University of Thessaly

Master Thesis

Voice Activity Detection Using Audio, Video and Depth Information

Author:
Spyridon Thermos

Supervisor:
Dr. Gerasimos Potamianos

Additional Committee Members:
Dr. Georgios Stamoulis
Dr. Antonios Argyriou

A thesis submitted in fulfilment of the requirements
for the degree of Master in Science and Technology of ECE

in the

Department of Electrical and Computer Engineering

July 2015

UNIVERSITY OF THESSALY

Abstract

Department of Electrical and Computer Engineering

Master in Science and Technology of ECE

Voice Activity Detection Using Audio, Video and Depth Information

by Spyridon THERMOS

The need for better human-computer interaction has been a great motivation for the development of robust audio-visual automatic speech recognition algorithms (AVASR). AVASR systems' performance is significantly affected from speech or voice activity detection (VAD). VAD is a fundamental task in various applications such as teleconference rooms and smart homes. It typically comprises a feature extraction and a speech non-speech separation mechanism.

This thesis presents a supervised learning VAD system and employs algorithms that utilize audio, video and depth information to detect human speech. For the needs of this thesis, a database was created to examine a two-speaker scenario captured by two Microsoft Kinect sensors. Experiments conducted on this database indicate that the method proposed in the current thesis proves quite robust in detecting the active speaker in cases of acoustic and visual noise.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Περίληψη

Αναγνώριση Δραστηριότητας Ομιλίας από Δεδομένα Ήχου, Βίντεο και Βάθους

Η ανάγκη για καλύτερη επικοινωνία μεταξύ ανθρώπου και υπολογιστή έχει υπάρξη μεγάλο κίνητρο για τη δημιουργία αποδοτικών αλγορίθμων για οπτικοακουστική αυτόματη αναγνώριση ομιλίας. Η απόδοση των συστημάτων οπτικοακουστικής αυτόματης αναγνώρισης ομιλίας εξαρτάται σημαντικά από την ανίχνευση φωνητικής δραστηριότητας ή λόγου. Η ανίχνευση φωνητικής δραστηριότητας είναι θεμελιώδης διεργασία σε ποικίλες εφαρμογές όπως οι αίθουσες τηλεδιάσκεψης και τα "έξυπνα" σπίτια. Συνήθως αποτελείται από ένα μηχανισμό εξαγωγής χαρακτηριστικών και έναν διαχωρισμού ομιλίας μη-ομιλίας.

Στην εργασία αυτή παρουσιάζεται ένα επιβλεπόμενης μάθησης (supervised learning) σύστημα ανίχνευσης φωνητικής δραστηριότητας και χρησιμοποιούνται αλγόριθμοι που αξιοποιούν ακουστική και οπτική πληροφορία καθώς και πληροφορία βάθους για να ανιχνεύσουν ανθρώπινη ομιλία. Για τις ανάγκες της εργασίας, δημιουργήθηκε μία βάση δεδομένων για να εξεταστεί η περίπτωση δύο ομιλητών που καταγράφονται από δύο Microsoft Kinect αισθητήρες. Τα πειράματα που διενεργήθηκαν σε αυτή τη βάση δεδομένων υποδεικνύουν ότι η μέθοδος που προτείνεται στη συγκεκριμένη εργασία αποδεικνύεται αρκετά αποδοτική στον εντοπισμό του ενεργού ομιλητή σε περιπτώσεις ακουστικού και οπτικού θορύβου.

Acknowledgements

I would like to thank my supervisor, Professor Gerasimos Potamianos for his invaluable support and guidance during my postgraduate studies. His critical observations and the way of analyzing and solving problems was inspiring. Our collaboration has made me a better engineer. I would also like to thank Dr. Stamoulis and Dr. Argyriou, for participating in the thesis committee.

Special thanks go to my close friend and colleague Konstantinos Kyritsis for his critical help during this thesis.

Finally, I would like to thank Iordana and my family for their constant support and their patience all these years.

Contents

List of Figures

List of Tables

Abbreviations

VAD	**V**oice **A**ctivity **D**etection
ASR	**A**utomatic **S**peech **R**ecognition
DCT	**D**iscrete **C**osine **T**ransform
ROI	**R**egion **O**f **I**nterest
SVM	**S**upport **V**ector **M**achines
FMN	**F**eature **M**ean **N**ormalization
LDA	**L**inear **D**iscriminant **A**nalysis
PCA	**P**rincipal **C**omponent **A**nalysis
CCA	**C**anonical **C**orrelation **A**nalysis

To Iordana and my friends …

Chapter 1

Introduction

1.1 Overview

Over the past few years, the need for alternative methods of interactions with computer systems has been a great motivation for researchers in the speech processing community to develop robust automatic speech recognition (ASR) algorithms. A significant step which affects directly the performance of ASR systems is the speech or voice activity detection (VAD). Voice activity detection is the problem of determining the existence of speech from an audio signal and separating the speech from the non-speech segments. VAD plays a vital role in different speech processing systems such as in speech coding, speech diarization and in front-end processing for speech recognition applications. However, when the audio signal is corrupted by acoustic noise, it can be very hard for an audio-only VAD system to distinguish between speech and non-speech or to identify a speaker. To overcome this problem, the research community exploited the visual speech information from the speaker's mouth region which is invariant to the acoustic environment and is available in a variety of speech related products such as teleconference rooms and smart homes. The main two components of VAD are feature extraction and classification. The typical VAD process is presented in figure 1.1.

FIGURE 1.1: The block diagram of a typical VAD system.

1.2 Contribution of this Thesis

In typical audio-visual VAD systems, visual speech information is extracted from planar frontal or profile video of speakers' faces [1] which results in robust detection even when it is combined with noisy audio information within a two-stream classifier. In this thesis, we present a three-stream VAD systemin a two-speaker scenario, exploiting the abilities of a novel multi-sensor device, the Microsoft Kinect. Furthermore, we present the methodology followed to collect a connected digits database using two Kinect sensors which acquire planar and depth data from two-speaker groups.

1.3 Thesis Organization

The remainder of this thesis is organized as follow. Chapter 2 presents a supervised learning VAD system that was developed for speech detection utilizing audio,color planar and depth information. Moreover, the same chapter presents an algorithm that examines the correlation between the different audio-visual modalities. Chapter 3 overviews the Microsoft Kinect device used for the data acquisition, presents our database and more specifically the setup, the environment and the details of the recordings. Chapter 4 presents the experiments conducted with the two algorithms mentioned in Chapter 2, using training and testing data from the database presented in Chapter 3. Finally, Chapter 5 summarizes and concludes the thesis.

Chapter 2

Theory and System Architecture

2.1 Acoustic Front End

With all the improvements over audio-only VAD, developing a three-stream audio-visual speech detection system [2] introduces new challenges and difficulties that should be tackled. Here, we choose to follow a traditional approach based on mel-frequency cepstral coefficients (MFCCs) as audio features. The analysis of the audio speech signal has been already extensively investigated in the speech processing community. Thus, audio speech detection will not be improved within this thesis. Instead, we choose to extract the Mel Frequency Cepstral Coefficients (MFCCs). The mel frequency cepstrum is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a logarithm power spectrum on a nonlinear mel scale of frequency. The difference between the cepstrum and the mel frequency cepstrum is that in the mel frequency cepstrum, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum.

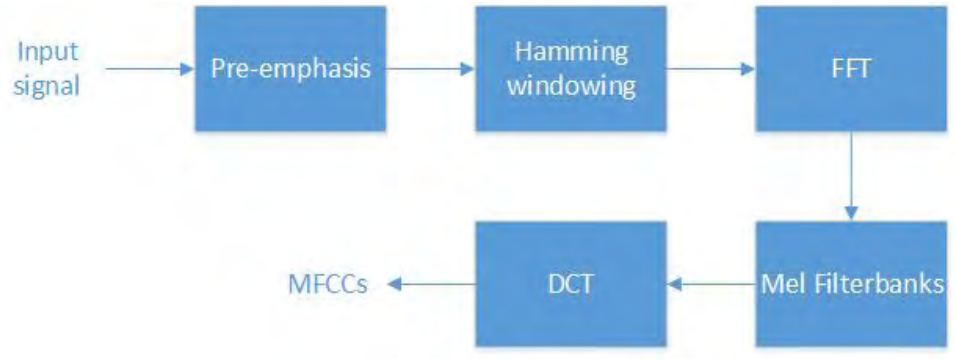


FIGURE 2.1: MFCCs extraction algorithm.

The required MFCCs from the audio signal are extracted with the following process, presented in figure 2.1. Firstly, the audio signal is framed into 25ms frames with 10ms frame overlap. Generally, the width of the frames should be between 20-40 ms because very short frames don't have enough samples to get a reliable spectral estimate and in very long frames the signal may change too much. Consequently, power spectrum of each frame was calculated by implementing Fourier Transform for each windowed signal. The calculation of the MFCCs is based on mel-spaced filterbanks. This is a set of 20-40 (26 is the standard) triangular filters. Our filterbank comes in the form of 26 vectors. Each vector is composed mostly of zero values except for a certain section of the spectrum. To calculate filterbank energies each filterbank is multiplied with the power spectrum and then the coefficients are added up. Finally, this computation results to 26 numbers that give us an indication of the amount of energy in each filterbank. In addition, these energy values are mapped on the mel-scale using the equation (2.1).

$$M(f) = 1125ln(1 + \frac{f}{700}) \tag{2.1}$$

Once the logarithm of these 26 energies is calculated, Discrete Cosine Transform (DCT) is applied resulting to 26 cepstral coefficients as shown in equation (2.2), where M represents the number of the filterbanks and $S[m]$ the logarithm of each filterbank's energy, respectively. Finally, we maintain the first 13 out of the 26 coefficients for the needs of the audio front-end:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos \frac{\pi n(m - \frac{1}{2})}{M} \qquad , 0 \leq n < M. \tag{2.2}$$

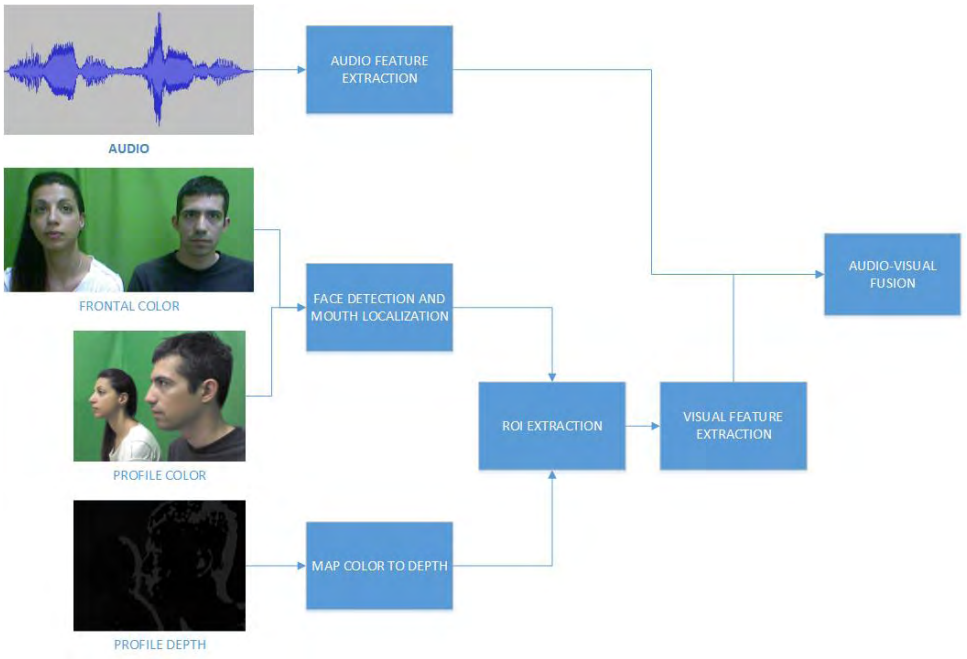


FIGURE 2.2: The main processing blocks of an audio-visual and depth VAD.

2.2 Visual Front-End

The major issue in audio-visual voice activity detection (AVVAD) is the visual front-end design. Given a group of visual streams of a person speaking, the main goal of the visual front-end system is to extract visual features from the mouth region that contain sufficient visual speech information. This process is composed of two stages, face detection and mouth localization.

2.2.1 Face Detection

The Viola-Jones detector [3] is used in order to achieve robust face detection. There are three steps that Viola and Jones combined to implement this method. The first is an image representation call “integral image”. The “integral image” computes a value at each pixel (x,y) that is the sum of the pixel values above an to the left of (x,y) inclusive, as shown in figure 2.3. This can be computed quickly in one pass through the image. Using the integral image any rectangular sum can be computed in four array references, presented in figure 2.4. The second is a simple and efficient classifier built using the Adaboost learning algorithm [4] to select the most critical features from a very large set of potential ones. Adaboost forces every weak classifiers to depend on only a single feature.

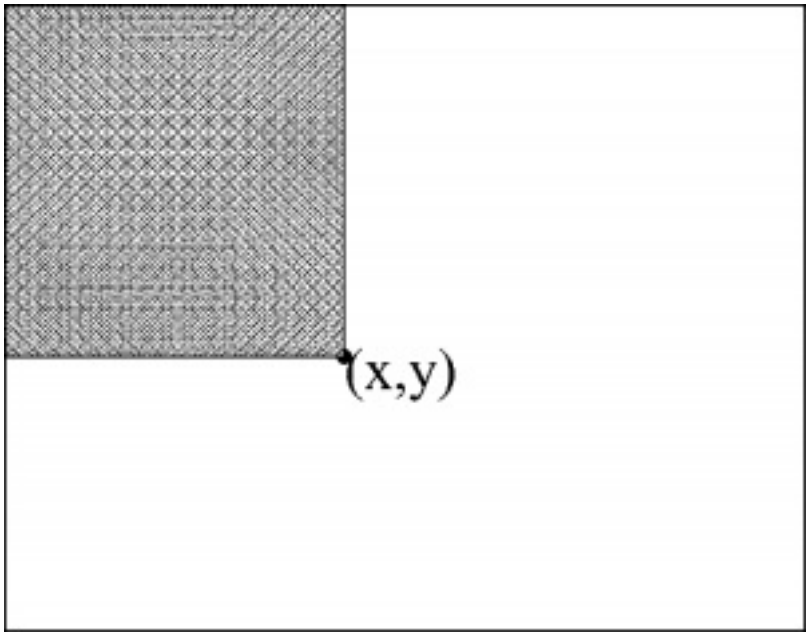


FIGURE 2.3: The value of the integral image at point (x,y) is the sum of the above and to the left pixels (from [5]).

As a result, every step of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. The third is the process of combining all the weak classifiers in a cascade [5], like Haar which allows background regions of the image to be quickly discarded and simultaneously spending more computation on the face-like regions, as we can see in figure 2.5. This detector is utilized twice in our implementation, once for the frontal frames and once for the profile frames. The implementation of the detector was accomplished using the OpenCV library (mentioned in Chapter 4).

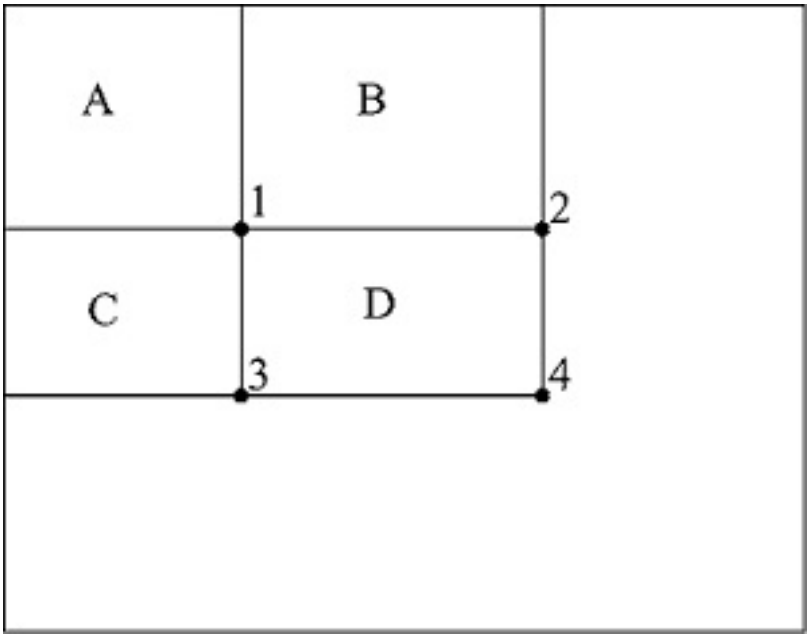


FIGURE 2.4: The sum of the pixels within rectangle D can be computed with four array references. Value at location 1 is A, at 2 is A+B, at 3 is A+C and at location 4 is A+B+C+D (from [5]).

2.2.2 Mouth Localization

As mentioned before, we need to locate the mouth region for every face in order to extract the required features. Thus, for every face detected, we implement the same method as face detection except for the use of the Haar mouth cascade, which substitutes the face cascade in the algorithm. This nested detection of the mouth decreases the number of possible false positives in the image, while preserving the performance at a high level. Moreover, by calculating the median coordinates of the bounding box for five consecutive frames, robust mouth detection is achieved, overcoming possible false detections and sudden movements. With these coordinates extracted from the color frames and using the coordinate mapper of Kinect SDK that maps the color to the depth pixels, we overcome the disparity of the two sensors and locate the mouth for the detected faces in depth frames. Finally, we resize the mouth bounding box to 64×64 pixels, so that we can implement the DCT and extract the desired coefficients. An example of this process for frontal and profile frames is depicted in figure 2.6.

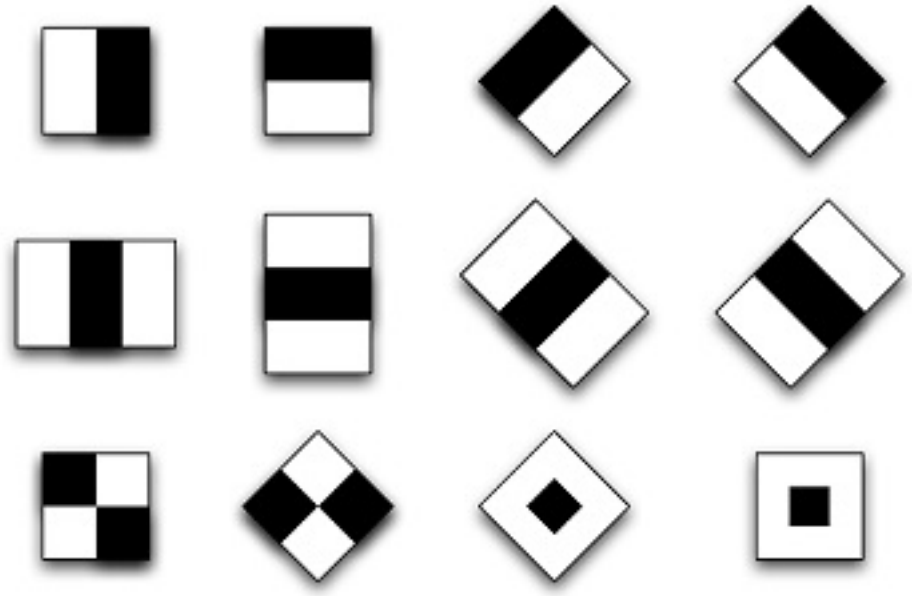


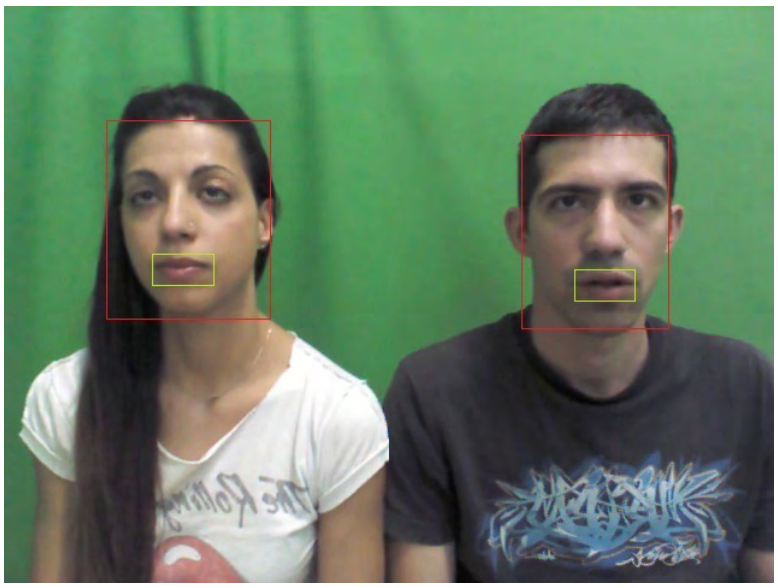
FIGURE 2.5: Haar-like edge, line and center-surround features (from [6]).

2.2.3 Color Feature Extraction

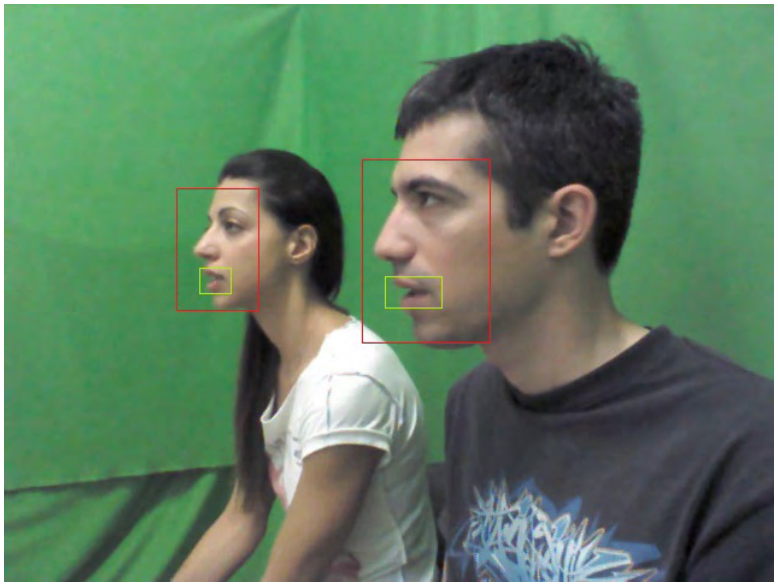
In order to extract the required color features the following process is proposed. The frames collected with the Kinect's RGB camera were set at 30 Frames Per Second (FPS). Initially, each frame of the video sequence was filtered for noise reduction and the face for both speakers was detected using Haar's face cascade (frontal and profile). In addition, for each face, the mouth region was detected implementing the Haar mouth cascade inside the face's bounding box.Furthermore, in order to ensure that the mouth tracking process has no false detections or other issues that can distort our data, the coordinates of the mouth region's bounding box were filtered by calculating the median value in a 30-frame window. The required Region of Interest (ROI) was obtained by resizing the mouth bounding box to 64×64 pixels.

2.2.4 Depth Feature Extraction

To exploit the correspondence of the Kinect's color and depth stream, an algorithm that maps the color to the depth frame pixels was used. Given the bounding box that represents the mouth region for the color frames and the synchronization of the color and the depth stream, the mouth region in the depth frame can be easily detected by simply mapping the color's bounding box coordinates to depth coordinates respectively. The color's bounding box coordinates were mapped to the depth frame using Kinect SDK's "coordinatemapper" class. Finally, the mouth ROI that contains the necessary depth features was resized to 64×64 pixels.



i Frontal face and mouth detection.



ii Profile face and mouth detection.

iii Left speaker frontal mouth region.

iv Right speaker frontal mouth region.

FIGURE 2.6: Face detection - Mouth localization example.

2.3 Voice Activity Detection

2.3.1 Multimodal Fusion

The dimensionality of the extracted feature vector from color(frontal and profile) and depth frames is too large (4096 features) to allow successful training and classification, thus dimensionality reduction is required. There are several linear transforms that apply dimensionality reduction and preserve the most relevant information. Most common transforms are DCT [1], Principal Component Analysis (PCA) [7], Linear Discriminant Analysis (LDA) [8] and discrete wavelet transform [9]. DCT is a transform borrowed from the image compression field, used to reduce dimensionality of the feature vector while preserving most of the information contained in the image. Once it is one of the most used transforms in audio-visual VAD and ASR, we present it here in detail. The two-dimensional DCT is a real and orthogonal transform. The definition of the transform [10] for an input image A and output image B is:

$$B_{pq} = a_p a_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{min} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \text{for } \begin{cases} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1. \end{cases} \tag{2.3}$$

$$\text{where } a_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1. \end{cases} \tag{2.4}$$

$$\text{and } a_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1. \end{cases} \tag{2.5}$$

where M and N are the row and column size of A, respectively. Because of the correlation in the images, many coefficients of the output image are close to zero. The high-energy coefficients are typically grouped in the upper-left corner of the image. Thus, in order to extract the DCT coefficients for our VAD system, a zig-zag scan is used, as shown in figure 2.7, to acquire the 45 high-energy coefficients.

Although our implementation requires audio and video streams to be synchronized, they differ. To resolve this problem, simple element-wise linear interpolation of the color-depth features to the audio framerate is implemented. Furthermore, recording conditions and different voice variations can be corrected by visual feature mean normalization (FMN), a method that consists in removing the temporal mean of the features for each sequence, that is, for each individual speaker in the database.

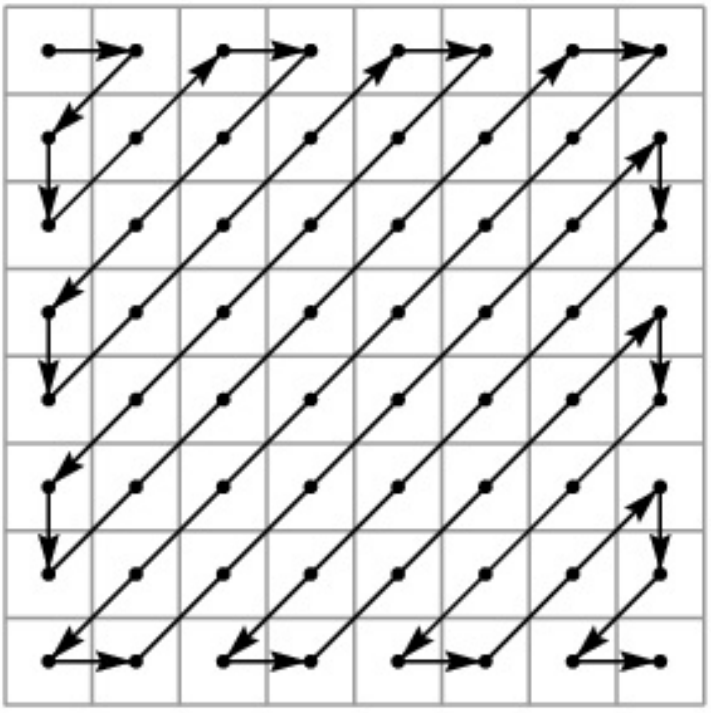


FIGURE 2.7: The zig-zag coefficient ordering scheme for 2D-DCT. The algorithm starts from the upper left corner and applies a zig-zag element scan (from [11]).

In addition, further dimensionality reduction can be achieved using LDA. LDA is a supervised transform that uses class labels and aims to separate and, at the same time, compact the classes. LDA maximizes the ratio of between-class scatter and within-class scatter and results to a lower-dimensional representation of the input vector. In order to capture dynamic visual speech information we concatenate j neighboring LDA feature vectors at each side of the current frame creating a $(2j + 1) \times 45$ matrix.

It should be mentioned that the LDA and the j neighboring concatenation is also applied to the MFCC vector. Finally, audio, color and depth feature vectors are concatenated in order to create a multimodal feature vector, with $4(2j + 1)$ dimension, as an input for the classification module.

2.3.2 Classification

Support Vector Machines are supervised classifiers with associated learning algorithms that analyze data and recognize patterns, used for classification. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. In our case we have four labels ('no speech', 'person1 speaking', 'person2 speaking', 'both speaking') so we need to train four one-versus-all SVM classifiers.The audio-visual and classification block diagram is given in figure 2.8.

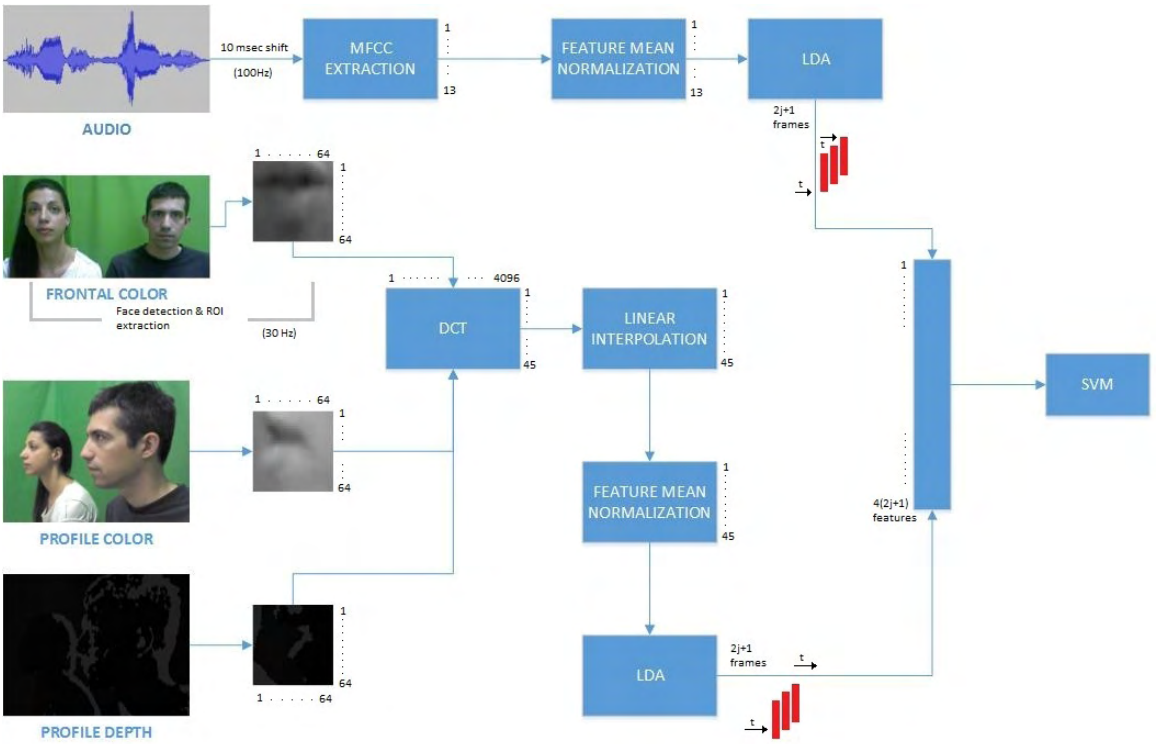


FIGURE 2.8: The algorithm that generates time-synchronous audio, color and depth feature vectors and train the classifier.

2.3.3 Canonical Correlation Analysis

In addition to the above method, we consider a different approach to the problem, based on canonical correlation analysis (CCA) [12], [13]. CCA is a method of judging correlation between two multidimensional variables. It can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors is mutually maximized. Let x and y be a DCT frame vector and a MFCC frame vector respectively, with dimensions N_x and N_y. CCA searches for two linear transformations A and B that maximize the mutual information between the transformed variables x' and y' where the multidimensional variables are represented with column vectors.

$$\begin{aligned} x' &= Ax \\ y' &= Bx, \end{aligned} \tag{2.6}$$

The A and B transformations are represented by $N \times N_x$ and $N \times N_y$ matrices where N is equal to the minimum dimension of N_x and N_y ($N \leq min(N_x, N_y)$). The rows of these matrices form an orthonormal basis for the corresponding transform space and are the correlation basis vectors a and b respectively. The first pair of these vectors, (a_1, b_1), is given by the directions along which the projections are maximally correlated:

$$(a_{x1}, b_{y1}) = \arg\max_{a_x, b_y} Corr(a_x^T x, b_y^T y) \tag{2.7}$$

The projections, $x'_1 = a_{x1}^T x$ and $y'_1 = b_{y1}^T y$, are the first pair of canonical components.

In order to get x and y for every frame of audio and video, we follow the same process as in the section above until the FMN. Instead of implementing LDA for dimensionality reduction, we use the canonical correlation analysis to create transformation matrices A and B, as well as the vector r that contains the correlation coefficients. As soon as we have created these correlation matrices we can use them to project any test vector of MFCCs and DCTs. As a result we have $x_{out} = Ax_{test}$ and $y_{out} = By_{test}$. Finally, we can calculate the correlation between the elements of each row for x_{out} and y_{out} and compare the result vector with the ground truth. The process above is depicted in figure 2.9.

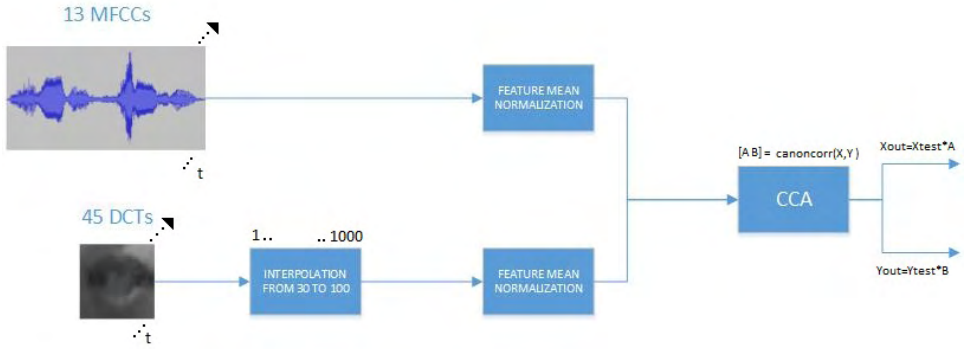


FIGURE 2.9: The block diagram of our CCA implementation.

Chapter 3

The Database

In this chapter, we provide a brief overview of the devices that can be used to acquire the required data we need, i.e. 2D cameras, depth cameras and microphones. Furthermore, we present in detail the device we finally use for the data acquisition, Microsoft's Kinect. Finally we describe the setup of the recordings and the type of data we collected.

3.1 Devices

3.1.1 2D Cameras

The 2D cameras capture the visible spectrum of light and provide color images such as RGB or YUV images. A 2D camera can have various image resolutions, HD (1920×1080), VGA (640×480) or QVGA (320×240) and capture images at various frame rates such as 25, 30 or even 60 FPS. Most 2D cameras are cheap and we usually find them embedded on laptops, smartphones, tablets and they tend to be sensitive to light variations.

3.1.2 Depth Cameras

Depth cameras offer a disparity image, showing relative distances to the camera. They are presented by many names such as ranging camera, flash lidar, time-of-flight (ToF) camera, and RGB-D camera. The underlying sensing mechanisms are equally varied like range-gated ToF, RF-modulated ToF, pulsed-light ToF, and projected-light stereo. The commonality is that all provide traditional (sometimes color) images and depth information for each pixel (depth images) at a certain framerate. We can use a depth camera to acquire valuable distance data for VAD.

3.1.3 Microphones

Audio data are crucial to voice activity detection. We use microphone or microphones to capture raw audio data through one or more channels and process these data so that we can extract from them the features we will need.

FIGURE 3.1: Microsoft Kinect for Windows (from [14])

3.2 Microsoft's Kinect

The Microsoft Kinect sensor, shown in figure 3.1, is a novel device developed mainly for gesture recognition and skeleton tracking. After years of research Microsoft decided to build an SDK, add various modules (i.e. near mode) and finally offered the community an opportunity to find out the limits of the device. The Kinect sensor can capture audio, video and depth data, as we can see in figure 3.2, making it a very practical device to use for computer vision, robotics and audio-visual research purposes. It has four build-in microphones, a VGA (640×480) camera and a depth camera with the same resolution (also available at 320×240). In order to capture depth information, Kinect sensor uses a laser emitter, an IR camera receiver and the structured light methodology [15]. In more detail, a laser beam passes through a grating, splits into more beams that are reflected from any object inside the FOV of the Kinect sensor and received from an infrared camera receiver, as shown in figure 3.3, making it possible to calculate the distance of the object using triangulation.

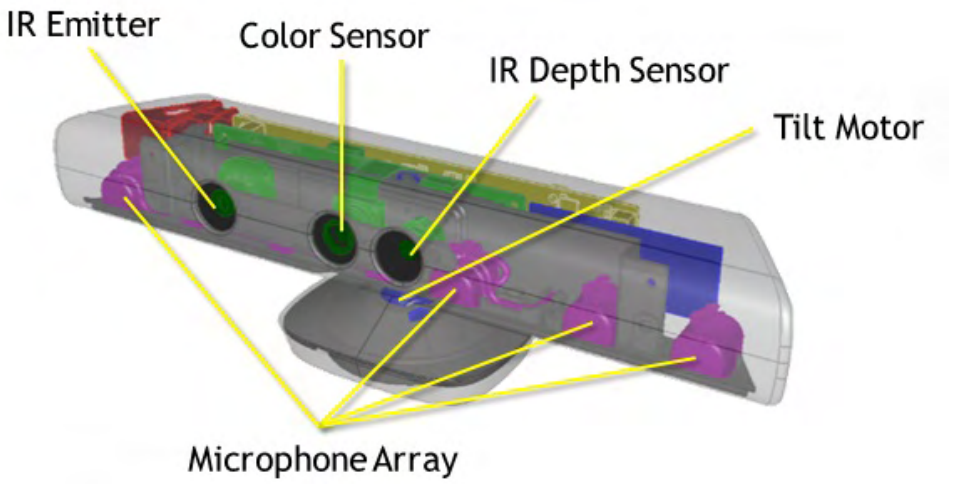


FIGURE 3.2: Microsoft Kinect sensors demonstration (from [14])

Moreover, it has two modes of capturing: the default and the near mode. The default mode allows the sensor to capture depth from minimum 800mm and maximum 8000mm distance, although the effective distance for capturing is 800mm to 4000mm. The near mode allows the sensor to capture depth from minimum 400mm and same maximum distance, while the effective range in this mode is 400mm to 3000mm, as we can see in figure 3.4. When an object is too near for both modes the sensor gives the pixels that represent its distance from the camera, the 0x0000 value and if it cannot define the distance for an object out of maximum range, it gives 0x1000 (near mode) or 0xfff8 (default mode). In our experiments we used two Kinect sensors, one for frontal view and one for profile, both at video resolution 640×480 (24-bit 30FPS), depth resolution 640×480 (16-bit 30FPS) and audio capture, 16-bit PCM format, at 16kHz.

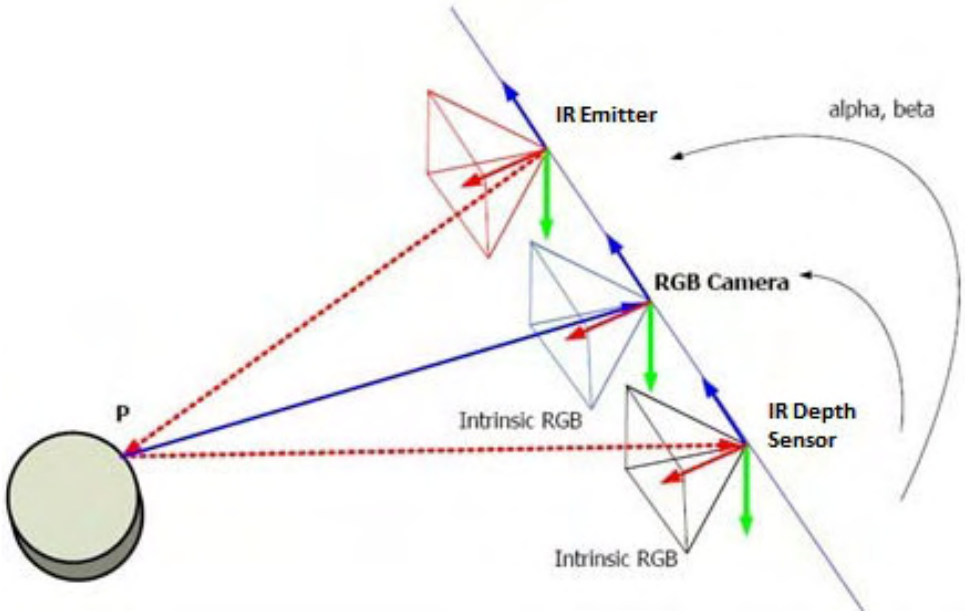


FIGURE 3.3: The structured light methodology for distance calculation through triangulation (from [15])

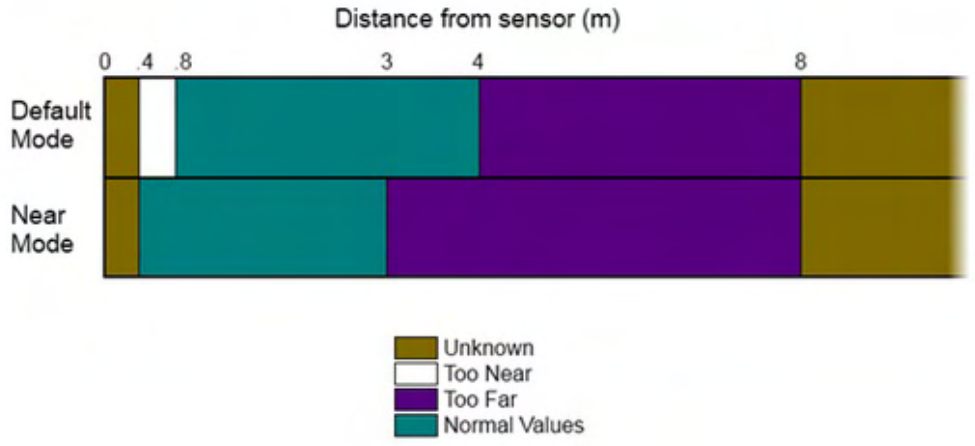


FIGURE 3.4: Kinect sensor's Default and Near mode ranges (from [16])

3.3 Setup

Due to the lack of databases that include profile depth information, the creation of our own database was required. There are quite a lot of databases that include audio-visual single or multi-speaker scenarios such as [17], others that include frontal depth information such as [18] however, for the needs of this thesis we created a database that consists of audio, color and depth information captured from both frontal and profile views. The database was shot at the Visual Computing Laboratory of Information Technology Institute (ITI) at the Centre of Research and Technology Hellas (CERTH) [19]. The lab was characterized by two main benefits for recording our database. First, the background noise level was minimized during the recording and also the illumination of the lab was controlled. Second, we used solid green background making face detection and head tracking easier and of course enabling the background removal, as shown in figure 3.5. Two Microsoft Kinects and 5 different people were used. The first Kinect was placed in front of the subjects at $0.8-0.9m$ head-to-camera distance, enabling the frontal face detection and mouth localization. The second one was placed at the right side of the subjects, at a 75° angle and $0.4-0.5m$ nearest head-to-camera distance, making it easier to capture more than one face. The database was built on two modes. In the first mode we capture depth from the side Kinect, while in the second we capture depth from the frontal Kinect.

FIGURE 3.5: The data collection process displaying the devices and the configuration.

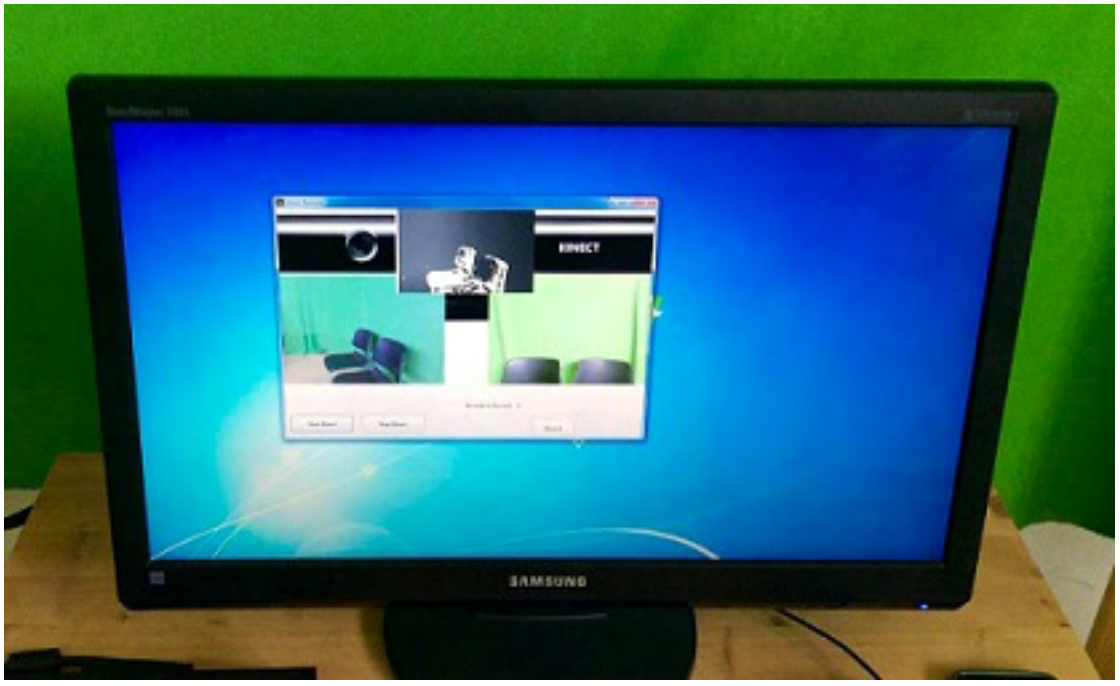


FIGURE 3.6: The monitor showing the GUI and the sequence of the 4-digit numbers.

3.4 The Data

The speaker/speakers were asked to read in a continuous manner random 4-digit numbers (digits 0-9 in English) that were displayed on a monitor behind the frontal Kinect, as we can see in figure 3.6. For the two-speaker mode, we included the possibility that one's reading can be extended over the other for a short period of time. From each Kinect sensor, we saved all three streams-audio, color, depth. The actual framerate of the streams was around 30 FPS but we synchronized the streams precisely at 30 FPS using frame timestamps. By recording the capturing timestamp of each frame we could synchronize all streams considering colorstream (RGB) as a reference and normalizing the framerate [18]. More specifically, we separated the stream in 10-frame segments and by using their timestamps we calculated the framerate of each segment. Therefore, if the segment's framerate was over 30 FPS we dropped a frame and if it was lower than 30 we used linear interpolation to duplicate frames. Because of the SDK that Microsoft offers [20], only in a few recordings synchronization and framerate normalization were needed. All of the frame sequences, audio color and depth were segmented at the 5-digit sequence level. The color and depth frames captured by the Kinect sensors were saved in PNG format, which has good compression rate and is lossless. Moreover, we saved raw depth data as well as ".depth" files. The color frames were saved as 24-bits three channel PNG images, while depth frames as single channel 13-bit grayscale PNG images. Actually the depth data were saved in 2-byte arrays where the 13 high-order bits have the depth information and the 3 low-order bits have the player index, that we don't need. So, by right logical shifting (3-bit shift) we isolated the data we needed. The audio from both Kinect sensors was saved as 16-bit PCM format at 16kHz and was also converted to WAV.

Chapter 4

Experiments

In this chapter we present the results of the experiments conducted to validate the effectiveness of using depth features to achieve higher VAD when audio-visual noise exists. Also we evaluate the results of implementing CCA using audio, planar color, profile color and profile depth features. For our experiments we use data from our database described at chapter 4 and more specifically, we use 9 from the 11 sessions as training set and the rest 2 as Test set, as shown in table 4.1.

Two-speaker groups	SVM	CCA
g01	training	training
g02	test	training
g03	training	training
g04	training	training
g05	training	test
g06	test	training
g07	training	training
g08	training	training
g09	training	test
g10	training	training
g11	training	training

TABLE 4.1: The table presents the two-speaker groups of the database and whether they were used for training or testing.

4.1 OpenCV library

OpenCV [21] is a library of programming functions mainly aimed at real-time computer vision. It is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. Also, there are more interfaces in Python, Java and Matlab/Octave. OpenCV library has many built in image processing and computer vision algorithms. Several cascade classifiers, such as Haar mentioned in Chapter 2, in XML format can be used from these algorithms for any purpose. There are cascades for face, nose, eyes, mouth and also for items such as clocks. We used the OpenCV library in Microsoft Visual Studio [22] environment to process our data and extract the features we need.

4.2 SVM test

In this section we present the experiments conducted under the influence of audio-visual noise following the process reported in Section 2.3.1 . We evaluate results based on three feature vectors given as input for testing, Audio-Frontal Color, Audio-Frontal Color-Profile Color and Audio-Frontal Color-Profile Depth (AFc, AFcPc and AFcPd respectively).

4.2.1 Audio Noise

To simulate cases such as background noise, stuttering and other distortions we mixed the Test set's audio with additive white Gaussian noise for various SNR values. As we can infer from figure 4.1, where the results of this experiment are, all three fusion strategies achieve highest accuracy for 10dB AWGN. Moreover, we can infer that for lower SNR values accuracy is significantly reduced but on the other hand when we add profile color or depth features the VAD increases and especially for -5 dB.

4.2.2 Visual Noise

In this section we present the results of the experiments conducted using two different types of visual noise. These two types are random Gaussian noise for different variances and random block noise for various block size. We use these two types of noise to simulate distortions from camera's malfunction or signal transmission error. Moreover, with block noise we can simulate information loss from abrupt mouth movements and possible scars or any other health condition causes on lip-mouth region.

We first examine the random Gaussian noise scenario. In this scenario we degrade the quality of the ROI of frontal color frames using zero mean and three different variance values, 0.01, 0.04 and 0.09 respectively. This degradation is depicted in figure 4.2.

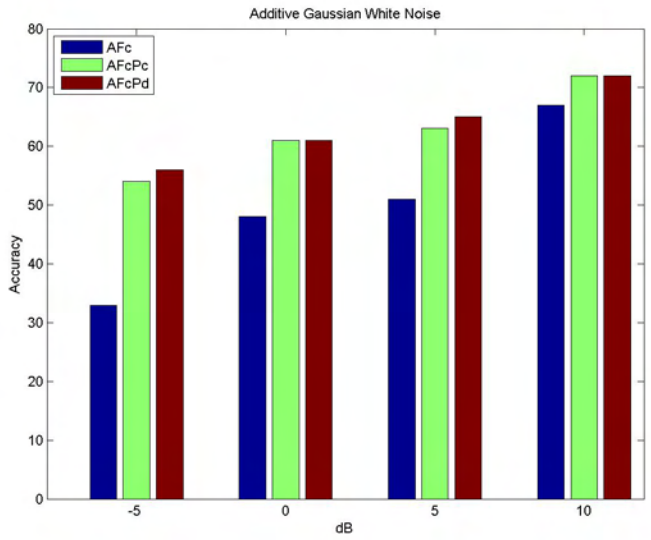
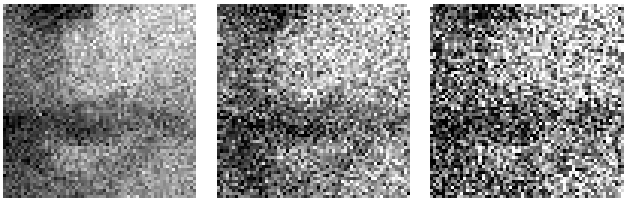


FIGURE 4.1: Voice activity detection accuracy of the three fusion setups for various SNR values.



(α′) (β′) (γ′)

FIGURE 4.2: ROI with Gaussian noise of variance **a.**0.01 **b.**0.04 and **c.**0.09.

As we can see all fusion setups achieve high quality in VAD for smaller variance. However, when variance has higher values, profile color and especially profile depth features are significantly helpful to increase accuracy and this scenario is depicted in figure 4.3 where for 0.09 variance Audio-Frontal Color accuracy decreases about 10% compared to 0.01 variance accuracy, while Audio-Frontal Color-Profile Depth's accuracy is about 70%.

As previously mentioned, the second type of visual noise we experimented with was block noise. In this experiment a random fixed size block of the ROI turns black. We tested our data for two block sizes 10×10 and 20×20 respectively. A sample of this type of visual noise in shown in figure 4.4

The results of block noise experiment are shown in figure 4.5. Here we observe that, such as in random Gaussian noise experiment, Audio-Frontal Color setup's accuracy decreases dramatically which was expected since a large portion of the ROI is occluded by a black block and we loose useful lip features. However, if we add profile color or profile depth features we achieve better accuracy and especially with the depth features that are less affected by false positives and illumination.

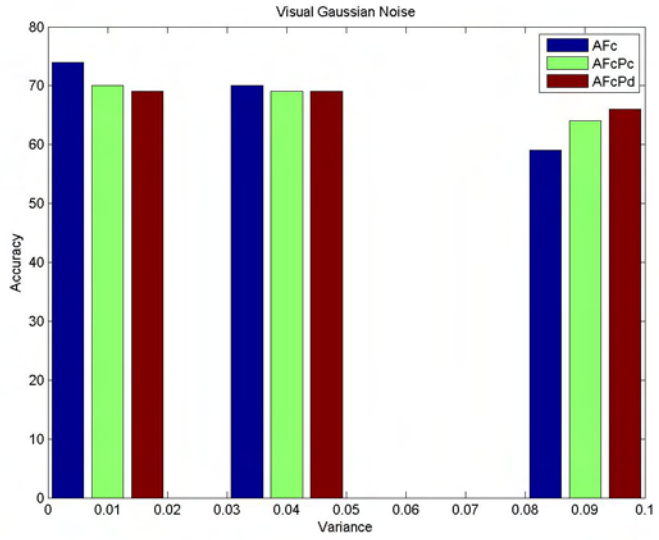
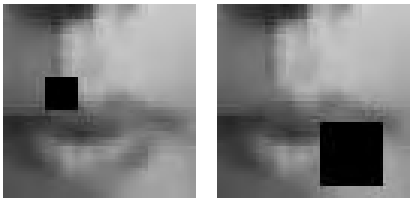


FIGURE 4.3: Voice activity detection accuracy of the three fusion setups for random Gaussian noise. The first three bars are for 0.01 variance, the second for 0.04 and the third for 0.09 respectively.



(α′) (β′)

FIGURE 4.4: ROI with block noise of block size **a.**10x10 and **b.**20x20.

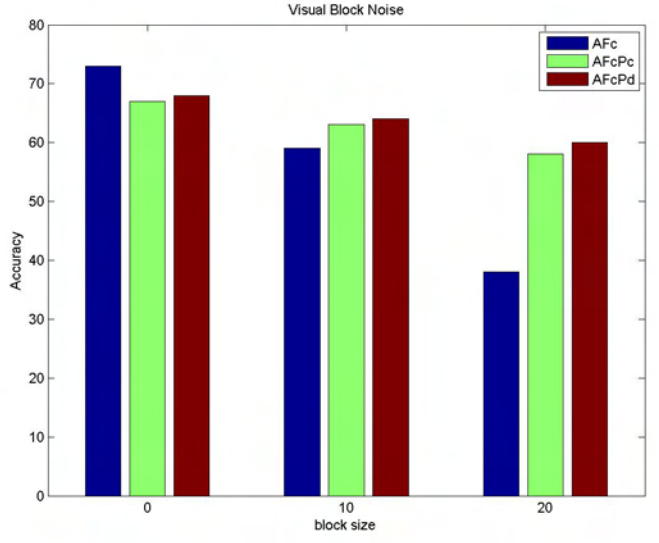


FIGURE 4.5: Voice activity detection accuracy of the three fusion setups for zero, 10x10 and 20x20 block noise.

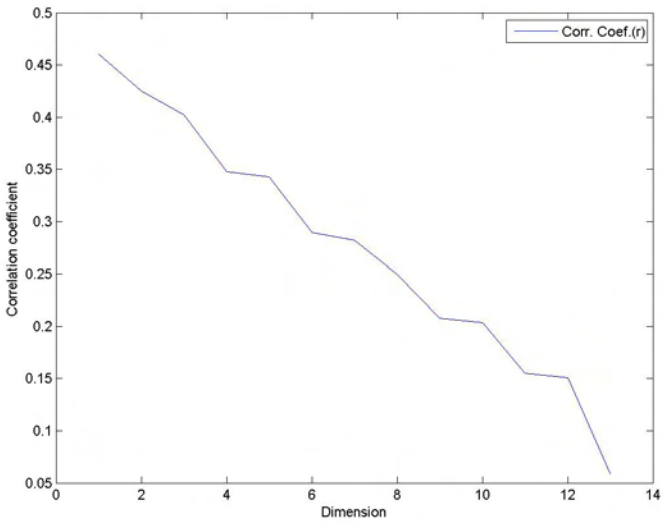


FIGURE 4.6: Canonical correlation from Audio-Frontal color analysis.

4.3 Canonical Correlation Analysis test

In this section we present the results of the experiment conducted to discover which group of features has the highest correlation and gives the highest accuracy of VAD. We use 9 of 11 groups as training set and the rest 2 as Test set and we implement canonical correlation analysis for audio-frontal color, audio-profile color and audio-profile depth setups. Following the process mentioned in Chapter 3, the lowest dimension of the input matrices (MFCC and DCT matrix) was 13 so the dimension of the first canonical basis matrix was 45×13 and the second 13×13. To exploit CCA we used a technique proposed in [12] and calculated correlation coefficients for our training set. As shown in figure 4.6, we observed that the maximum correlation coefficient was around 0.47 and the 12 out of 13 coefficient was higher than 0.17 which we used as a threshold, so we discarded the last correlation coefficient. Consequently, we defined the highly correlated components as the projections of the original data onto the A and B basis matrices along with the canonical coefficients that was above the threshold mentioned previously.

As we can infer from the test results, audio and profile depth features have slightly lower correlation than the other setups and for group 5 and 9.The highest accuracy for our fifth group was 69% and was achieved from audio-frontal color and also from audio-profile color setups, while in our ninth group it was achieved from audio-profile color setup and was 70%. The results of this experiment, for both test groups of speakers are depicted in figure 4.7.

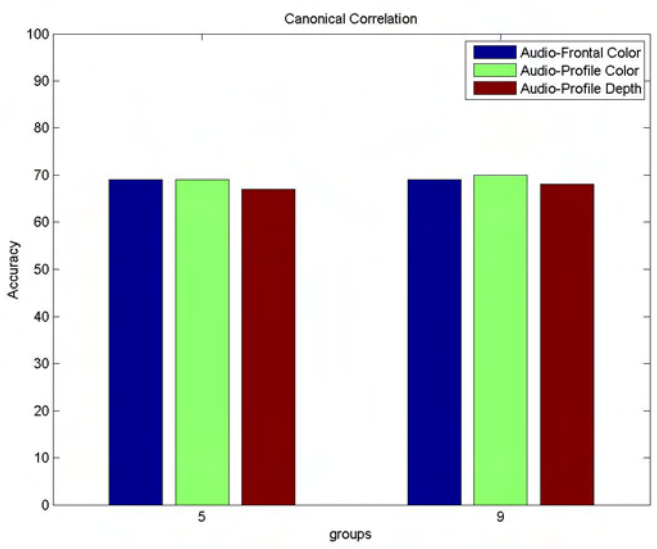


FIGURE 4.7: Voice activity detection accuracy for the 3 CCA setups.

Chapter 5

Conclusions and Extensions

5.1 Conclusions

In this thesis, we examined the contribution of depth information as an additional modality for AVVAD in a two-speaker scenario. Moreover, we created our own database for the specific frontal-profile three-stream setup, using an innovative sensor, Microsoft's Kinect , which can capture audio, color and depth information. We presented an audio-visual front-end for robust VAD.

Specifically, in chapter 2, we presented an approach for a VAD system that exploits the profile depth information in order to achieve more robust and accurate VAD and we also implemented a CCA to test which pair of audio-visual streams are highly correlation.

In Chapter 3, we overviewed the device that was used to acquire the three-stream data in the two-speaker scenario and analyzed the advantages and the limitations of this device. Furthermore, we presented the data collected as long as the environmental conditions during the recordings and the various reasons that led as to create this novel database for our experiments.

Finally, in Chapter 4, we evaluated our VAD system under noisy audio-visual conditions. Our main result regarding various types of noise, applied to frontal color and audio input data, is that depth information can improve VAD and that it is highly correlated with the audio when they are synchronized.

5.2 Extensions

Our goals for future work include further research on AVVAD with the presence of depth information. We believe that VAD systems can be improved as they can be used in various innovative audio-visual related products such as smart teleconference rooms, smart telecommunications and especially smart homes.

We also intend to expand our database, increasing the number of the speakers, creating a multi-speaker environment to evaluate our and other VAD systems and trying to simulate real-life noisy environments.

Bibliography

[1] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proc. IEEE*, 91(9):1306–1326, 2003.

[2] G. Galatas, G. Potamianos, and F. Makedon. Audio-visual speech recognition incorporating facial depth information captured by the kinect. *EUSIPCO*, 2012.

[3] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision an Pattern Recognition*, pages 1–9, 2001.

[4] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, pages 771–780, September 1999.

[5] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[6] K. Berggren and P. Gregersson. Camera focus controlled by face detection on gpu. *Lund University*.

[7] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Trans. Multimedia*, 2:141–151, September 2000.

[8] G. Potamianos, H.P. Graf, and E. Cosatto. An image transform approach for HMM based automatic lipreading. *Proc. Int. Conf. Image Processing*, 1:173–177, 1998.

[9] P. Duchnowski, U. Meier, and A. Waibel. See me, hear me: Integrating automatic speech recognition and lip-reading. *Proc. Int. Conf. Spoken Language Processing*, pages 547–550, 1994.

[10] A.K. Jain. Fundamentals of digital image processing. *Prentice-Hall*, 1989.

[11] American Mathematical Society. URL http://www.ams.org/samplings/feature-column/fcarc-image-compression.

[12] M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp. Audiovisual synchronization and fusion using canonical correlation analysis. *Multimedia, IEEE Transactions*, 9:1396–1403, 2007.

[13] D.R. Hardoon, S. Szedmak, and J.S. Taylor. Canonical correlation analysis: An overview with application learning. *Technical Report, Department of Computer Science, University of London*, CSD-TR-03-02, 2003.

[14] Kinect for Windows. URL https://www.microsoft.com/en-us/kinectforwindows/.

[15] C. Liebe, C. Padgett, J.Chapsky, D. Wilson, K.Brown, S. Jerebets, H. Goldberg, and J. Schroeder. Spacecraft hazard avoidance utilizing structured light. *Proc. of the IEEE Aerospace Conference*, page 10, 2006.

[16] MSDN. URL http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/20/near-mode-what-it-is-and-isn-t.aspx.

[17] E.K. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy. CUAVE: a new audio-visual database for multimodal human-computer interface research. *Acoustics, Speech and Signal Processing (ICASSP) IEEE International Conference*, 2:II–2017 – II–2020, 2002.

[18] G. Galatas, G.Potamianos, D. Kosmopoulos, C. McMurrough, and F. Makedon. Bilingual corpus for avasr using multiple sensors and depth information. *Proc. of the International Conference on Auditory-Visual Speech Processing (AVSP)*, pages 103–106, September 2011.

[19] Visual Computing Laboratory. URL http://vcl.iti.gr/.

[20] The Microsoft Kinect SDK. URL https://www.microsoft.com/en-us/download/details.aspx?id=44561.

[21] OpenCV. URL http://www.opencv.org.

[22] Microsoft Visual Studio. URL https://www.visualstudio.com/.