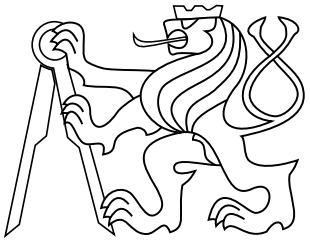
bachelor's thesis

Audio-Visual Speech Activity Detector

Hana Šarbortová



January 2015

Ing. Jan Čech, PhD.

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Bc. Hana Šarbortová**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Audio-vizuální detektor řeči**

Pokyny pro vypracování:

Speech (or lip) activity detector [1] is an algorithm that automatically identifies whether a person in a video speaks at a time. This task is important in audio-visual diarisation problem [2,3], and in audio-visual cross modal identity recognition/learning. There are typical situations when multiple people are in a camera field of view and an audio signal is perceived. The objective is speaker identification.
The speech activity detector can be visual only (based on observing the lip motion in the image), or audio-visual (which exploits the audio-visual synchrony between these modalities [4]).
In the diploma thesis, do
(1) Review the state of the art in lip activity detection
(2) Modify an existing well performing algorithm (or design a new or combined one),for speech activity detection.
(3) Collect a dataset of training examples.
(4) Evaluate the performance of the proposed algorithm.
A code for precise localization of facial landmarks [5], and [6] will be provided.

Seznam odborné literatury:

[1] K. C. van Bree. Design and realisation of an audiovisual speech activity detector. Technical Report PR-TN 2006/00169, Philips research Europe, 2006.
[2] Felicien Vallet, Slim Essid, and Jean Carrive. A Multimodal Approach to Speaker Diarization on TV Talk-Shows. IEEE Trans. on Multimedia, 15(3), 2013.
[3] Athanasios Noulas, Gwenn Englebienne, and Ben J.A. Kroese. Multimodal Speaker Diarization. IEEE Trans. on PAMI, 34(1), 2012.
[4] Einat Kidron, Yoav Y. Schechner, and Michael Elad. Pixels that sound. In CVPR, 2005.
[5] Jan Cech, Vojtech Franc, Jiri Matas. A 3D Approach to Facial Landmarks: Detection, Refinement, and Tracking. In Proc. ICPR, 2014.
[6] M. Uricar, V. Franc and V. Hlavac, Detector of Facial Landmarks Learned by the Structured Output SVM. In VISAPP 2012. http://cmp.felk.cvut.cz/~uricamic/flandmark/

Vedoucí: Ing. Jan Čech, Ph.D.

Platnost zadání: the summer semester 2015/2016

L.S.

Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 23. 12. 2014

Acknowledgement

I would like to express the greatest appreciation to my supervisor Jan Čech for his patience, a magnificent guidance throughout this thesis and the amount of time he spent on consultations.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

....
Date Signature

Abstract

Cílem této práce je vytvořit audio-vizuální detektor řeči, t.j. algoritmus, který automaticky identifikuje, zda osoba ve videozáznamu v danou chvíli mluví. Tato úloha je důležitá pro tzv. audio-vizuální diarizaci nebo audio-vizuální rozpoznávání a učení identit. Navržený detektor řeči má dvě fáze. V první fázi je řeč detekována pouze na základě vizuální informace. V druhé části jsou již detekované části videa testovány na synchronnost s audio signálem. Z lokalizovaných významných bodů na detekované tváři jsou extrahovány geometrické video příznaky. Mel-frequency cepstral coefficients jsou použity jako audio příznaky. Synchronnost audio a video příznaků je testována kanonickou korelační analýzou s fixními projekčními koeficienty. Tento algoritmus je schopen vizuálně detekovat řeč a spolehlivě ověřit synchronnost na dostatečně dlouhé části videozáznamu, podle experimentů alespoň 8 sekund.

Klíčová slova

Detektor řeči, Audio-vizuální synchronnost, MFCC, CCA

Abstract

The aim of this thesis is to create an audio-visual speech activity detector, i.e. an algorithm which automatically identifies whether a person in a video is speaking at a time. This task is important in audio-visual diarisation problem and in audio-visual cross modal identity recognition and learning. A two phase speech activity detector is proposed. First, the speech activity is detected in a video sequence based on the visual information only. Second, the parts detected in the first phase are tested on synchrony with the audio signal. Geometrical video features are extracted from facial landmarks that are localized in a region found by a face detector. Mel-Frequency Cepstral Coefficients are used as audio features. The synchrony of audio and video features is tested by Canonical Correlation Analysis with fixed projection coefficients. The algorithm is able to detect lip activity and reliably confirm the synchrony on a sufficiently long audio-video sequences, at least 8 seconds according to the experiments.

Keywords

Speech Activity Detection, Audio-Video Synchrony, MFCC, CCA

Contents

Abbreviations

AV	Audio video
MFCC	Mel Frequency Cepstral Coefficients
SAD	Speech Activity Detection
PLP	Perceptual Linear Prediction
HMM	Hidden Markov Model
CRF	Conditional Random Field
GMM	Gaussian Mixture Model
PCA	Principal Component Analysis
CoIA	Coinertia Analysis
DBN	Dynamic Bayesian Network
fHMM	factorial Hidden Markov Model
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
LBP	Local Binary Patterns
PR	Precision-Recall
CCA	Canonical Correlation Analysis

1 Introduction

Audio-visual speech activity detector is an algorithm which identifies whether a person visible in a video speaks at a time. There are typical situations that occur with respect to a captured scene and sound. First, only a single person is visible in a video. Then the task is to determine whether the person is a speaker. Second, several people are captured in the video and its respective audio signal contains a speech of one of them. In that case we should determine who is the speaker. Third, multiple persons are visible but we don't know if one of them is a speaker.

This task can be solved in two different manners, video-only or audio-video. We can say that a person is speaking when his/her lips are moving. It can be assumed that a person is not likely to perform any lip motion when not speaking, or that the lip motion is rather very small and slow (i.e. are showing only emotional expressions). However, this might be a false assumption when dealing with multimedia data. Typically in TV news, an anchor is speaking while a scene containing a different person is being showed on the screen. This person might had been speaking in the original scene but the original speech has been replaced by the anchor's comments. Since a person should be classified as a speaker only if the same person is seen and heard, it is necessary to check whether the lip motion is *synchronous* with the audio signal.

1.1 Motivation

In recent years, the use of media increased rapidly, and the virtual distances between places got shorter. There are tons of TV shows, series, films and documentaries produced every year, and it is common to meet other people through video calls and conferences. Sometimes, we just want to sort our media in a meaningful manner. Sometimes, selecting a part of the whole media file might be useful. The ability to automatically identify a speaker can become a useful tool in several different applications.

First of all, it can enhance liability of *voice detection* algorithms in noisy environments. It can be assumed that a voice signal is likely to be detected if a person's lips visible on a video are moving. Voice detection is a necessary preliminary step for speech recognition. This might be useful even when making a video call. In that case, we want to transmit only the signal relevant to the conversation in the highest quality, not a background noise. We can assume that the relevant signal to be transmitted is when a person visible on the camera is speaking. Therefore, even a distant voice signal of a person not visible in the video signal should be treated as a noise.

There are several audio-video recording types, such as video conferences and broadcasted news, where having a way how to extract a contribution of each speaker might become handy. *Diarization* and *video indexing* are exactly the algorithms needed for an easy orientation within a wast number of files and video hours. A diarization algorithm takes a video as an input and gives a timeline of who was speaking when as an output. It has to learn speaker models from the video automatically. This involves a speaker

recognition method based on both audio and visual identity, and further assignment of these two modalities together. A preliminary step for this is a reliable *speaker detector*, that can indicate monologue sections within a video sequence.

After all, once we know how to *detect synchrony* automatically, it might be useful to make shifted audio-visual signals synchronous, assuming one signal has been delayed with respect to the other. This can be particularly useful when aligning media that has been incorrectly received, corrupted by conversion or just asynchronously captured.

1.2 Problem statement

The goal of this thesis is to propose and test an audio-visual speech detector, an algorithm that indicates a video subsequences where a speaker is visible on the video by investigating both audio and video signals jointly. A lip activity detector will be employed first, as a lip motion is a necessary condition for speaking. In addition to that, the audio and video signals will be tested on synchrony. Only the subsequeces selected by the video lip activity detector and classified as synchronous will be considered as speech subsequences.

1.3 Thesis structure

The thesis is structured as follows: Section 2 provides an overview of the algorithms that have been published in the recent years. The cited publications briefly introduce problems of the speech activity detection based on both audio and visual signals, audio-visual synchrony and speaker identification. The Section 3 describes the proposed methods. The feature extraction from both audio and video signals, as well as their synchrony measure, is explained. The functionality tests of the proposed methods and their results are shown in Section 4. The results are discussed and conclusions are made in Section 5.

2 Related work

Humans combine audio and visual information in deciding what has been spoken, especially in noisy environments; *human speech perception is bimodal in nature.* The visual modality benefit to speech intelligibility in noise has been quantified as far back as in [1]. Cross-modal fusion of audio and visual stimuli in perceiving speech has been demonstrated by the McGurk effect [2]. For example, when the spoken sound "ga" is superimposed on the video of a person uttering "ba", most people perceive the speaker as uttering the sound "da". This illusory effect clearly demonstrates the importance of optical information in the process of speech perception. There are three key reasons why vision benefits human speech perception [3]: It helps speaker (audio source) localization, it contains speech segmental information that supplements the audio, and it provides complimentary information about the place of articulation.

Voice is produced by the vibration of the vocal cord and the configuration of the vocal tract that is composed of articulatory organs, including the nasal cavity, tongue and lips. Since some of these articulators are visible, there is an inherent relationship between the acoustic and visible speech. The basic unit that describes how speech conveys linguistic information is the *phoneme.* Approximately 42 phonemes in American English [4] and 36 in Czech [5] are generated by specific positions or movements of the vocal tract articulators, but only some of them are visible. Such visible units are called *visemes.* The number of visually distinguishable units, visems, is much smaller the number of audible units, phonems, therefore a viseme often corresponds to a set of phonemes. There is no universal agreement about the exact partitioning of phonemes into visemes, but some visemes are well-defined, such as the bilabial viseme consisting of phoneme set "p", "b", "m". A typical clustering into 13 visemes introduced in [6] is often used to conduct visual speech modeling experiments.

Speech consists of consonants and vowels. Vowels are voiced and can be considered as quasi periodic source of excitation, while consonant are unvoiced and are similar to random noise [7]. Grouping consonants and vowels form different phonemes and visemes. Some visually distinctive consonants are acoustically easily confusable and visa versa [3]. Information about the place of articulation can help disambiguate, for example, the pairs "p" - "k", "b" - "d" and "m"-"n". On the other hand, consonances "d" and "n" sound different but look the same as they are members of the same viseme class. There are only few consonants that require lips to close firmly, specifically "b", "p" and "m", and the lower lip to touch the upper teeth, "v" and "f". All vowels need mouth to be open when uttered.

2.1 Speech activity detection

Speech activity detection (SAD) is an important first step of speech processing algorithms, such as speech recognition and speaker recognition. It is usually understood as a process of identifying all segments containing speech in an audio signal. Several

types of speech activity detectors can be distinguished according to the data they work with: audio, video and audio-video.

2.1.1 Audio speech activity detection

There are many audio-domain SAD algorithms available. Audio SAD can be classified as either time-domain approaches or frequency-domain approaches. Implementation of time-domain algorithms is computationally simple but better quality of speech detection is usually obtained with the frequency-domain algorithms. SAD include both unsupervised systems that threshold against some value of an energy or voicing feature [8] and supervised systems which train a classifier with features such as Mel frequency cepstral coefficients (MFCCs) or perceptual linear prediction coefficients (PLPs). Classifiers such as support vector machines [9], gaussian mixture models [10], and multi-layer perceptrons [10] have been successfully used. Algorithms for structured prediction have also found success, including both hidden markov models (HMMs) [11] and conditional random fields (CRFs) [12].

However, the reliability of the above audio-domain SAD algorithms deteriorates significantly with the presence of highly non-stationary noise. The vision associated with the concurrent audio source, as already mentioned, contains complementary information about the sound which is not affected by acoustic noise, and therefore has the potential to improve audio-domain processing.

2.1.2 Video speech activity detection

The visual aspects of speech detection can be categorized into geometric and non-geometric analysis. Geometric analysis focuses on the shape of the mouth region and lip features. The non-geometric analysis makes use of transformation to other domains to represent certain aspects of the region of interest [13].

Exploiting the bimodal coherence of speech, a family of visual SAD algorithms has been developed, exhibiting advantages in adverse environments. The algorithm in [14] uses a single Gaussian kernel to model the silence periods and Gaussian mixture models (GMM) for speech, with principal component analysis (PCA) for visual feature representation. A dynamic lip parameter is defined in [15] to indicate lip motion, which is low-pass filtered and thresholded to classify an audio frame. HMMs with post-filtering are applied in [16], to model the dynamic changes of the motion field in the mouth region for silence detection. This HMM model is however trained only on the visual information from the silence periods, i.e. without using those from the active periods.

2.2 Audio-video synchrony

In the previous sections, it has been shown that it is possible to detect and understand information from an audio or video signals. Also, that their cross-modal analysis, assuming signals are synchronous, can help to seek information in an adverse environment. But are they really synchronous? Is the talking face on the video really the source of the audio signal?

In order to investigate synchrony, the source of an audio signal has to be located in a video first. It is necessary to pinpoint only pixels associated with audio sources and distinguish them from other moving sources. This problem can be general (anything can be a sound source), limited on a specific class or adapted on talking faces only.

2.2.1 Audio source localization

The problem of a source location that is unrestricted to a specific class of objects is investigated in [17]. They presented a robust approach for audio-visual dynamic localization, based on a single microphone. The solution had to overcome the fact that audio and visual data are inherently difficult to compare because of the huge dimensionality gap between these modalities. The proposed algorithm is based on canonical correlation analysis (CCA) with removed inherent ill-posedness by exploiting the typical spatial sparsity of audio-visual events.

An example of a class specific detection is a speaker localization in a large space where the speaker's face cannot be seen. In this case, the decision has to be based only on body motion or sound source location. An approach to audio-video speaker localization in a large unconstrained environment has been proposed in [18]. This method automatically detects the dominant speaker in a conversation with the idea that gesturing means speaking. The classification is based on CCA of MFCC features and optical flow associated with moving pixels regions. In order to increase preciseness, a triangulation of the information coming from the microphones is computed to estimate the position of the actual audio source.

Talking faces are usually not detected by investigating audio-video synchrony. A face detector is used instead and the synchrony is checked for each individual detected face.

2.2.2 Talking faces speech synchrony

Asynchrony of a talking face and an audio signal can occur in several cases. First, an audio or video signal can be delayed. Second, a narrated scene where someone speaking is being shown while a narrator is speaking. Third, a not talking face or a picture is visible but the audio signal consists of a speech. Each of these cases are likely to happen in different types of videos, therefore methods dealing with asynchrony detection are usually problem specific. Work [19] presents an empirical study to review definitions of audiovisual synchrony and examine their empirical behavior.

Audio-video latency is a common problem when broadcasting or streaming media, this delay is sometimes corrected by the end user device or application. All the other cases are synchronous from the technical point of view, however the speaker is not visible in the video. Speech and narrated scenes are usually being distinguished in news, sports or TV shows. Only small time windows can be considered for synchrony detection as the scene can be very dynamic. This problem is then naturally expanded to video indexing and diarization, discussed in section 2.3.1.

Liveness is a test that ensures that biometric cues are acquired from a live person who is actually present at the time of capture [20]. The liveness detection is used as impostor attack prevention in biometric systems based on speaker identification and

face recognition. The talking face modality provides richer opportunities for verification than any ordinary multimodal fusion does [21]. The signal contains not only voice and image but also a third source of information, the simultaneous dynamics of these features. Therefore, the liveness check is performed by measuring the degree of synchrony between the lips and the voice extracted from a video sequence. The systems are usually based on CCA [21, 22, 23] and coinertia analysis (CoIA)[21, 22, 23, 20]. Also, HMMs were used in [20].

2.3 Speaker identification

Speaker identification can be a very complex problem and its solution has to be almost exclusively tailored for a specific application. Usually, the existence of only one speaker at a time is assumed, however some papers are dealing with multiple active speakers [24]. Sometimes, we know what is being said, i.e. a video transcript is available. A high precision of automatic naming of characters in TV videos [25] was achieved by combining multiple sources of information, both visual and textual. Sometimes, we even don't know whether there is a speaker in a video.

Among the different methods that perform speaker detection, only a few are performing the fusion of both audio and video modalities. Some of them just select the active face among all detected faces based on the distance between the peak of audio cross correlation and the position of the detected faces in the azimuth domain [26, 27]. A few of the existing approaches perform the fusion directly at the feature level, which relies on explicit or implicit use of mutual information [28, 27, 29]. Most of them address the detection of active speaker among a few face candidates, where it is assumed that all the faces of speakers can be successfully detected by the video modality.

A good example of an application, where speaker detection has to be performed in real time, is a video conference (distributed meeting). A boosting-based multimodal speaker detection algorithm is proposed by [30]. They compute audio features from the output of sound source localization, place them in the same pool as the video features, and let the logistic AdaBoost algorithm select the best features. This speaker detector has been implemented in Microsoft RoundTable. A detection based on mouth motion only is deployed in [31].

2.3.1 Diarization and video indexing

Speaker *diarization* is the task of determining "who spoke when?" in an audio or video recording that contains an unknown amount of speech and also an unknown number of speakers . More formally this requires the unsupervised identification of each speaker within an audio stream and the intervals during which each speaker is active [32]. The application domains, from broadcast news, to lectures and meetings, vary greatly and pose different problems, such as having access to multiple microphones and multimodal information or overlapping speech. Clear examples of applications for speaker diarization algorithms include *speech and speaker indexing*, document content structuring, speaker recognition (in the presence of multiple or competing speakers), to help in speech-to-text transcription, speech translation est.

Speaker diarization of AV recordings usually deals with two major domains, broadcasted news (BN) and conference meetings. The algorithms has to be adapted according to the differences in the nature of the data. BN usually has better signal-to-noise ratio but the audio often contain music and applause. Also, people occur less frequently and the actions are less spontaneous. On the other hand, meetings are more dynamic with possibly overlapping speech. A multimodal approach to speaker diarization on TV talk-shows is proposed in [33]. Work [34] proposed a Dynamic Bayesian Network (DBN) framework that is an extension of a factorial Hidden Markov Model (fHMM) and models the people appearing in an audiovisual recording as multimodal entities that generate observations in the audio stream, the video stream, and the joint audio-visual space.

Indexing is trying to structure TV-content by person allowing a user to navigate through the sequences of the same person [35]. This structuration has to be done without any predefined dictionary of people. Most methods propose to index people independently by the audio and visual information, and associate the indexes to obtain the talking-face one [25, 36, 37, 38]. This approach combines clustering errors provided in each modality which is trying to be overcome in [35] .

3 Methods

The implementation of an audio-visual speech detector consists of several steps that have to deal with both audio and video stream. As not all pixels in video frames are important for speech detection, as well as not all audio frequencies, the dimensionality can be reduced by extracting only the important features. Features from audio and video streams are extracted separately and then analyzed together in order to check their synchrony.

The number of video frames is significantly smaller than the number of audio samples. However, we want to analyze frames and samples that correspond to each other. In this work, a subsequence of samples corresponding to a frame is considered as in *Figure 1*. Video features are extracted form each individual frame and audio features from the corresponding subsequence of audio samples, therefore there is exactly one audio and one video feature vector describing each frame. There is no audio samples overlap of two consecutive subsequences.

All extracted features need to be as independent on the speaker as possible. The used methods have to treat both female and male voices, as well as video features have to be invariant to the face size and head pose.

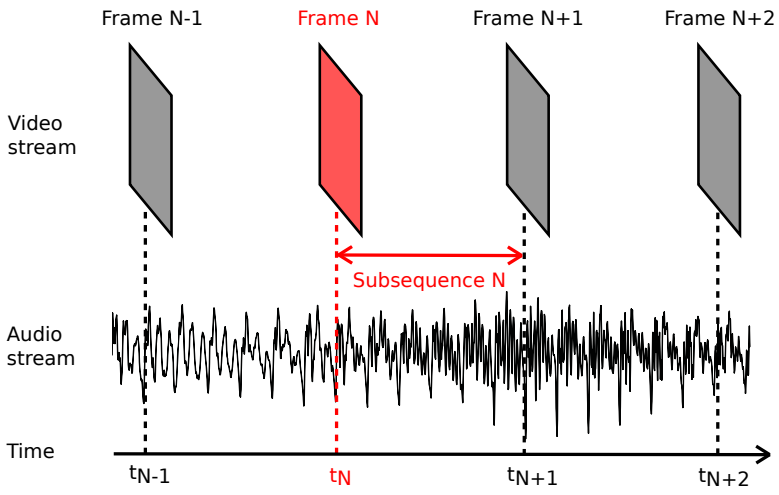


Figure 1 Audio subsequence corresponding to a frame N

3.1 Audio features

An acoustic speech signal contains a variety of information. It contains a message content as well as information from which the speaker speaker be identified. The most commonly used audio features for both speech and speaker recognition are Mel-frequency

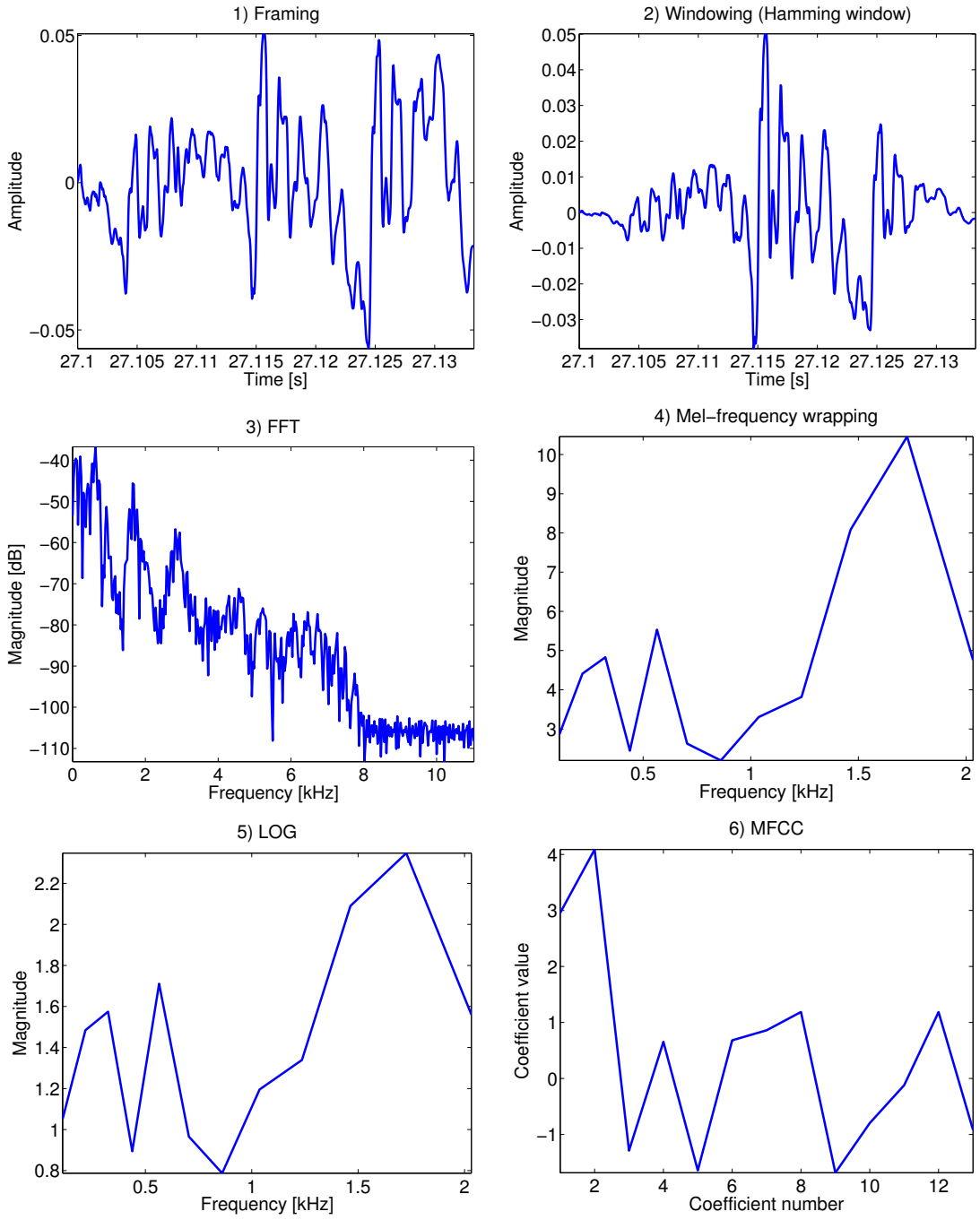


Figure 2 MFCC computation steps along with the results for each step of a single processed audio subsequence

Cepstral Coefficients (MFCCs), so they were chosen as audio features for this work.

Computing of MFCC features consists of six steps (as shown in *Figure 2*): framing, windowing, Fast Fourier Transform (FFT), Mel-frequency wrapping, logarithm and Discrete Cosine Transform (DCT).

The first step of MFCC computation is *framing*. A frame can be seen as the result of a speech waveform multiplied by a rectangular pulse whose width is equal to the frame

length. As we want to compute one audio feature vector in respect to each video frame, the frame length will be equal to the time difference between two consecutive video frames (i.e. 40ms for frame rate 25fps). *Windowing* by a rectangular shape window would introduce a significant high frequency noise at the beginning and end points of each frame, because of the sudden changes from zero to signal and from signal to zero. To reduce this edge effect, the Hamming window is used instead. The coefficients of a Hamming window **w** are computed from the following equation.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \tag{1}$$

The window length is N.

The next step is a conversion from the time domain to the frequency domain by computing *Fast Fourier Transform* (FFT). The FFT is a fast implementation of the Discrete Fourier Transform (DFT), shown in equation (2). The DFT of a vector **x** of length N, a vector of the audio signal values in a particular frame , multiplied by the windowing function vector, is another vector **X** of length N.

$$X(k) = \sum_{n=0}^{N-1} x(n)w(n)\omega_N^{kn}, \quad k = 0, 1, \ldots, N-1 \tag{2}$$

where

$$\omega_N^{kn} = e^{-j\left(\frac{2\pi}{N}\right)} \tag{3}$$

The procedure continues with *Mel-frequency warping*. A Mel is a unit based on the humanly perceived pitch difference between two particular frequencies. The Mel scale has approximately linear frequency spacing below 1000 Hz, and logarithmic spacing above [39]. Two times higher value in Mel indicates two times higher pitch (the pitch difference equal to one octave). The reference point between this scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 Mel to a 1000 Hz tone, 40 dB above the listener's threshold [40]. The conversion formula is expressed by the following equation.

$$mel(f) = 2595 \log\left(1 + \frac{f}{700}\right) \tag{4}$$

After the FFT block, the spectrum of each frame is filtered by a set of filters. The filter bank consists of 32 triangular shaped band-pass filters, whose centers are equally spaced in the Mel space, see *Figure 3*. This gives a vector **c** of 32 discrete values, but only the first thirteen, including the zeroth order coefficient, are further considered. Their logarithm is then transformed by Discrete Cosine Transform (DCT), as shown in the following equation,

$$Y(k) = a(k) \sum_{m=1}^{M} c(m) \cos\left(\frac{\pi}{2M}(2m-1)(k-1)\right), \quad k = 1, 2, \ldots, M \tag{5}$$

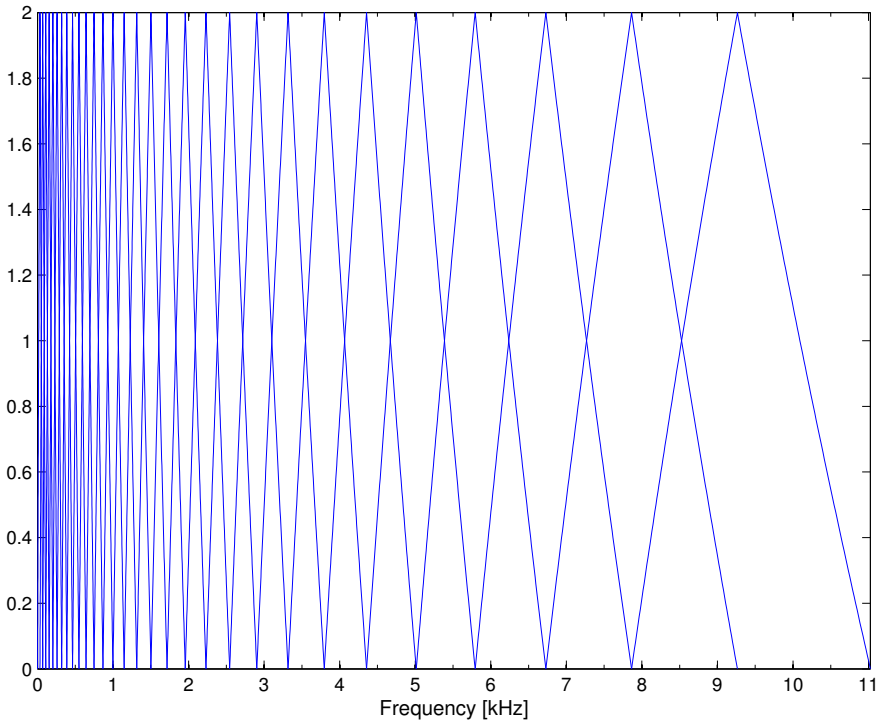


Figure 3 Triangular shaped band-pass filters in the frequency space

where

$$a(k) = \begin{cases} \frac{a}{\sqrt{M}}, & k = 1 \\ \sqrt{\frac{2}{M}}, & 2 \leq k \leq M \end{cases} \tag{6}$$

M is the number of cepstral coefficients (i.e. the length of the vector $\mathbf{c}$). The resulting MFCC features along with the original speech signal and its spectrogram can be seen in *Figure 4*.

The original MFCCs are used along with their delta (first order derivatives) and delta-delta (second order derivatives) values, as they describe the change between frames. In total, 39 audio features are extracted from a subsequence of audio samples corresponding to each video frame. In all experiments, *Voicebox* speech processing toolbox for Matlab [41] is used to extract MFCCs.

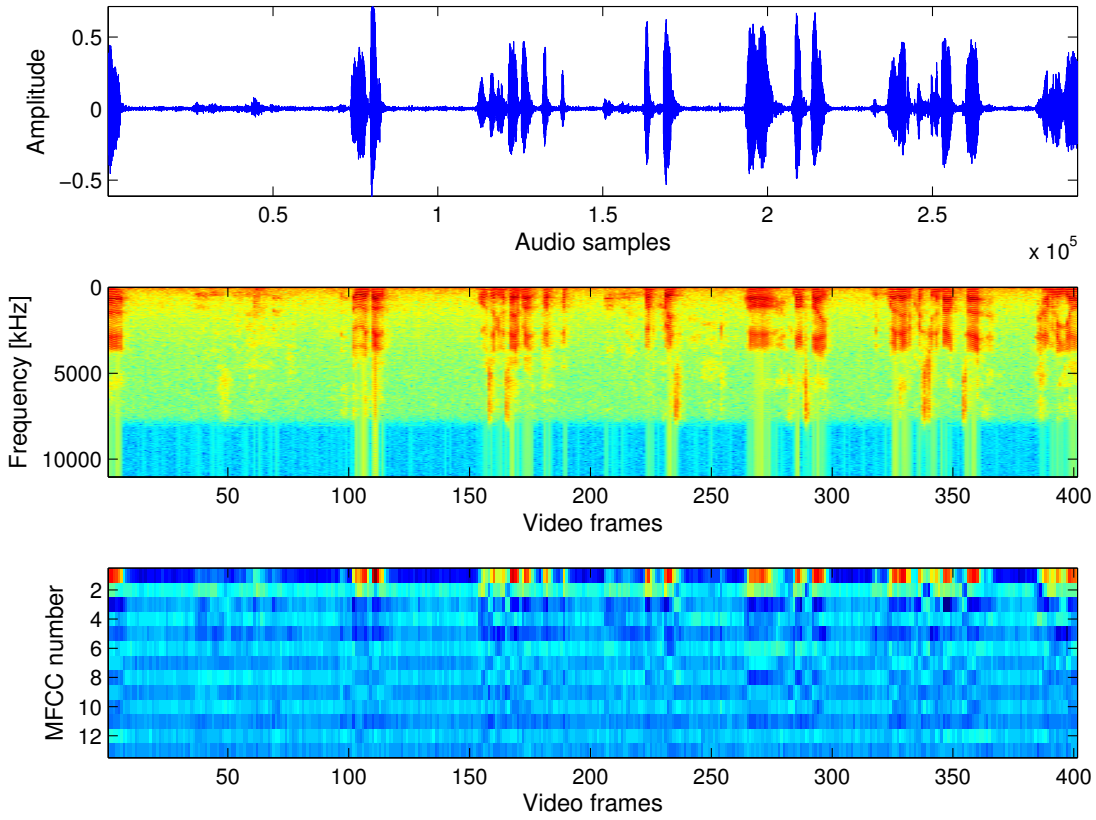


Figure 4 Speech signal and its corresponding spectrogram and MFCC features, resectively

3.2 Video features

Video features describe the lip motion of a potential speaker. In this work, geometrical lip features are used, therefore the features describe the shape of a mouth rather than raw intensity values in the mouth region. The extraction of such features consists of several steps, namely *face detection*, *facial landmarks localization* and *geometrical features extraction*. Also, the features have to be normalized so that they are invariant to the face size and head pose. However, we suppose that all analyzed speakers are facing the camera; no profile views are considered for simplicity.

3.2.1 Face detection and facial landmarks localization

The first step of extracting video features is a face detection. In this work, a reliable commercial face detector *Eyedea* [43], based on Waldboost [42], is used. This detector provides a bounding box for each face that appears on a frame and the detection confidence.

As the next step, facial landmarks are estimated on each of the detected faces. The *Chehra* detector [44] is used for facial landmark localization in all experiments. The method estimates landmark locations by finding parameters of a 3D shape model by means of regression. A facial model is initiated and then the incremental update of the cascade of linear regression is done by propagating the results from one level to the next. As learning the cascade of regression is by nature a Monte-Carlo procedure [45], every level is trained independently using only the statistics of the previous level.

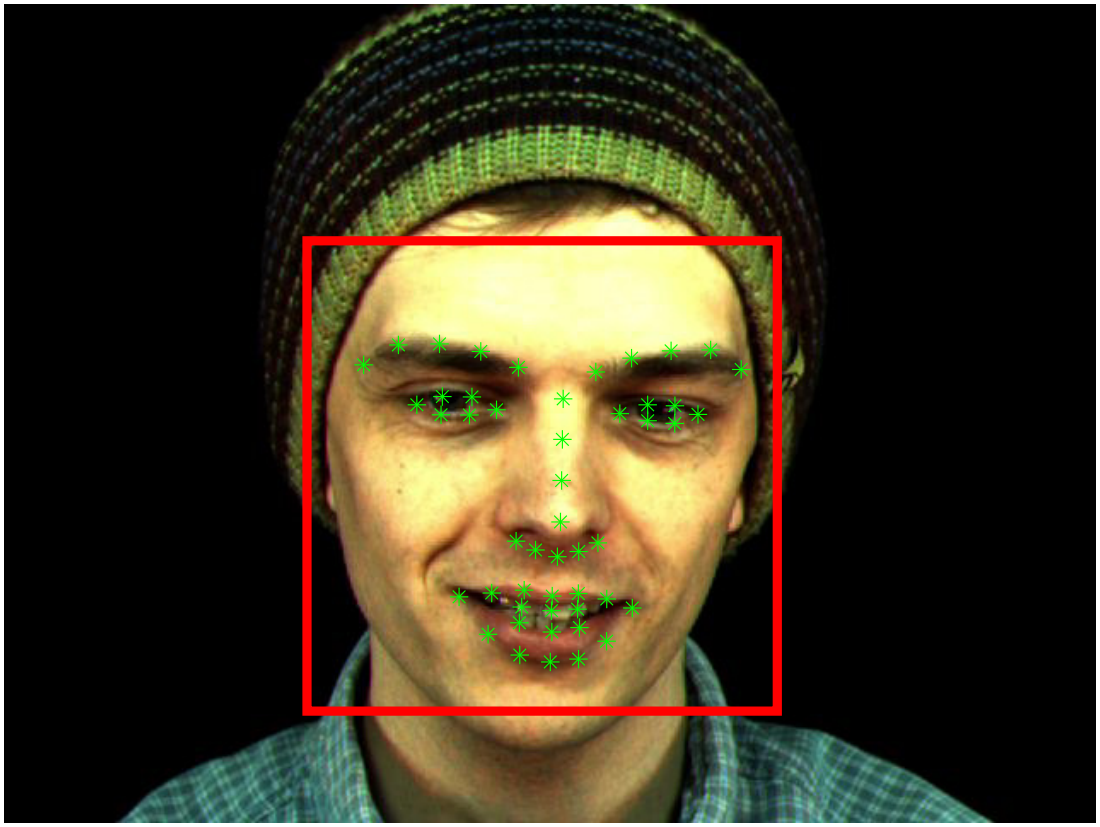


Figure 5 A result of the face detection (the red box) and facial landmarks localization (the green stars)

Finally the landmarks are found by projecting the model to the image.

The detection and facial landmark localization is done in each frame independently, no tracking is used. This is mainly due to the code availability, as the authors of *Chehra* provided Matlab codes just for landmarks localization in a single image, even though the tracking procedure is described in [44].

The result of the face detection is a bounding box and the facial landmark detector provides positions of 49 facial landmarks for each face in a video frame. An example of the detection and localization result is shown in *Figure 5*.

3.2.2 Feature extraction and normalization

The positions of facial landmarks cannot be used directly as video features, since their value depends on the face position and size in a particular frame. Neither coordinates within the face bounding box can be used, as the description would not be invariant to the head pose. Also, some of the landmarks are redundant for lip activity description.

First, the positions of facial landmarks are normalized. The normalization is done by a *homography* mapping from the estimated facial landmarks coordinates in a frame $\mathbf{x}_e$ to the corresponding initial facial model coordinates $\mathbf{x}_i$ used by the *Chehra* detector. The mapping is given by the equation $\mathbf{x}_i = \mathbf{H}\mathbf{x}_e$ [46], where points $\mathbf{x}_i$ and $\mathbf{x}_e$ are in homogeneous coordinates and $\mathbf{H}$ is a 3×3 matrix. The matrix H has to be estimated

from the facial points, preferably those that are not in the mouth region, as the lip landmarks are the main subject of normalization.

The equation $\mathbf{x}_i = \mathbf{H}\mathbf{x}_e$ may be expressed in terms of the vector cross product as $\mathbf{x}_i \times \mathbf{H}\mathbf{x}_e = \mathbf{0}$. If the j-th row of the matrix H is denoted by $\mathbf{h}^{j\top}$, than we may write

$$\mathbf{H}\mathbf{x}_e = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_e \\ \mathbf{h}^{2\top}\mathbf{x}_e \\ \mathbf{h}^{3\top}\mathbf{x}_e \end{pmatrix} \tag{7}$$

.
Writing $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$, the cross product may be given explicitly as

$$\mathbf{x}_i \times \mathbf{H}\mathbf{x}_e = \begin{pmatrix} x_i^2\mathbf{h}^{3\top}\mathbf{x}_e - x_i^3\mathbf{h}^{2\top}\mathbf{x}_e \\ x_i^3\mathbf{h}^{1\top}\mathbf{x}_e - x_i^1\mathbf{h}^{3\top}\mathbf{x}_e \\ x_i^1\mathbf{h}^{2\top}\mathbf{x}_e - x_i^2\mathbf{h}^{1\top}\mathbf{x}_e \end{pmatrix} \tag{8}$$

.
Since $\mathbf{h}^{j\top}\mathbf{x}_e = \mathbf{x}_e^\top\mathbf{h}^j$ for $j = 1, 2, 3$, this gives a set of three equations in the entries of $\mathbf{H}$, which may be written in the form

$$\begin{bmatrix} \mathbf{0}^\top & -x_i^3\mathbf{x}_e^\top & x_i^2\mathbf{x}_e^\top \\ x_i^3\mathbf{x}_e^\top & \mathbf{0}^\top & -x_i^1\mathbf{x}_e^\top \\ -x_i^2\mathbf{x}_e^\top & x_i^1\mathbf{x}_e^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{9}$$

.
These equations have the form of $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where $\mathbf{A}_i$ is a 3×9 matrix and $\mathbf{h}$ is a 9-vector made up of the entries of the matrix $\mathbf{H}$. As only two of the equations in 9 are linearly independent, it is usual to omit the third equation [46]. Then the set of equations becomes

$$\begin{bmatrix} \mathbf{0}^\top & -x_i^3\mathbf{x}_e^\top & x_i^2\mathbf{x}_e^\top \\ x_i^3\mathbf{x}_e^\top & \mathbf{0}^\top & -x_i^1\mathbf{x}_e^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{10}$$

.

Each point correspondence gives rise to two independent equations in the entries of $\mathbf{H}$. Given a set of four such point correspondences, we obtain a set of equations $\mathbf{A}\mathbf{h} = \mathbf{0}$, where $\mathbf{A}$ is the matrix of equation coefficients build from the matrix rows $\mathbf{A}_i$ contributed from each correspondence, and $\mathbf{h}$ is the vector of unknown entries of $\mathbf{H}$. We seek a non-zero solution $\mathbf{h}$. The matrix $\mathbf{A}$ is a 8×9 matrix of rank 8, and thus has a 1-dimensional null-space which provides a solution for $\mathbf{h}$.

The coefficients of a homography matrix $\mathbf{H}$ are computed from four point correspondences that are shown in *Figure 6.* These points were selected for these reasons: no or minimal movement of these points when speaking, no selected point is in the mouth region, no three points are on a single line. This gives suitable normalization of the facial landmarks, especially of the mouth region.

As not all facial landmarks are needed for speech detection, only those from the mouth region are further considered. Actually, as lips tend to have shape of an ellipse when uttering vowels, only two parameters describing the minor and major axes are needed. Therefore, only the distances between lips in the horizontal and vertical directions are measured. These two distances are computed from the normalized facial landmarks as

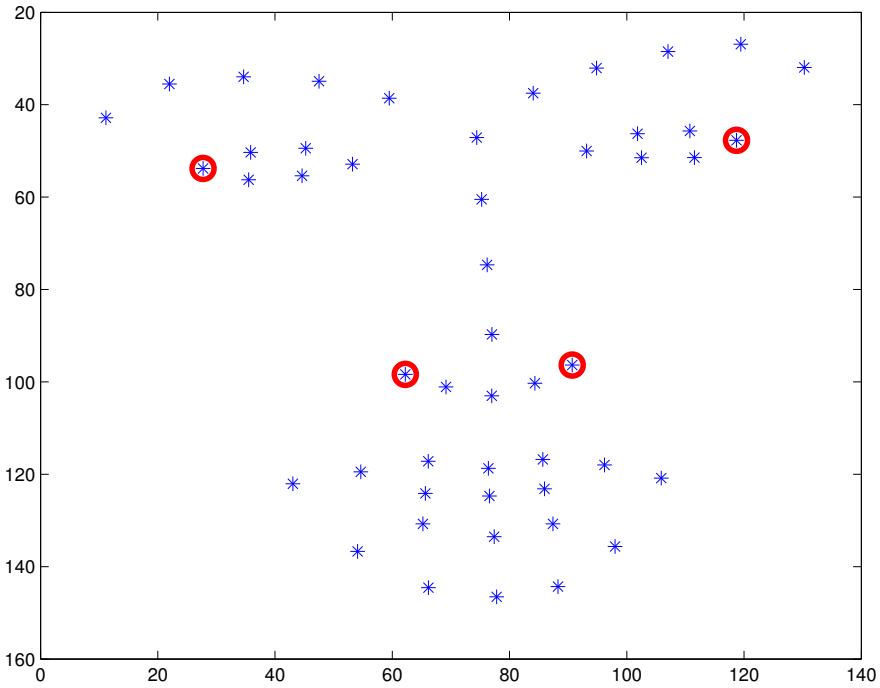


Figure 6 The four points used to estimate the homography mapping (red circles)

shown in *Figure 7*. The video descriptor for each face in a frame is a vector these two distances and their first and second derivatives, in total 6 features.

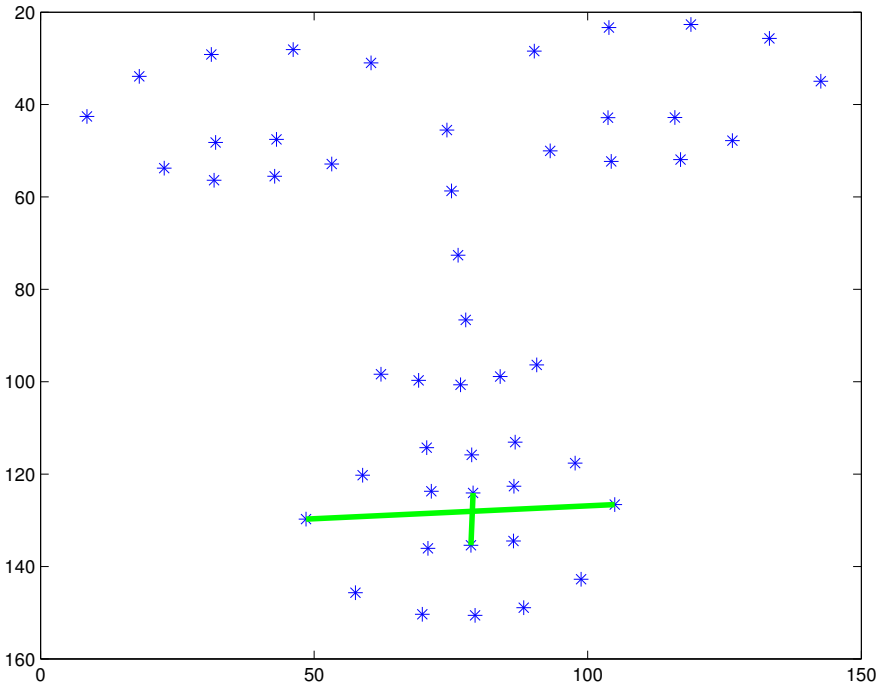


Figure 7 Two distances used as video features (green line segments)

3.3 Canonical correlation analysis

An important tool for understanding the relationship between audio and video data is canonical correlation analysis (CCA). The CCA is a statistical method to measure the relationship between two sets of multidimensional data [47]. We can use it to find the linear combinations of audio variables and video variables whose correlations are mutually maximized. In other words, it finds the best direction to combine all the audio and image data, projecting them onto a single axis.

Let $\mathbf{v}$ represent the video features and $\mathbf{a}$ be a vector of audio features corresponding to a single frame, as defined in sections 3.2 and 3.1 respectively. As the number of audio and video features is different, the length of the vectors $\mathbf{v}$ and $\mathbf{a}$ is also different. Both signals are considered as random vectors due to their temporal variations. Both vectors $\mathbf{v}$ and $\mathbf{a}$ are assumed to have a zero mean. Each of these vectors is projected onto a one dimensional subspace by coefficient vectors $\mathbf{w}_v$ and $\mathbf{w}_a$, respectively. The result of these projections is a pair of canonical variables $\mathbf{v}^\top \mathbf{w}_v$ and $\mathbf{a}^\top \mathbf{w}_v$. The correlation coefficients of these two variables defines the canonical correlation between $\mathbf{v}$ and $\mathbf{a}$ [48],

$$\rho = \frac{E[\mathbf{w}_v^\top \mathbf{v}\mathbf{a}^\top \mathbf{w}_a]}{\sqrt{E[\mathbf{w}_v^\top \mathbf{v}\mathbf{v}^\top \mathbf{w}_v] E[\mathbf{w}_a^\top \mathbf{a}\mathbf{a}^\top \mathbf{w}_a]}} = \frac{\mathbf{w}_v^\top \mathbf{C}_{va} \mathbf{w}_a}{\sqrt{\mathbf{w}_v^v \mathbf{C}_{vv} \mathbf{w}_v \mathbf{w}_a^\top \mathbf{C}_{aa} \mathbf{w}_a}}, \tag{11}$$

where E denotes the expectation and $\mathbf{C}$ is a covariance matrix. Specifically $\mathbf{C}_{vv}$ and $\mathbf{C}_{aa}$ are the covariance matrices of $\mathbf{v}$ and $\mathbf{a}$, respectively; $\mathbf{C}_{va}$ is the cross-covariance matrix of the two vectors.

Let N_v be the dimension of visual features, N_a the dimension of audio features and N_F the number of frames. Define a matrix $\mathbf{V} \in \mathcal{R}^{N_F \times N_v}$, where a row t contains the video features vector $\mathbf{v}^\top(t)$. In the same way, define a matrix $\mathbf{A} \in \mathcal{R}^{N_F \times N_a}$, where a row t contains the coefficients of the audio signal $\mathbf{a}^\top(t)$. The empirical canonical correlation from equation (11) becomes

$$\rho = \frac{\mathbf{w}_v^\top (\mathbf{V}^\top \mathbf{A}) \mathbf{w}_a}{\sqrt{\mathbf{w}_v^\top (\mathbf{V}^\top \mathbf{V}) \mathbf{w}_v \mathbf{w}_a^\top (\mathbf{A}^\top \mathbf{A}) \mathbf{w}_a}} \tag{12}$$

The CCA is defined as the maximum ρ over the coefficients $\mathbf{w}_a$ and $\mathbf{w}_v$,

$$\rho^* = \max_{\mathbf{w}_v, \mathbf{w}_a} \rho. \tag{13}$$

This problem is solved by the equivalent eigenvalue problem [48]:

$$\begin{aligned} \mathbf{C}_{vv}^{-1} \mathbf{C}_{va} \mathbf{C}_{aa}^{-1} \mathbf{C}_{av} \mathbf{w}_v &= \rho^2 \mathbf{w}_v \\ \mathbf{C}_{aa}^{-1} \mathbf{C}_{av} \mathbf{C}_{vv}^{-1} \mathbf{C}_{va} \mathbf{w}_a &= \rho^2 \mathbf{w}_a \end{aligned} \tag{14}$$

Maximizing the correlation is equivalent to finding the largest eigenvalue and its corresponding eigenvector. It is necessary to solve only one of these equations, as the solution of the other easily follows by using the solution of the first one.

The coefficients $\mathbf{w_a}$ and $\mathbf{w_v}$ are usually estimated for each couple of matrices $\mathbf{V}$ and $\mathbf{A}$ separately, i.e. the coefficients are different for each investigated audio-video subsequence. This is computationally inefficient as the eigenvalue problem in equation (14) has to be solved for each subsequence. Also, as the direction of the highest correlation is different every time, the discrimination between synchronous and asynchronous subsequences might not be very accurate. Depending on the length of the subsequence, the coefficient might be estimated so that the correlation is actually higher for asynchronous frames rather than synchronous. For these reasons, it is preferred to estimate coefficients on a longer training sequence and keep them fixed for test subsequences.

In this work, the coefficients $\mathbf{w_a}$ and $\mathbf{w_v}$ are estimated only once on a long audio video sequence and then used for computing ρ of each tested subsequence. The idea is to classify the subsequences with a high ρ value as synchronized and vice versa. In order to do so, an optimal threshold on ρ is found.

4 Experiments

The speaker is detected in two different ways, by considering the visual information only and by adding a synchrony test. First, a *video only lip activity detector*, which estimates speech parts of a video from visual features only, was tested. Then the *properties of the CCA* used as a statistical score of synchrony are investigated. At last, an audio-visual speech detector is proposed as an *application of the CCA* with the use of the video lip activity detector.

4.1 Video lip activity detection

The lip activity detection is based on a simple fact that the lips are moving when a person is speaking. The video features described in Chater 3.2 directly describe how much is a the mouth open at a time. The idea is to classify a part of a frame sequence as "speaking" only when the lips are closing and opening quickly. If a mouth is open for a long time or the motion is too slow, it is probably just an expression of an emotion, such as smile or surprise. *Figure 8* shows the measured vertical distance between lips (blue line); the horizontal distance depends on what is uttered, but it doesn't play a key role in the lip activity detection.

Sliding window filter As we refer to a quick change of the distance between lips, it is better to work with the first derivatives, that are shown in *Figure 8* (green line). To detect the speaking parts, two sliding window filters and thresholding were employed. The first sliding window filter computes the standard deviation of the first derivatives in a particular window. As thresholding of this result usually provides a lot of short "speaking" subsequences rather than less longer ones, the result is further filtered by a mean filter. The mean filter reduces the number of short time low value results, therefore we get fewer subsequences but of a longer duration. An example of the signal after filtering is shown in *Figure 9*.

Window size dependency The dependency of the result on sizes of the sliding windows has been tested. We thresholded the filtered signal by different thresholds; if the threshold is low, the rate of false positives is very high and vice versa. The overall error, the rate of false positives and false negatives, is decreasing with the window size. The test results can be seen in *Figure 10*. The result is better if the standard deviation filter window is bigger and can be improved even more by enlarging the mean filter window size. A larger window also means a higher threshold used for classification. Overall, the error does not depend on the window sizes significantly. The key aspect of windowing by a larger windows is to detect longer subsequences, rather than a lot of shorter ones, as it is easier to determine synchrony on more frames.

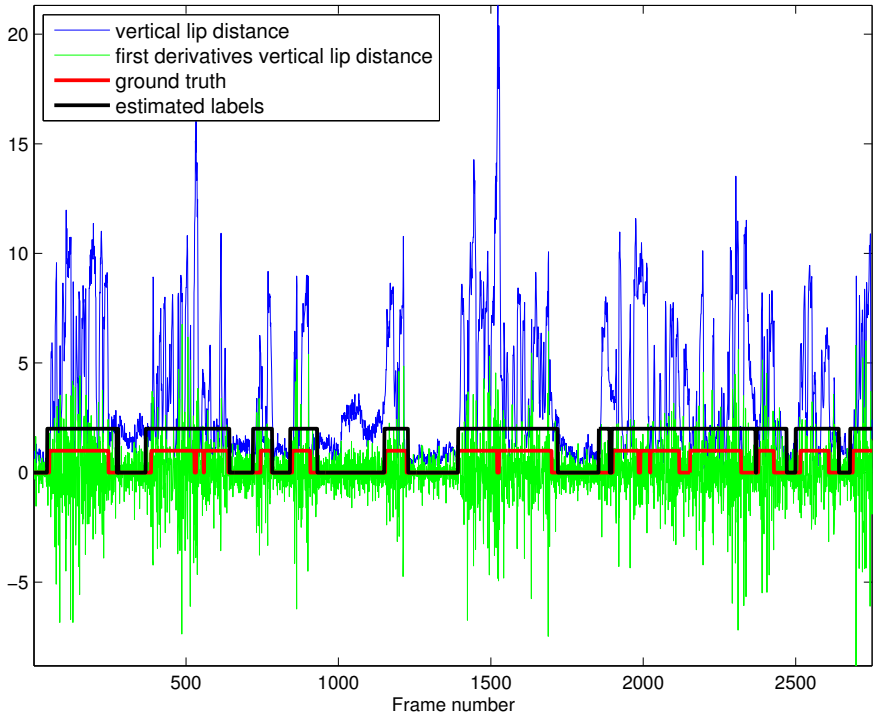


Figure 8 Original video features (vertical distance between lips and its firt derivatives) along with their original and estimated labels of the "speaking" parts

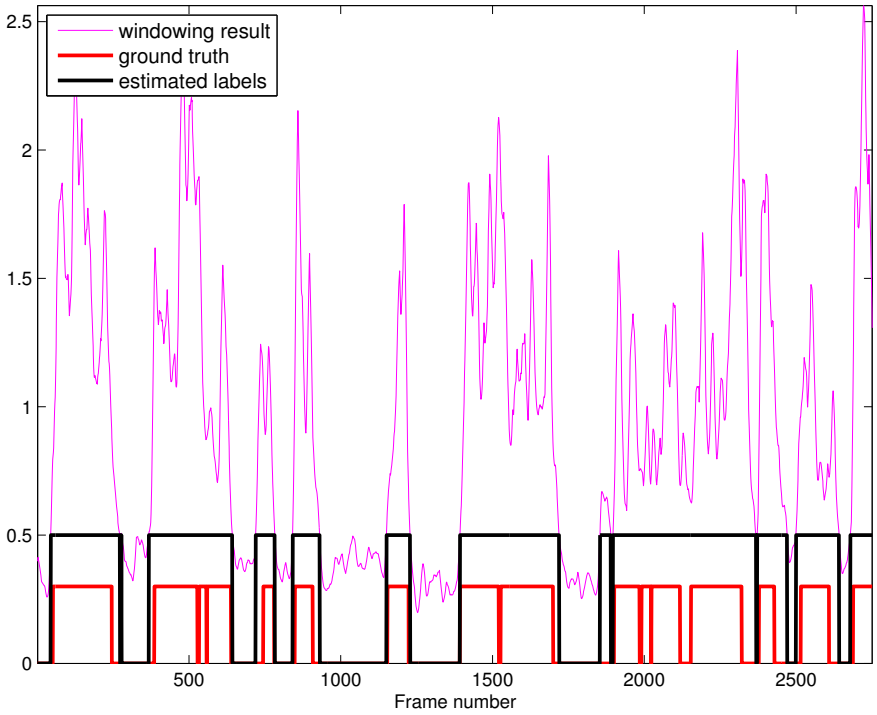


Figure 9 First derivatives of the vertical distance between lips filtered by std and mean sliding window filters along with their original and estimated labels of the "speaking" parts (the same video segment as in Figure 8)

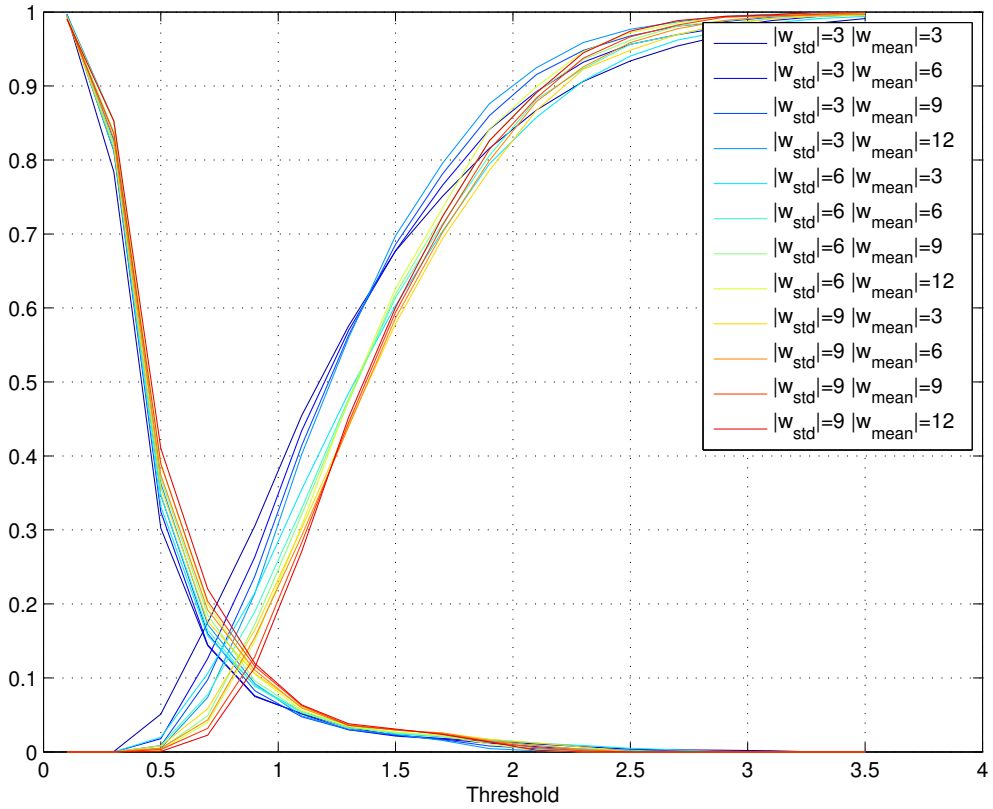


Figure 10 Dependency of false negative and false positive rate on the sliding windows sizes and threshold for the video lip activity detector($|w_{std}|$ and$|w_{mean|}$ are sizes of the std and mean window filters, respectively)

The lip activity detector was tested on the Cardiff Conversational Database [49], The achieved accuracy was 78.9%, using window sizes 5 for the standard deviation filter and 15 for the mean filter, and threshold 0.8, a detailed result can be seen in Table 1. The major source of inaccuracy was an incorrect localization of facial landmarks, which could not be corrected by the lip activity detector itself. Moreover, the method is very sensitive to the threshold setting as every person opens mouth in a different way and the landmark localization work with different accuracy for each person. The lip motion is a biometric marker for people identification [50]. However, not all speech parts can be detected correctly even if all the features were perfect. There are some typical motions that look like visemes, but actually do not come with any sound, such us breathing in with an open mouth before starting speaking.

4.2 Introspection of synchrony detection by the CCA

Canonical correlation analysis expresses the correlation between audio and video signals and is being used as a statistical score of synchrony in this work. Several test has been done in order to prove the capability of CCA to express the level of synchrony between audio and video signals. The dependency of the CCA result on the number of used frames has been investigated for both the training and testing. Also, a possibility of using more than one eigenvector in equation (14) is discussed.

The *training and testing* of the CCA was done on two independent video sequences.

Both videos contain a single speaker talking without pauses, the speaker is different in both videos. The CCA coefficients were estimated on a single video file, using the whole audio-video sequence. The testing was done on the second video using different number of frames in tested subsequences. The videos were captured by a conventional web camera at resolution 720x480 at frame rate 25 fps. The duration of the training and testing video is 139 and 63 seconds, respectively.

4.2.1 Coefficient estimation

The CCA coefficients are estimated from the audio and video features of a video sequence. The result vary according to the length of the used sequence. The coefficient estimated on a short sequences have dominant values, however the selection of these dominant elements vary a lot from sequence to sequence. It has been shown that for training coefficient values $\mathbf{w_a}$ and $\mathbf{w_v}$ converge to unique vectors with the increasing length of the used sequence. *Figure 11* shows the convergence of the estimated coefficients.

The expressiveness of the CCA was tested by sliding window of length $|w|$ that is equal to the number of frames in a tested video subsequence. This window was shifted one by one frame throughout the whole tested video sequence and the CCA result was computed on each of them. Taking a sequence of synchronous audio and video signals, the audio signal has been shifted by N frames with respect to the video signal in each window. This way, both synchronous and asynchronous signal samples are investigated. Such results for each window and audio shift are shown in correlation diagrams.

Correlation diagram The correlation diagrams show the capability of the CCA to express the level of synchrony. The correlation diagram is a 2-D matrix where each column corresponds to a video *frame number* and row to the *audio shift* by N frames. By a frame number we mean the center frame of a tested window. Audio shift value means that the audio signal was shifted by N frames with respect to the video signal. The CCA result for each window center frame and audio shift is expressed by the color at a particular position. The values were mapped to a colormap according to their value so that the higher values are warmer and the lower are colder colors.

The CCA coefficient were estimated on the entire training video. The quality of the statistical score of sychrony computed by using these estimated coefficients was first tested on the training video. A correlation diagram has been computed for a window size 200 frames, see *Figure 12*. This initial test shows that the difference between the synchronous and asynchronous tested windows is readily apparent. In all further experiments, we use this fixed projection.

4.2.2 Synchrony analysis

It has been shown that the CCA result is capable to distinguish synchronous and asynchronous sequences. However, the test in *Figure 12* was carried out on the same video as the CCA coefficients were estimated from. In order to prove that the CCA has the the capability to distinguish synchronous and asynchronous sequences in general, the CCA has been tested on an independent video. In the following tests, the same coefficients as in the Section 4.2.1 were used (coefficient vectors estimated on the entire

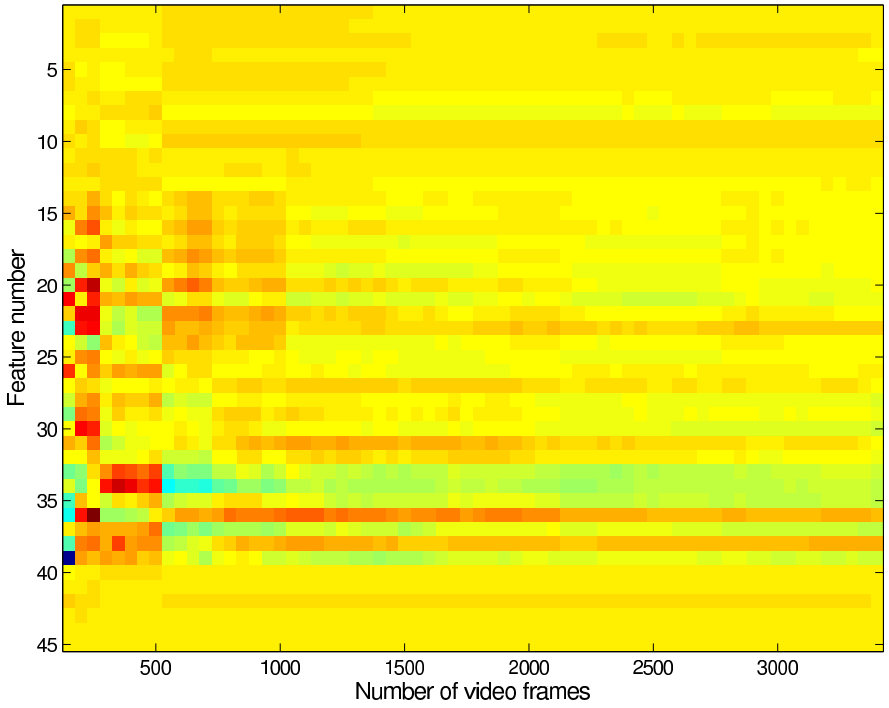


Figure 11 Convergence of the CCA coefficients with a longer audio-video sequence used for the estimation (each column contains concatenated vectors $\mathbf{w}_a$ and $\mathbf{w}_v$). The beginning of the sequence used for estimation is the same for all columns, only the last frame number changes in each column, therefore in each column a different number of frames was used for coefficient estimation.

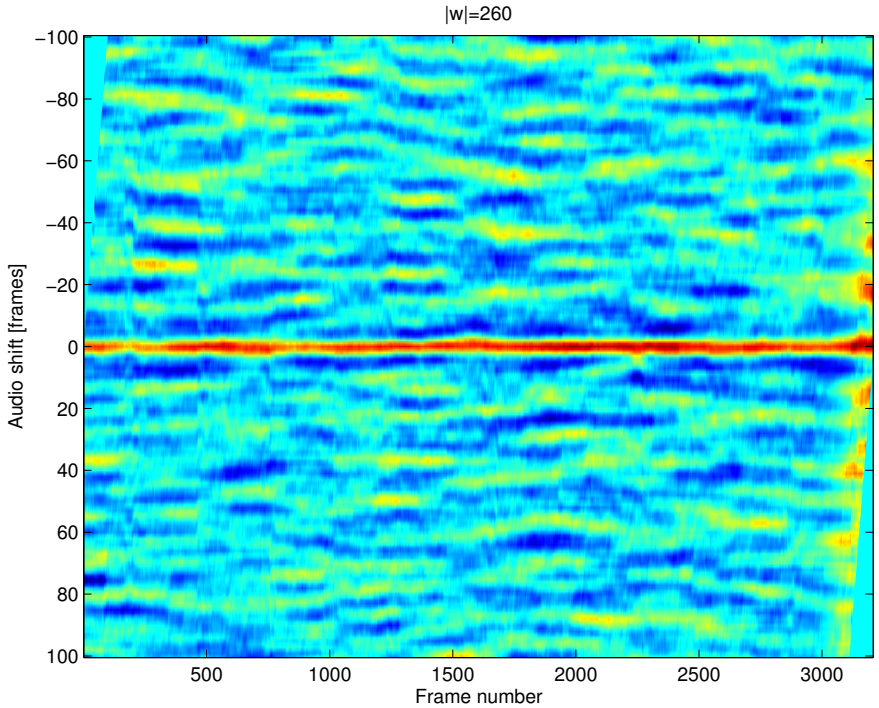


Figure 12 The dependency of the CCA result on an audio shift shown on a correlation diagram (testing and training on the same video). The CCA was computed for a sliding window of the length 260 frames. The distinct red line of synchronous sequences is nicely visible.

training video), but a different video with a different speaker and a different text was used for testing.

Window size dependency The correlation diagrams were computed for different window sizes. The large windows provide an expected highly discriminating results, however the small windows suffer from aperture problem. For small windows, a locally high correlation are observed even for asynchronous windows. The dependency can be seen in *Figure 13*. A clearly visible difference between synchronous windows and windows with the sound track shifted by N frames can be seen for window size $|w|$ longer than 200. Window size between 100 and 200 provides a high value for synchronous sequences, however some asynchronous sequences show high values of correlation as well. This phenomena is even more noticeable for $|w|$ smaller than 100. This is probably caused by a local similarity of the signal. A vertical cross-section through the diagram in *Figure 13* for frame number 1200 can be seen in *Figure 14*. The values for small window are rather oscillating, however an obvious peak expressing the synchrony is visible for windows bigger than 100 frames. The reliability of the synchrony check by the CCA highly depends on the window size.

Classification The CCA provides a single value for the whole AV subsequence. As the resulting value increases with a level of synchrony, it is intuitive to use thresholding for the classification. In order to find the best threshold, we constructed histograms of values for synchronous and asynchronous subsequences. The *histograms* are computed from the diagrams shown in *Figure 13* and can be seen in *Figure 15*. The histogram of synchronous samples is computed from the diagram row of audio shift 0; all windows except for those between audio shifts -5 and 5 were used to computed histograms of the asynchronous samples. The values close to the audio shift 0 were eliminated as the audio shift is too small to consider them as asynchronous, however they are not entirely synchronous. As expected, the error (histogram overlap) decreases with the window size. The mean of asynchronous subsequences remains at zero for all window sizes, but their standard deviation is decreasing with larger windows. We can observe the same phenomenon for synchronous subsequences. The values for a window as small as ten frames are almost indistinguishable. However, the discriminability grows with a longer window size. This means that it is possible to classify an AV subsequece as synchronous or asynchronous by thresholding if the used window is large enough.

Projections to other directions The number of CCA coefficient vectors is equal to the minimum number of features of both audio and video feature vectors. As we use 6 video features and 39 audio features, the number of CCA coefficient vectors is 6. The linear combination of the first coefficient vector, an eigenvector with the highest corresponding eigenvalue, and the feature vector gives projection to the directions with the highest correlation. Therefore, the first coefficient vector is the most significant for the synchrony evaluation. As all the coefficient vectors, the eigenvectors in equation (14), are orthogonal, their projection provides an independent results that can be used either separately or combined together. Correlation diagrams for each individual coefficient vector computed by a sliding window of size 250 frames can be seen in *Figure 16*. The result of the first coefficient vector gives an already expected result. However, the second shows that the CCA values are higher for more subsequences that are synchronous. The other four does not seem to have a significant meaning while used

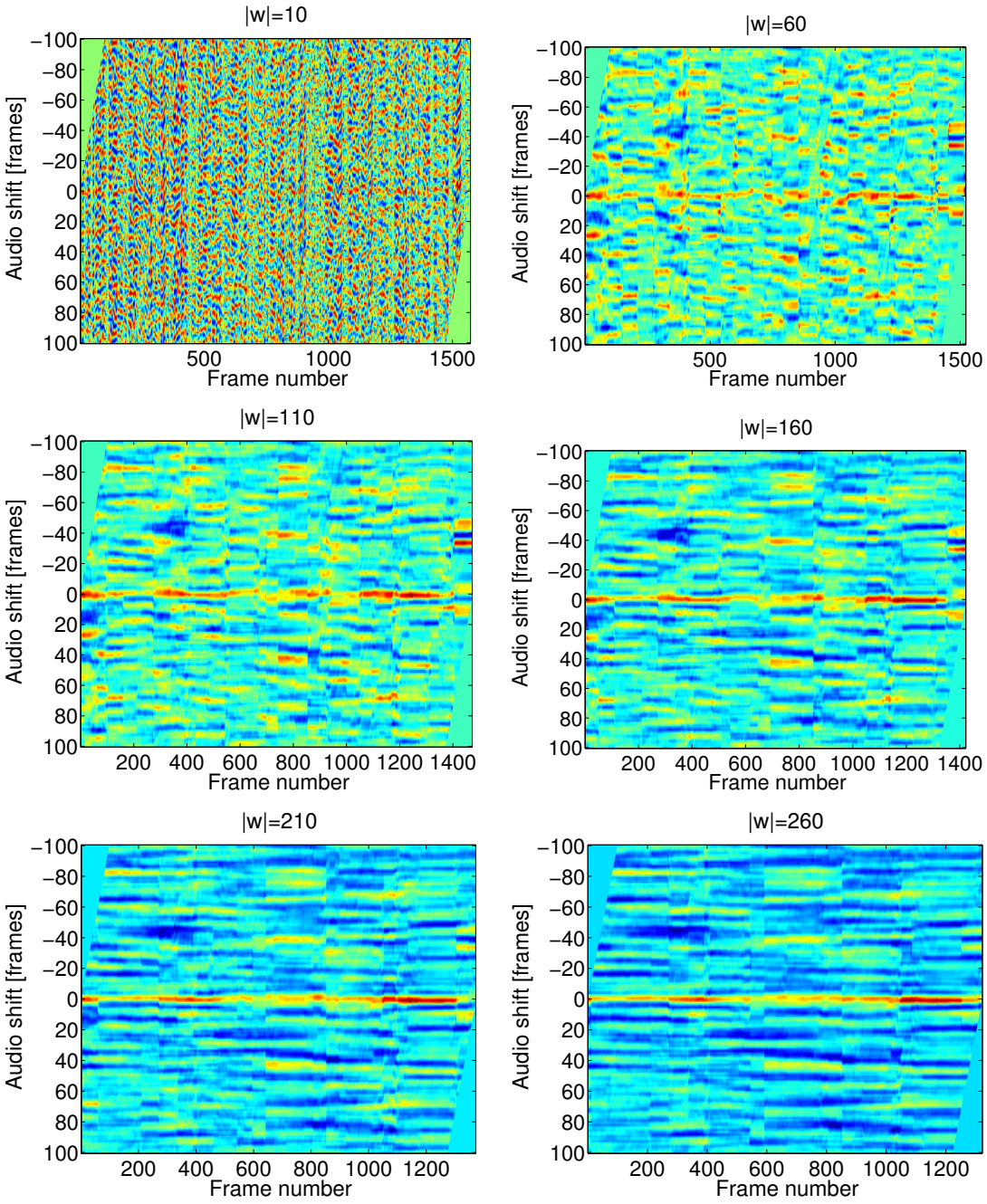


Figure 13 Correlation diagrams computed with a different length of the sliding window $|w|$ (the number of frames that are used for computation). The result of the CCA is very dependent on the number of frames used for the computation

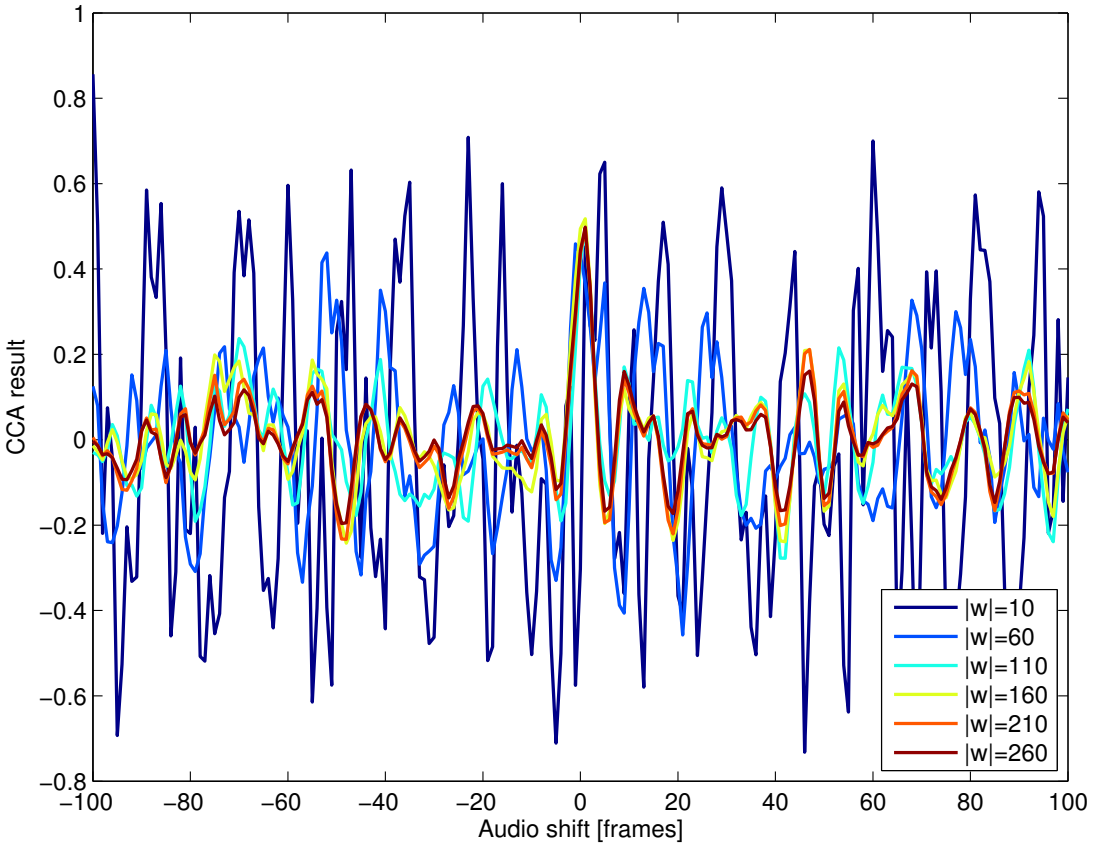


Figure 14 Cross-section (one column) of the diagrams in Figure 13 (window centered at the frame number 1200). A distinct mode of synchronized subsequences when CCA was computed on a large window, oscillations otherwise.

alone, however their combination may have a better result than the first coefficient only.

Averaging projections The idea is to average results of multiple coefficient vectors in order to increase values for synchronous frames and vice versa. Since the contributions of the coefficient vectors are not correlated, their combination might filter out random fluctuation of high correlation for small windows. The resulting diagrams of averaging are in *Figure 17*. The first diagram is a result of the first coefficient vector only, the second is an average of the first two, the third is an average of the first three etc. The number of high values is increasing for windows in the row of synchronous frames, as we wanted. However, the values belonging to asynchronous frames are increasing as well. This might not be an improvement for the final classification.

Error The histograms shown in *Figure 15* are computed from the first CCA coefficient vector only. It might be interesting to see whether the averaging of results from multiple coefficient vectors can decrease the error. The error is estimated from the histograms by integrating their overlap. The intersection of the histogram curves defines the optimal threshold. The error is then half of the sum of incorrectly classified subsequences. A graph that shows a dependency of error on the window size for multiple averaging results is shown in *Figure 18*. The slope of curves representing errors of averaged results

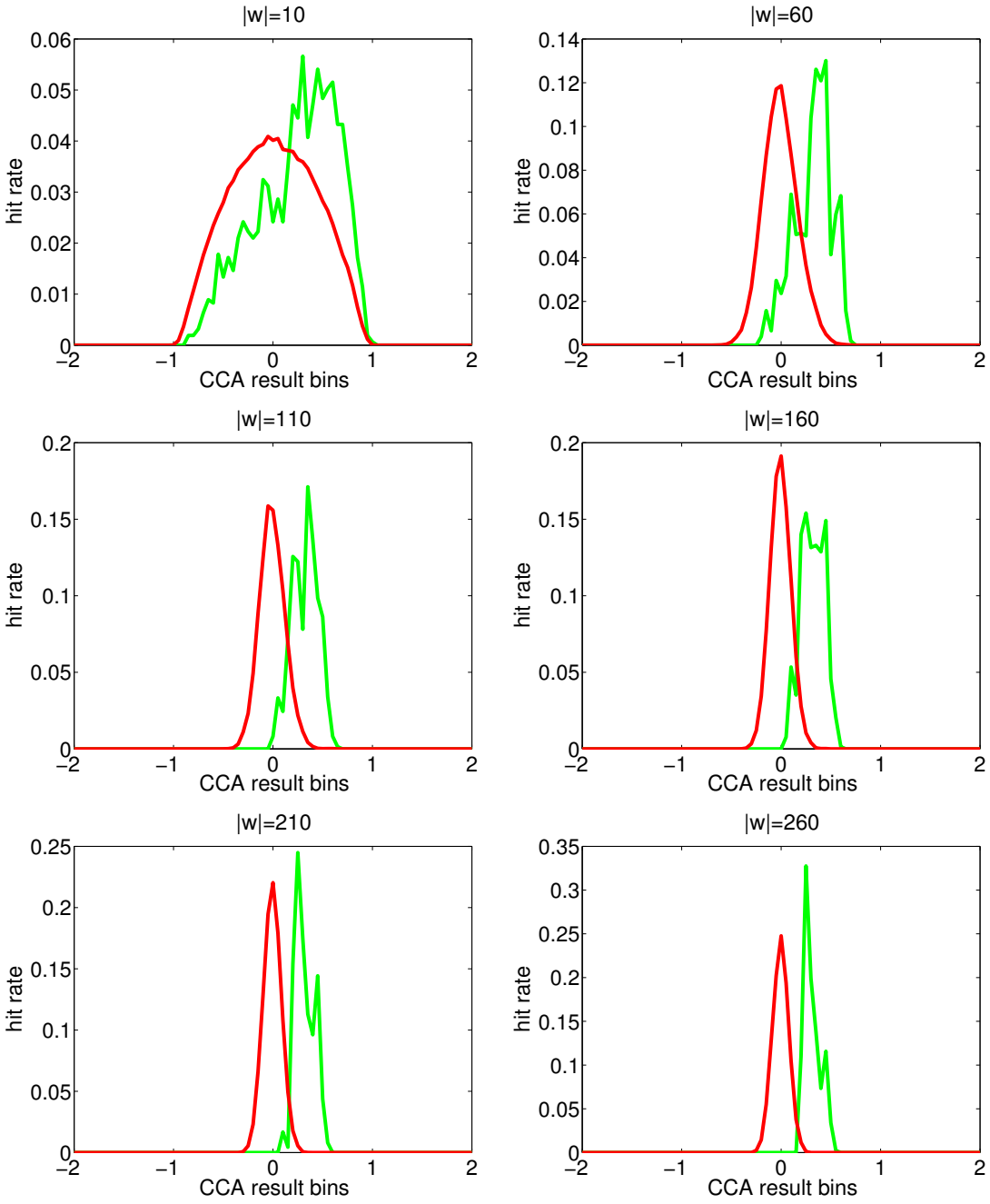


Figure 15 Histograms of the CCA results for asynchronous (red curve) and synchronous (green curve) video subsequences computed from the diagrams shown in *Figure 13*. The histograms are shown for an increasing window size.

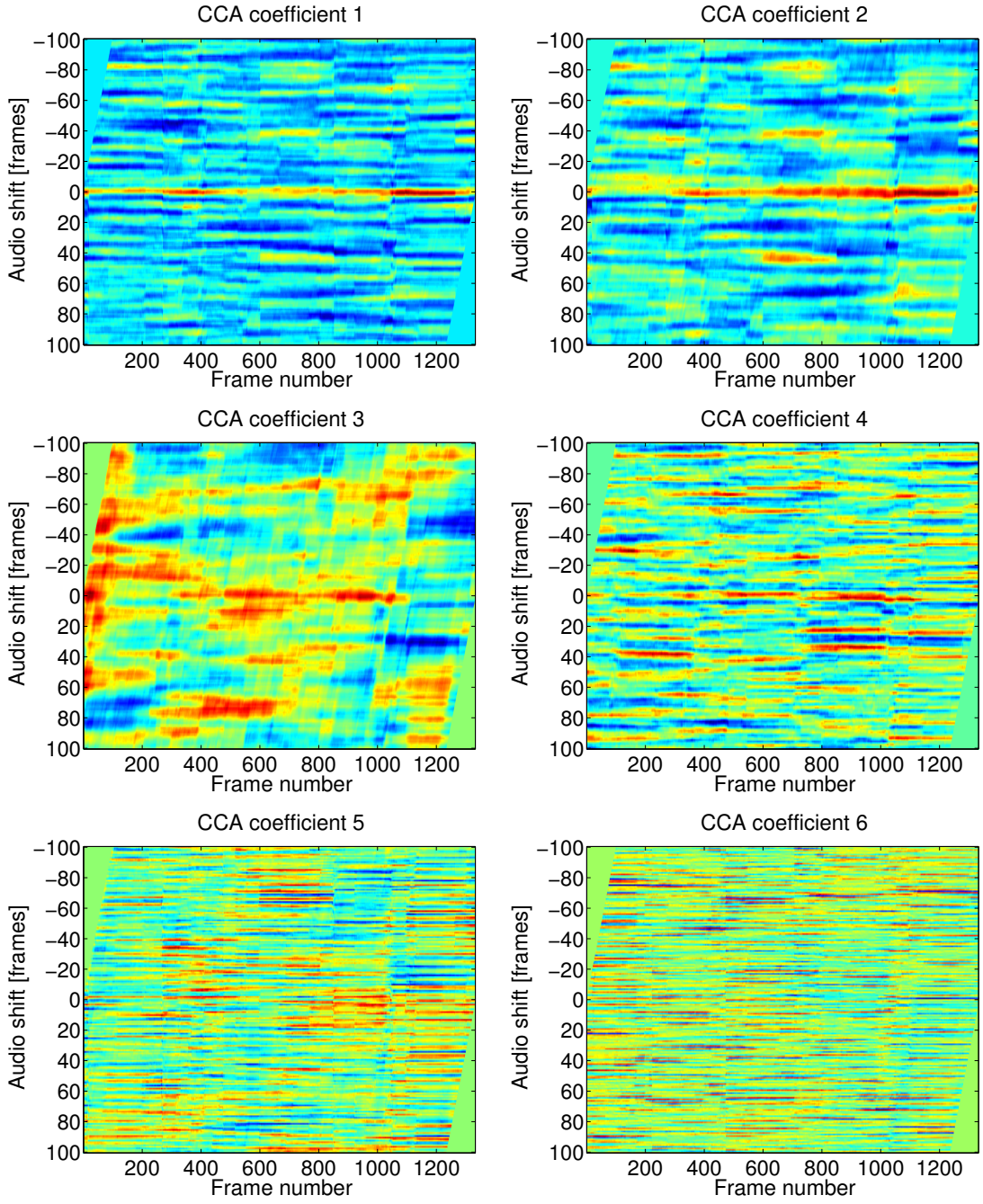


Figure 16 Correlation diagram computed by a fixed size window. Each of the correlation diagrams is a projection by a different CCA coefficient vector. The diagrams are sorted by the eigenvalue corresponding to each of the coefficient vectors in the decreasing order.

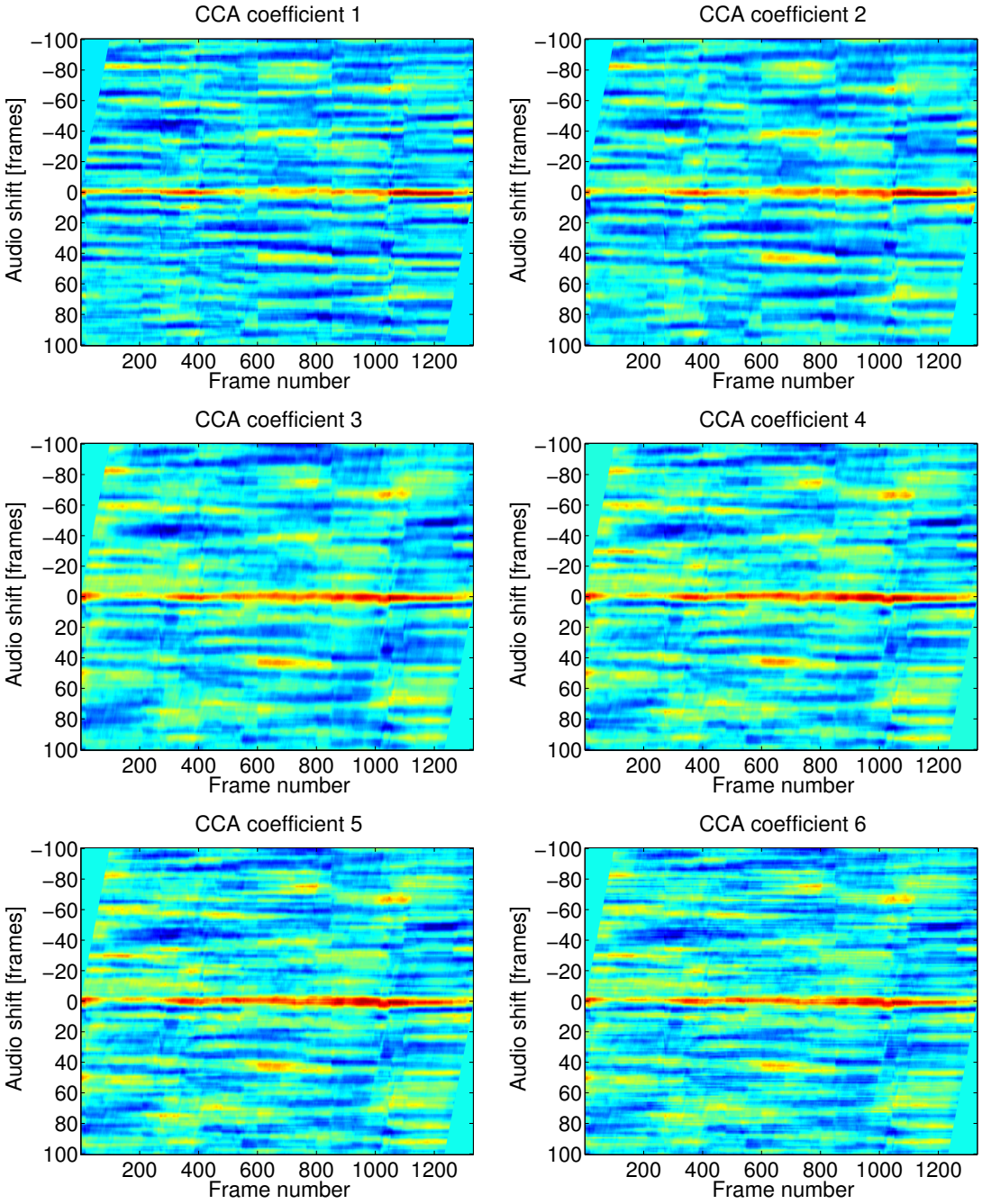


Figure 17 Correlation diagrams computed by averaging the results from each CCA coefficient vector (i.e. average of the diagrams in *Figure 16*). The first diagram is the result of the first coefficient vector only, the second is the average of the first and second one, the third of the first three est.

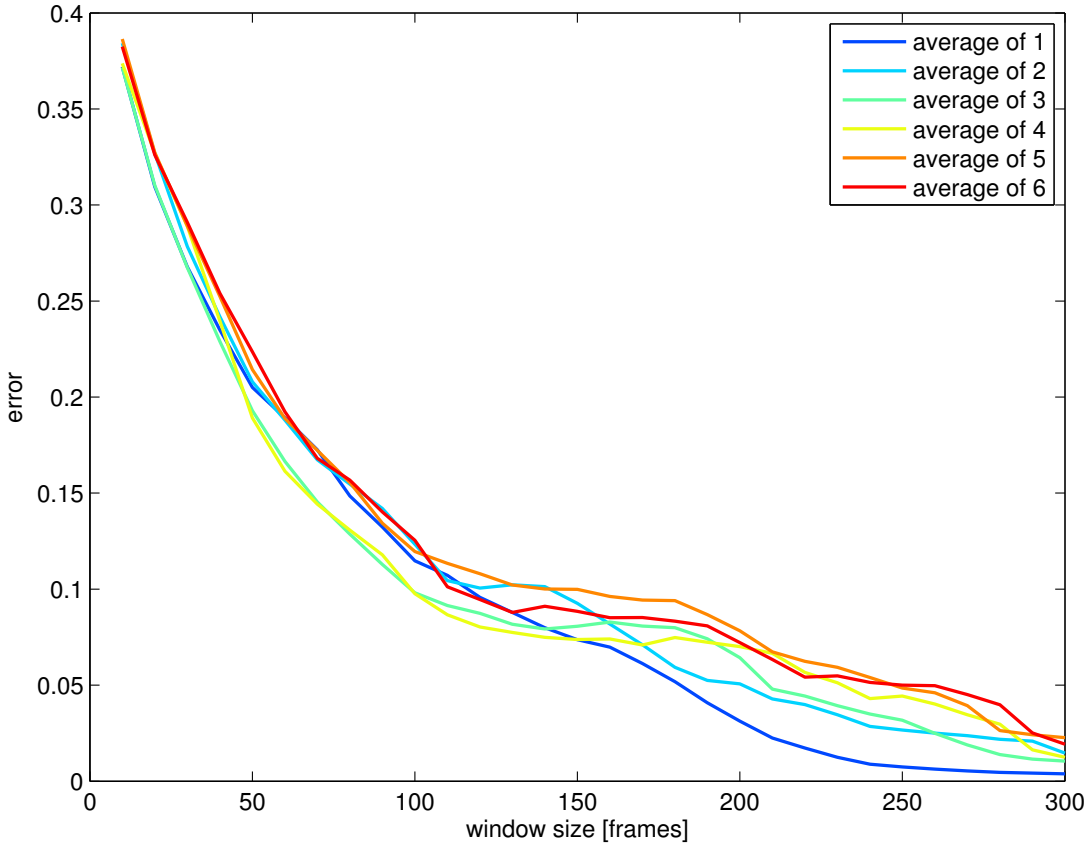


Figure 18 Error (computed form the histogram overlaps) dependency on the window size and averaging of the results contributed by individual coefficient vectors

from the first three and four coefficient vectors is decreasing more steeply than of the others. Although, this phenomenon changes at window size 150 frames. The averages from the first five and six coefficient vectors seems to have the worst result. The best result, especially for large windows, provides the first coefficient vector without any averaging. Its error is as low as 1% for a window of size 300 frames.

Precision-recall curves As the last test, we investigated precision-recall (P-R) curves. Precision is a fraction of all positive classifications that are true positive; recall is a fraction of all originally positive instances that were classified as positive. The precision and recall is measured for an increasing threshold, which spans a P-R curve. The best curve is the one which has the biggest area under the curve. The dependency of the P-R curve on the window size can be seen in *Figure 19*. It clearly shows that a longer window improves the P-R curve. *Figure 20* shows the effect of averaging is not very significant. However, the P-R curves of the averaged results indicate a slightly better performance for certain threshold settings.

4.3 Applications of the CCA

The CCA can be employed as a synchrony test. It has been used to improve the video only lip detector by testing the synchrony. Additionally, the CCA result for

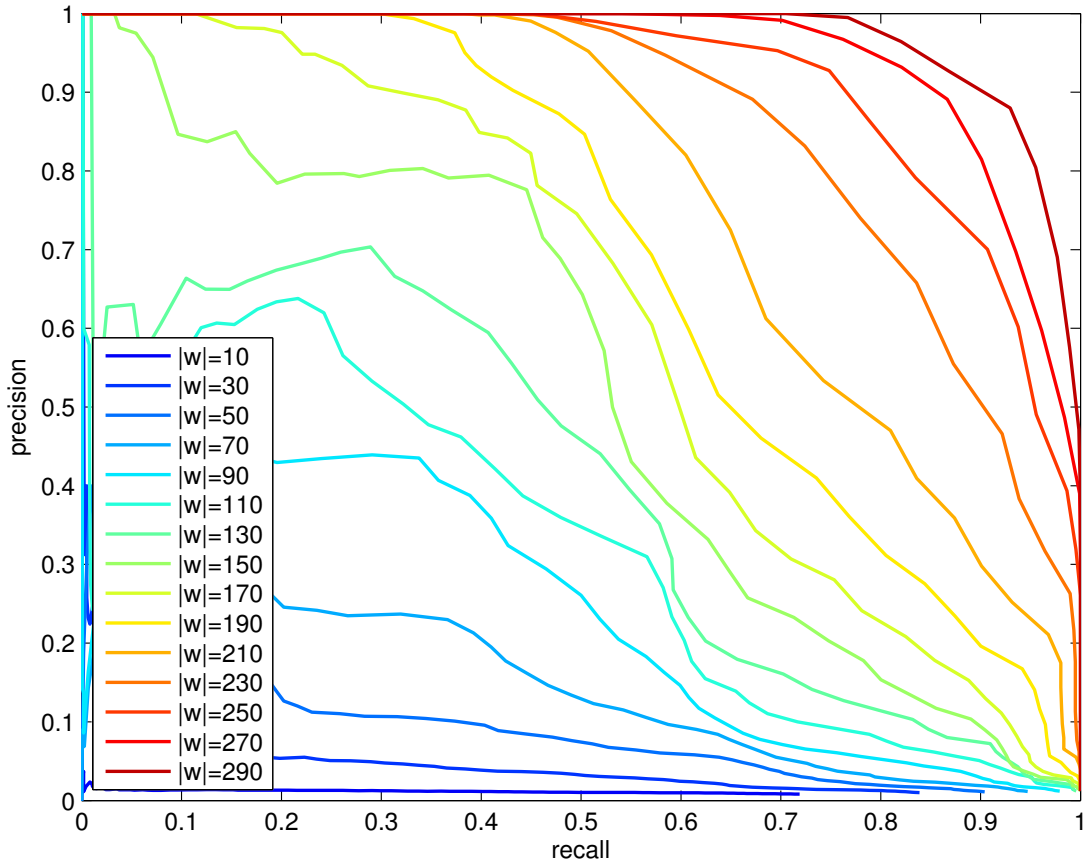


Figure 19 Precision-Recall curve of the CCA results for different window sizes on which the CCA is computed (only the first coefficient vector is used)

shifted windows provides a tool that can estimate an unknown delay between audio and video signals in case of a corrupted sequence.

4.3.1 Detection of synchronized video segments

The speech can be estimated from the lip motion only, supposing that the speaker is always speaking when his/her lips are moving. However, in some cases, the lip motion may not always indicate speech. There are some situations when lips are moving and nothing is uttered, such as breathing in or smiling, or the uttered sound is not necessarily a speech, i.e. emotional expressions. Also, the speaker might be different from the person that is visible on the video, however this case is rather rare.

It has been shown that the synchrony can be determined on a large windows with a high accuracy. However, to test a large window in an arbitrary video might not be always reliable. Almost all videos that require a speech detection has some level of dynamics. A change of speakers within a tested video sequence would introduce boundary artifacts. A part of the tested sequence would contain a voice, face or both of a different speaker and the synchrony test would not be reliable.

In order to overcome these problems, we propose a *two phase audio-visual lip activity detector*. In the first phase, the potential speech parts are detected by the *video lip activity detector* described in 4.1. This eliminates the boundary artifacts, i.e. it

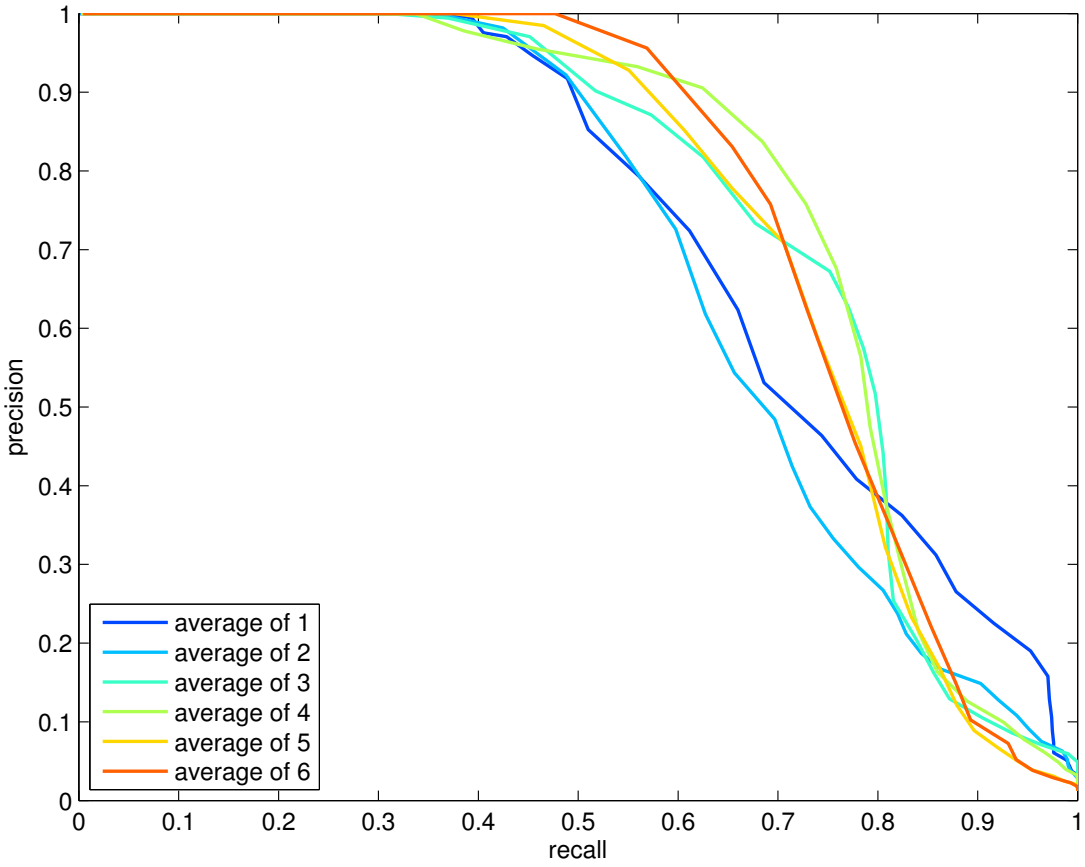


Figure 20 Precision-Recall curve of the averaging CCA results for window size $|w| = 200$. Each curve was obtained by averaging the results of individual coefficient vectors

ensures that the tested parts really contain only the face of one potential speaker. The preliminary speech part detection also provides subsequences as long as possible, so the window for testing the synchrony is long enough. The second phase is the *synchrony test* by computing the CCA. The CCA is computed on each of the subsequences selected by the lip activity detector. The subsequence represented by the CCA result is then classified by thresholding.

Cardiff Conversational Database The proposed audio-visual speech activity detector was tested on the Cardiff Conversational Database [49]. The database contains of 16 videos that represent 8 conversations. Each speaker was captured separately on a single video, however the two videos contain a single conversation. Therefore, we tested the videos one by one, but the two videos belonging to one conversation were tested with audio signal computed by summing the two audio signals together. This way, each video contained both synchronous and asynchronous parts.

First, the video only lip activity detector with threshold 0.8, standard deviation filter window size 5 and mean filter window size 15, was used to detect the speech parts. Each part was then analyzed for synchrony. The CCA was computed on each subsequence selected by the lip activity detector and classified as synchronous if the CCA result value was higher than 0.2. The classification error was then computed as the sum of all incorrectly classified frames over the number of frames in the tested video. The CCA

video name	video only	audio-video	CCA on ground-truth segments
P1_P2_1402_C1	0.2680	0.1816	0.1150
P1_P2_1402_C2	0.2114	0.2383	0.1603
P1_P3_1502_C1	0.2806	0.2801	0.1032
P1_P3_1502_C2	0.1979	0.1920	0.0928
P3_P4_1502_C1	0.1208	0.2271	0.0518
P3_P4_1502_C2	0.1683	0.1941	0.0819
P5_P2_1003_C1	0.1355	0.1511	0.0144
P5_P2_1003_C2	0.4654	0.3797	0.0933
P5_P3_2202_C1	0.2220	0.2537	0.0732
P5_P3_2202_C2	0.2196	0.2217	0.0590
P6_P2_1602_C1	0.1644	0.2054	0.0324
P6_P2_1602_C2	0.2408	0.3088	0.0214
P6_P3_1602_C1	0.1401	0.3253	0.3234
P6_P3_1602_C2	0.2394	0.2361	0.0322
P6_P4_1602_C1	0.1432	0.2248	0.0946
P6_P4_1602_C2	0.1624	0.1948	0.0455
average	0.2079	0.1920	0.0871

Table 1 Classification error of the video only lip activity detector (video only), two phase audio-video speech detection, i.e. the result of video only lip activity detector tested on synchrony by CCA (audio-video) and CCA classification of an the ground-truth segments (CCA on ground truth). The video names are constructed as follows: ID of speaker 1, ID of speaker 2, conversation number, video number of the conversation

projection coefficients were fixed as in all previous tests. The classification results can be seen in *Table 1.*

The overall classification error is lower for the audio-visual detection than the video detection only. However, the results of the speech detection are very dependent on the tested videos. The classification error of the CCA estimated on the ground truth subsequences, i.e. the ground-truth subsequences of speech classified as asynchronous, is lower.

There are several sources of errors that can significantly influence the result accuracy. First, the CCA coefficients are trained on a single video that is in a different language (trained on female voice in Czech, tested on male voices in Welsh English). Second, due to the lighting conditions of the dataset videos, the *Chehra* detector is very inaccurate. Third, all the steps are very sensitive to the threshold selection. Especially the threshold of the video only lip detector significantly influence the result of the synchrony check. A high threshold can result in a large number of short windows, rather than smaller amount of long windows, and the CCA synchrony test is very inaccurate for very small windows. Selection of the optimal parameters is highly dependent on the speaker and capturing conditions.

4.3.2 Estimation of audio delay

The CCA synchrony test provides a tool to estimate an audio shift for a video signal with latency. Let $v(t)$ be the video signal and $a(t+\tau)$ be the audio signal with unknown

delay τ. The delay is found by

$$\tau^* = \arg\max_{\tau} \mathrm{CCA}\big(v(t), a(t+\tau)\big), \tag{15}$$

that is implemented by shifting the audio signal back and forth. The frame difference between the original position and the maximum of the CCA results is the desired audio shift.

The shift estimation was tested on all videos previously used in this work. It has been tested on 10 seconds long subsequences. The video alignment finds the exact delay in 99.1% subsequences. The error is caused by estimating the maximum somewhere very close to the correct shift, but not exactly to the desired one.

5 Conclusion and future work

An audio-visual speech activity detector has been proposed. The algorithm has two phases, the *video only lip activity detection* and *audio-video synchrony test*. The visual lip activity is detected from the first derivatives of the vertical distance between lips by applying two sliding window filters, the standard deviation and the mean, of different sizes and thresholding. The synchrony is checked by CCA on the video subsequences pre-segmented by the visual lip activity detector. The CCA had fixed coefficients estimated on a single long video sequence. The CCA result is thresholded in order to classify subsequences as synchronous or asynchronous. Only the parts of the video segmented by the visual lip activity detector and confirmed by the CCA result thresholding as synchronous are labeled as speech part of the tested videos.

The *video only lip activity detection* gives good results if the optimal threshold is used. Unfortunately, the threshold varies a lot from speaker to speaker and it is hard to find a threshold that can be used generally. The way a person opens the mouth is a biometrical marker. The performance of the visual lip activity detector is also limited by the quality of the video features. The *Chehra* landmark localization is not always reliable, and is especially sensitive to illumination. This fact significantly influenced the results of testing on the Cardiff Conversational Database.

The *synchrony test* was done by the CCA. The CCA is a standard tool for a cross-modal analysis. It projects feature vectors into a one-dimensional space in the direction of the highest correlation. The CCA coefficients are usually estimated on each tested subsequence separately, however we have shown that they converge to a unique vector when estimated on a single long sequence. Therefore, a fixed projection was used. The CCA is very accurate for long tested sequences. The local high correlation fluctuation is apparent for short subsequences and the test result is rather unreliable. We can accurately distinguish between synchronous and asynchronous audio-video sequences if a sufficient number of frames is used. The reliable result is for a sequence 8 seconds long. As a side effect, the audio shift can be estimated for videos when one of the audio or video signals was delayed.

The proposed solution uses a simple classification based on a projection to a one-dimensional space (a linear combination), regardless of the speech content, and has fixed thresholds that does not accommodate to a particular speaker. Speech signals are not stationary. Certain phonemes clearly maps to corresponding visemes; however, the recogntion of the synchrony is probably different for each viseme and phoneme. For certain speech audio-visual segments, the synchrony recognition is intrinsically ambiguous. As a future work, we would like to investigate a machine learning based solution. A solution that would learn from data what audio features are important for particular audio features automatically. We aim the decision on the synchrony be possible from a shorter sequence of several frames only. This way, we would probably not be limited by a long subsequences and could directly classify video parts without a preliminary segmentation, which might significantly decrease the classification error.

Bibliography

[1] W.H. Sumby and I. Pollack. “Visual contribution to speech intelligibility in noise”. In: *Journal of the Acoustical Society of America* 26(2) (1954), pp. 212–215.

[2] H. McGurk and J. MacDonald. “Hearing lips and seeing voices”. In: *Nature* 264 (1976), pp. 746–748.

[3] A.Q. Summerfield. “Some preliminaries to a comprehensive account of audio-visual speech perception”. In: *Dodd, B., Campbell, R. (Eds.), Hearing by Eye: The Psychology of Lip-Reading. London, United Kingdom: Lawrence Erlbaum Associates* (1987), pp. 3–51.

[4] J.R. Deller, J.G. Proakis, and J.H.L. Hansen. *Discrete-Time Processing of Speech Signals*. Englewood Cliffs, New Jersey, USA: Macmillan Publishing Company, 1993.

[5] M. Krčmová. *Fonetika a fonologie*. URL: `http://is.muni.cz/do/rect/el/estud/ff/js08/fonetika/ucebnice/index.html` (visited on 12/17/2014).

[6] C. Neti et al. “Audio-Visual Speech Recognition”. In: *Workshop Final Report* (2000).

[7] M. Goyani, N. Dave, and N.M. Patel. “Performance Analysis of Lip Synchronization Using LPC, MFCC and PLP Speech Parameters”. In: *International Conference on Computational Intelligence and Communication Networks* (2010), pp. 582–587.

[8] S. Sadjadi and J. Hansen. “Unsupervised speech activity detection using voicing measures and perceptual spectral flux”. In: *IEEE Signal Processing Letters* 20 (2013), pp. 197–200.

[9] N. Mesgarani, M. Slaney, and S.A. Shamma. “Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14(3) (2006), pp. 920–930.

[10] Tim Ng et al. “Developing a speech activity detection system for the DARPA RATS program”. In: *Interspeech* (2012).

[11] T. Pfau, D.P. Ellis, and A. Stolcke. “Multispeaker Speech Activity Detection for the ICSI Meeting Recorder”. In: *Automatic Speech Recognition and Understanding* (2001), pp. 107–110.

[12] A. Saito et al. “Voice activity detection based on conditional random fields using multiple features”. In: *Interspeech* (2010), pp. 2086–2089.

[13] K.C. van Bree. “Design and realisation of an audiovisual speech activity detector”. Masters thesis. Eindhoven University of Technology, 2006.

[14] Peng Liu and Zuoying Wang. “Voice activity detection using visual information”. In: *International Conference on Acoustics, Speech, and Signal Processing* 1 (2004), pp. 609–612.

[15] D. Sodoyer et al. “A study of lip movements during spontaneous dialog and its application to voice activity detection”. In: *Journal of The Acoustical Society of America* 125(2) (2009), pp. 1184–1196.

[16] A. Aubrey, Y. Hicks, and J. Chambers. “Visual voice activity detection with optical flow”. In: *IET Image Process* 4(6) (2010), pp. 463–472.

[17] E. Kidron, Y.Y. Schechner, and M. Elad. “Pixels that Sound”. In: *IEEE Computer Vision and Pattern Recognition* 1 (2005), pp. 88–95.

[18] E. D’Arca, N.M. Robertson, and J. Hopgood. “Look Who’s Talking”. In: *Intelligent Signal Processing Conference* (2013), pp. 1–6.

[19] H.J. Nock, G. Iyengar, and C. Neti. “Speaker Localisation Using Audio-Visual Synchrony: An Empirical Study”. In: *Lecture Notes in Computer Science* 2728 (2003), pp. 565–570.

[20] E.A. Rúa et al. “Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models”. In: *Pattern Analysis and Applications* 12(3) (2009), pp. 271–284.

[21] H. Bredin and G. Chollet. “Audio-Visual Speech Synchrony Measure for Talking-Face Identity Verification”. In: *International Conference on Acoustics, Speech, and Signal Processing* 2 (2007), pp. 233–236.

[22] N. Eveno and L. Besacier. “A speaker independent "liveness" test for audio-visual biometrics”. In: *Interspeech* (2005), pp. 3081–3084.

[23] Zheng-Yu Zhu et al. “Liveness detection using time drift between lip movement and voice”. In: *International Conference on Machine Learning and Cybernetics* 2 (2013), pp. 973–978.

[24] Zhao Li et al. “Multiple active speaker localization based on audio-visual fusion in two stages”. In: *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems* (2012), pp. 1–7.

[25] M. Everingham, J. Sivic, and A. Zisserman. ““Hello! My name is... Buffy” – Automatic Naming of Characters in TV Video”. In: *British Machine Vision Conference* (2006), pp. 92.1–92.10.

[26] F. Talantzis, A. Pnevmatikakis, and A.G. Constantinides. “Audio-visual active speaker tracking in cluttered indoors environments”. In: *IEEE Transaction on Systems, Man, and Cybernetics* (2009), pp. 799–807.

[27] E.A. Lehmann and A.M. Johansson. “Particle filter with integrated voice activity detection for acoustic source tracking”. In: *Eurasip Journal on Advances in Signal Processing* (2007).

[28] P. Besson et al. “Extraction of audio features specific to speech production for multimodal speaker detection”. In: *IEEE Transaction on Multimedia* 10(1) (2008), pp. 63–73.

[29] M. Beal, H. Attias, and N. Jojic. “Audio-visual sensor fusion with probabilistic graphical models”. In: *European Conference on Computer Vision* 1 (2002), pp. 736–750.

[30] Cha Zhang et al. “Boosting-Based Multimodal Speaker Detection for Distributed Meeting Videos”. In: *IEEE Transactions on Multimedia* 10(8) (2008), pp. 1541–1552.

[31] Yingen Xiong, Bing Fang, and F. Quek. "Detection of Mouth Movements and its Applications to Cross-Modal Analysis of Planning Meetings". In: *International Conference on Multimedia Information Networking and Security* 1 (2009), pp. 225–229.

[32] X. Anguera et al. "Speaker diarization: A review of recent research". In: *IEEE Transactions on Audio, Speech and Language Processing* 20(2) (2010), pp. 356–370.

[33] F. Vallet, S. Essid, and J. Carrive. "A Multimodal Approach to Speaker Diarization on TV Talk-Shows". In: *IEEE Transactions on Multimedia* 15(3) (2013), pp. 509–520.

[34] A. Noulas, G. Englebienne, and B.J.A. Krose. "Multimodal Speaker Diarization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(1) (2012), pp. 79–93.

[35] M. Bendris, D. Charlet, and G. Chollet. "People indexing in TV-content using lip-activity and unsupervised audio-visual identity verification". In: *International Workshop on Content-Based Multimedia Indexing* (2011), pp. 139–144.

[36] E. El Khoury et al. "Association of Audio and Video Segmentations for Automatic Person Indexing". In: *International Workshop on Content-Based Multimedia Indexing* (2007), pp. 287–294.

[37] M. Bendris, D. Charlet, and G. Chollet. "Talking faces indexing in TV-content". In: *International Workshop on Content-Based Multimedia Indexing* (2010), pp. 1–6.

[38] Zhu Liu and Yao Wang. "Major Cast Detection in Video Using Both Speaker and Face Information". In: *IEEE Transactions on Multimedia* 9(1) (2007), pp. 89–101.

[39] A. Das, M.R. Jena, and K.K. Barik. "Mel-Frequency Cepstral Coefficient (MFCC) - a Novel Method for Speaker Recognition". In: *Digital Technologies* 1(1) (2014), pp. 1–3.

[40] H. Seddik, A. Rahmouni, and M. Sayadi. "Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier". In: *First International Symposium on Control, Communications and Signal Processing, Proceedings of IEEE* (2004), pp. 631–634.

[41] *VOICEBOX: Speech Processing Toolbox for MATLAB*. URL: `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html` (visited on 11/02/2014).

[42] J. Šochman and J. Matas. "WaldBoost - Learning for Time Constrained Sequential Detection". In: *Conference on Computer Vision and Pattern Recognition* (2005), pp. 150–157.

[43] *Eyedea Face Detection*. URL: `http://www.eyedea.cz/eyeface-sdk/` (visited on 10/07/2014).

[44] A. Asthana et al. "Incremental Face Alignment in the Wild". In: *Conference on Computer Vision and Pattern Recognition* (2014).

[45] Xuehan Xiong and F. De la Torre. "Supervised descent method and its application to face alignment". In: *Conference on Computer Vision and Pattern Recognition* (2013).

[46] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000, pp. 71–72.

[47] H. Hotelling. “Relations between two sets of variates”. In: *Biometrika* 28 (1936), pp. 321–377.

[48] H. Knutsson, M. Borga, and T. Landelius. “Learning canonical correlations”. In: *Tech. Rep. LiTH-ISY-R-1761, Computer Vision Laboratory, S-581 83, Linkoping University, Sweden* (1995).

[49] A.J. Aubrey et al. “Cardiff Conversation Database (CCDb): A Database of Natural Dyadic Conversations”. In: *V & L Net Workshop on Language for Vision* (2013).

[50] B. Goswami et al. “Speaker Authentication using Video based Lip Information”. In: *Proc. ICASSP* (2001), pp. 1908–1911.

Appendix A

Contents of the enclosed DVD

directory	content
thesis	This thesis in PDF
code	Matlab code for the proposed method
data	Not publicly available data used in this thesis

Table 2 Content of the attached DVD