coordinates (*bounding_box_topleft_x*, *bounding_box_topleft_y*, *bounding_box_width* and *bounding_box_height*) are in relation to the image size, in order to be resolution independent.

4.5 Evaluations

Recent developments in video capture & editing programs available to consumers are beginning to show how the technology tested and emulated in the first iteration of MyVideos could be integrated into consumer applications. For example, Apple's iMovie now offers editing templates with which users are guided in a choice of a sequence of "types of shot" in order to create, for example, the feel of a movie trailer. Crucially, this kind of functionality is being complemented by improving automatic analysis of media content. A key lesson learnt from the evaluation of the MyVideos system, was that a reliable form of shot type and quality detection were required as well as a system for annotation of who is present in a given video clip.

A significant element of the MyVideos workflow used the XML import-export functionality of the off-the-shelf non-linear editing package Final Cut Pro. Using style sheets it was possible to convert the output of the audio alignment tool into an XML based project file that could be imported into the Final Cut software and allow for a visual representation of all the clips aligned on the timeline. This representation was then annotated by the professional user using time line markers in the Final Cut software. These markers were subsequently exported in XML format to inform the narrative structures represented in NSL.

Given this pre-existing workflow it was decided to evaluate the latest release of Apples Final Cut Pro, version X, which is a recent release and represents a cutting edge application in the consumer sphere. This version offers new video analysis features the output of which, when combined with the results in the database of the Semantic Video Annotation Suite, could provide a powerful automatic annotation solution.

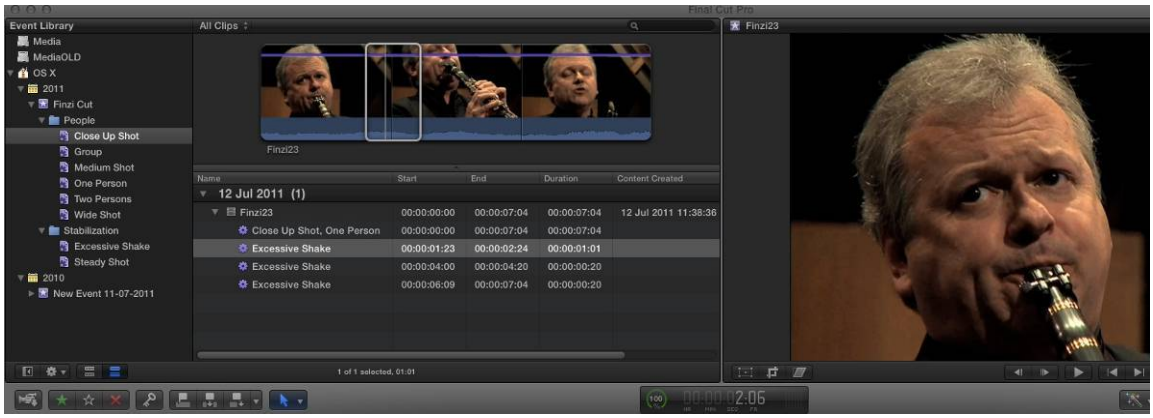

Figure 65: A screenshot of automatic annotation in Final Cut Pro X

4.5.1 Evaluation of automatic analysis in Final Cut X

Final Cut X provides the following automatic annotation functions:

1. Shot type – close up, medium shot, wide shot.
2. Number of people in the shot – one person, two persons, group.
3. Excessive shake (in, out).

We carried out an evaluation using 2 types of footage – the material from the big band concert that was the subject of the evaluation for MyVideos and some footage that was shot professionally at a professional classical music chamber concert.

The system turned out to be very effective on the professional shot footage, correctly identifying the vast majority of types of shot and number of people in shot. However, this was only the case when the

clips were short and contained one type of shot. It was much less effective when all the material was assembled together in one long clip and it could not successfully delineate all the different shots.

It was also much less effective when employed on the set of amateur MyVideos concert clips. The reason for this is a key point to note when it comes to footage capture – it was not necessarily the factor of quality (though this does have some influence) but rather the limitations in the shooting technique of the users and in the capabilities of cameras involved.

Modern smartphones can now offer consistently high quality video capture, however they are limited in their ability to zoom and users have not formed the habit of doing so. Also with a lack of experience, and also physical limitations in a concert venue, users have not formed the habit of varying the shot sizes they take. The main problem was in the MyVideos footage, when compared to the professionally shot footage, was the fact there were few genuine close-ups or mid-shots to analyse but instead a sequence of wider shots.

Readily available consumer technology is fast making the automation in editing and compiling of video material a possibility. However the analysis systems employed are not capable enough in order to exploit properly most amateur created footage and more research is required.

4.5.2 User profiling based on recording behaviour

The research has also been extended to better understand how annotations can provide cues about the user recording behaviour and social relations. As with media annotations, a profile of the user can facilitate exploration of assets. Traditional ways of profiling users include activity monitoring (log) and user personal data. While these approaches can provide relevant results for a statistically significant group of people interacting during a significant time span, they do not take into account small groups of users, and their recording behaviour. An important outcome from the MyVideos evaluations was that annotations can provide useful indicators to complement traditional profiling. Figure 66 shows the results of analysing the metadata associated to the media captured during the high school concert recorded in Amsterdam in April 2010. In this figure, the recording behaviour of a mother towards her child is compared with the average behaviour of the rest of the parents. We can notice that the affection level towards a performer greatly influences the overall time a recorder spends capturing that person. Based on the results, we can have a strong sense that recording habits and relationships provide an important estimate that needs to be considered by recommender systems.

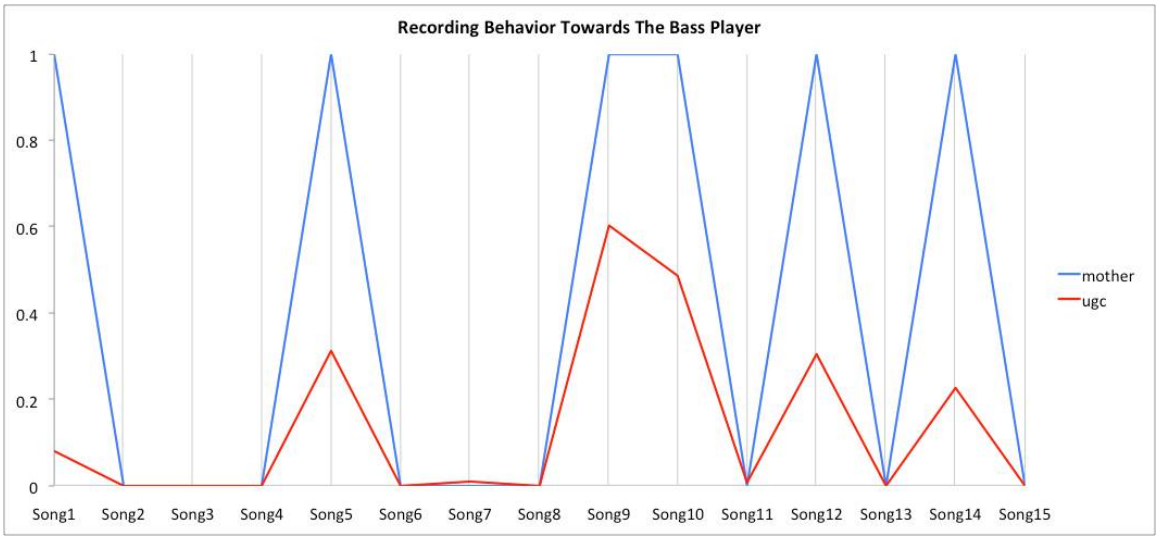

Figure 66 : Comparison between the normalised recording behaviour factor of a mother towards her daughter and the average behaviour of the rest of the parents

4.6 Conclusion

The results obtained from the evaluations show that content annotation is a key step in the content preparation process, since it allows for easier search of subjects of interest, e.g. events and performers. Nevertheless, users do not want to spend time manually annotating their own material, neither video content recorded by others. And, as shown in the previous sections, automatic annotation of user-generated content is challenging because the video encoding, the quality, and the lighting are not always optimal.

The MyVideos demonstrator to be further evaluated during autumn 2011 with pupils, parents and friends from Woodbridge School/UK (http://www.woodbridge.suffolk.sch.uk/). This is a private school located in Woodbridge, near to BT's offices at Martlesham Heath in the UK. It has a strong tradition of music with a significant number of pupils taking part in performances throughout the school year. In 2009, the school allowed the BT MyVideos team to record digital media from a concert rehearsal. The recorded media was used to carry out the first automatic analysis tests and gain important knowledge.

The school has agreed to stage a special concert, targeted for November 2011, in which musicians will perform for parents and friends, as well as members of the TA2 team. A small number of older pupils (probably aged 15-18) with parents and friends where possible is being recruited for the trial, and they will meet evaluators at least once before the concert. The team will also investigate the possibility of acquiring video and still images relating to the concert performers beforehand (i.e. during rehearsals) and afterwards (commenting on their performance).

Once the concert recordings have been captured and processed using the techniques described in this chapter, a similar laboratory-based evaluation will be held in which the triallists are invited to use the MyVideos system and provide their feedback via a combination of structured interviewing and unstructured feedback.

4.7 References

[1] J.-M. Verrier, "Audio boards and video synchronisation", in Proceedings of the AES UK 14th Conference: Audio - The Second Century, London, UK, 1999.

[2] G. P. Stein, "Tracking from multiple view points: self calibration of space and time", in Proceedings of the DARPA IU Workshop, pp. 521–527, 1998.

[3] Y. Caspi, D. Simakov, and M. Irani, "Feature based sequence-to-sequence matching", in Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 2004.

[4] C. Lei and Y. H. Yang, "Tri-focal tensor based multiple video synchronization with sub-frame optimization", in IEEE Trans. on Image Processing, 2005.

[5] A. Whitehead, R. Laganire, and P. Bose, "Temporal synchronization of video sequences in theory and in practice", in Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing, pp. 132–137, 2005.

[6] P. Shrestha, H. Weda, M. Barbieri, and D. Sekulovski, "Synchronization of multiple videos using still camera flashes", in Proceedings of the 14th ACM International Conference on Multimedia, pp. 137–140, 2006.

[7] Y. Caspi and M. Irani, "Aligning non-overlapping sequences", in International Journal of Computer Vision, vol. 48, n. 1, pp. 39–51, 2002.

[8] W. Yan and M. S. Kankanhalli, "Detection and removal of lighting & shaking artefacts in home videos", in Proceedings of the 10th ACM international conference on Multimedia, pp. 107–116, 2002.

[9] S. N. Sinha and M. Pollefeys, "Visual-hull reconstruction from uncalibrated and unsynchronized video streams", in Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium, 2004.

[10] T. Tuytelaars and L. Van Gool, "Synchronizing video sequences", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

[11] J. P. Bello, L. Daudet, S. Abdallah, et al., "A tutorial on onset detection in music signals", in IEEE Transactions on Speech and Audio Processing, vol. 13, issue 5, part 2, 2005.

[12] L. Kennedy and M. Naaman, "Less talk, more rock: automated organization of community-contributed collections of concert videos", in Proceedings of the 18th ACM International Conference on World Wide Web, pp. 311–320, 2009.

[13] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system", in Proceedings of the International Symposium on Music Information Retrieval, 2002.

[14] P. Shrestha, H. Weda, and M. Barbieri, "Synchronization of multi-camera video recordings based on audio", in Proceedings of the 15th annual ACM International Conference on Multimedia, 545–548, 2007.

[15] D. Korchagin, P. N. Garner, and J. Dines, "Automatic temporal alignment of AV data with confidence estimation", in Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, USA, 2010.

[16] D. Korchagin, "Impact of excitation frequency on short-term recording synchronisation and confidence estimation", in Proceedings European Signal Processing Conference (EUSIPCO), Barcelona, Spain, 2011.

[17] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental", in Pattern Recognition and Artificial Intelligence, C. H. Chen, Ed., pp. 374–388, Academic, New York, USA, 1976.

[18] L. Rabiner, B.-H. Juang, "Fundamentals of Speech Recognition", Prentice Hall, Upper Saddle River, NJ, USA, 1993.

[19] Open source multiformat multimedia conversion tool "FFMPEG", http://www.ffmpeg.org

[20] D. Korchagin, "Out-of-scene AV data detection", in Proceedings IADIS International Conference on Applied Computing, vol. 2, pp. 244–248, Rome, Italy, 2009.

[21] BT.1359-1, "Relative timing of sound and vision for broadcasting", in Recommendation ITU-R, 1998.

[22] V. N. Vapnik, The nature of statistical learning theory, Springer, 2nd edition, 2000.

[23] D. Korchagin, "Automatic time skew detection and correction", in Proceedings International Conference on Signal Acquisition and Processing (ICSAP), vol. 1, pp. 363–366, Singapore, 2011.

[24] P. Viola and M. Jones, 2001, "Rapid object detection using a boosted cascade of simple features", in Proceedings of CVPR, Hawaii, USA, 2001.

[25] T. Georgiou, OpenCVWiki, http://opencv.willowgarage.com/wiki/FastMatchTemplate, accessed 05/2011.

[26] G. Welch, G. Bishop, "An introduction to the Kalman filter", University of North Carolina, Department of Computer Science, TR 95-041, 1995.

[27] J. H. Kotecha, P. Djuric, "Gaussian particle filtering", IEEE Transactions Signal Processing 51 (10), 2003.

[28] I. T. Jolliffe, "Principal component analysis", in Springer Series in Statistics, 2nd ed., Springer, NY, XXIX, ISBN 978-0-387-95442-4, 2002.

[29] B. Schölkopf, A. Smola, "Learning with kernels: support vector machines, regularization, optimization, and beyond (adaptive computation and machine learning)", MIT Press, Cambridge, MA, ISBN 0-262-19475-9, 2002.

[30] N. Cristianini, J. Shawe-Taylor, "Kernel methods for pattern analysis", Cambridge University Press, Cambridge, ISBN 0-521-81397-2, 2004.

[31] J. Zhu, M. I Vai and P. Un Mak, "A new enhanced nearest feature space (ENFS) classifier for Gabor wavelets features-based face recognition", ICBA, in Lecture Notes in Computer Science 3072, Springer. pp. 124–131, 2004.

[32] M. Safari, M. T. Harandi, B. N. Araabi, "A SVM-based method for face recognition using a wavelet PCA representation of faces", in Proceedings International Conference on Image Processing (ICIP), vol. 2, pp.853–856, 2004.

[33] D. R. Kisku, H. Mehrotra, J.K. Sing, P. Gupta, "SVM-based multiview face recognition by generalization of discriminant analysis", in Proceedings of CoRR, 2010.

[34] K. Delac, M. Grgic, M. Stewart Bartlett, "Recent advances in face recognition", IN-TECH, ISBN 9537619346, 2008.

[35] M. Turk, A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neurosicence, vol. 3, no. 1, pp. 71–86, 1991.

