UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VICENTE PERUFFO MINOTTO

Audiovisual Voice Activity Detection and Localization of Simultaneous Speech Sources

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Claudio Rosito Jung
Advisor

Porto Alegre, July 2013

CIP – CATALOGING-IN-PUBLICATION

Peruffo Minotto, Vicente

Audiovisual Voice Activity Detection and Localization of Simultaneous Speech Sources / Vicente Peruffo Minotto. – Porto Alegre: PPGC da UFRGS, 2013.

111 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2013. Advisor: Claudio Rosito Jung.

1. Voice Activity Detection. 2. Sound Source Localization. 3. Multiple Speakers. 4. Competing Sources. 5. Multimodal Fusion. 6. Microphone Array. 7. Hidden Markov Model. 8. Support Vector Machine. 9. GPU Programming. I. Rosito Jung, Claudio. II. Título.

“Good things only fall apart so better things may fall together.”

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

HMM	Hidden Markov Model
SRP	Steered Response Power
SRP-PHAT	Steered Response Power with Phase Transform
GCF	Global Coherence Field
SVM	Support Vector Machine
SSL	Sound Source Localization
VAD	Voice Activity Detection
AVAD	Audio-based Voice Activity Detection
VVAD	Visual Voice Activity Detection
MVAD	Multimodal Voice Activity Detection
RGB-D	Red, Green, Blue - Depth
GPU	Graphics Processing Unit
CUDA	Computer Unified Device Architecture

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Given the tendency of creating interfaces between human and machines that increasingly allow simple ways of interaction, it is only natural that research effort is put into techniques that seek to simulate the most conventional mean of communication humans use: the speech. In the human auditory system, voice is automatically processed by the brain in an effortless and effective way, also commonly aided by visual cues, such as mouth movement and location of the speakers. This processing done by the brain includes two important components that speech-based communication require: Voice Activity Detection (VAD) and Sound Source Localization (SSL). Consequently, VAD and SSL also serve as mandatory preprocessing tools for high-end Human Computer Interface (HCI) applications in a computing environment, as the case of automatic speech recognition and speaker identification. However, VAD and SSL are still challenging problems when dealing with realistic acoustic scenarios, particularly in the presence of noise, reverberation and multiple simultaneous speakers. In this work we propose some approaches for tackling these problems using audiovisual information, both for the single source and the competing sources scenario, exploiting distinct ways of fusing the audio and video modalities. Our work also employs a microphone array for the audio processing, which allows the spatial information of the acoustic signals to be explored through the state-of-the art method Steered Response Power (SRP). As an additional consequence, a very fast GPU version of the SRP is developed, so that real-time processing is achieved. Our experiments show an average accuracy of 95% when performing VAD of up to three simultaneous speakers and an average error of 10cm when locating such speakers.

Keywords: Voice Activity Detection, Sound Source Localization, Multiple Speakers, Competing Sources, Multimodal Fusion, Microphone Array, Hidden Markov Model, Support Vector Machine, GPU Programming.

Detecção de Atividade de Voz e Localização de Fontes Sonoras Simultâneas utilizando Informações Audiovisuais

RESUMO

Em vista da tentência de se criarem intefaces entre humanos e máquinas que cada vez mais permitam meios simples de interação, é natural que sejam realizadas pesquisas em técnicas que procuram simular o meio mais convencional de comunicação que os humanos usam: a fala. No sistema auditivo humano, a voz é automaticamente processada pelo cérebro de modo efetivo e fácil, também comumente auxiliada por informações visuais, como movimentação labial e localizacão dos locutores. Este processamento realizado pelo cérebro inclui dois componentes importantes que a comunicação baseada em fala requere: Detecção de Atividade de Voz (*Voice Activity Detection* - VAD) e Localização de Fontes Sonoras (*Sound Source Localization* - SSL). Consequentemente, VAD e SSL também servem como ferramentas mandatórias de pré-processamento em aplicações de Interfaces Humano-Computador (*Human Computer Interface* - HCI), como no caso de reconhecimento automático de voz e identificação de locutor. Entretanto, VAD e SSL ainda são problemas desafiadores quando se lidando com cenários acústicos realísticos, particularmente na presença de ruído, reverberação e locutores simultâneos. Neste trabalho, são propostas abordagens para tratar tais problemas, para os casos de uma e múltiplas fontes sonoras, através do uso de informações audiovisuais, explorando-se variadas maneiras de se fundir as modalidades de áudio e vídeo. Este trabalho também emprega um arranjo de microfones para o processamento de som, o qual permite que as informações espaciais dos sinais acústicos sejam exploradas através do algoritmo estado-da-arte SRP (*Steered Response Power*). Por consequência adicional, uma eficiente implementação em GPU do SRP foi desenvolvida, possibilitando processamento em tempo real do algoritmo. Os experimentos realizados mostram uma acurácia média de 95% ao se efetuar VAD de até três locutores simultâneos, e um erro médio de 10cm ao se localizar tais locutores.

Palavras-chave: Detecção de Atividade de Voz, Lcalização de Fontes Sonoras, Múltiplos Locutores, Fusão Multimodal, Arranjo de Microfones, Modelo de Markov Oculto, Support Vector Machine, Programação em GPU.

1 INTRODUCTION

In most cases where a computer is used for a generic task, a mouse and a keyboard are employed as interfaces between the user and the machine. Despite being of easy use, they may not be adequate for a variety of applications, implying on the fact that other ways of human-machine interaction might be more promising (such an example would be the touchscreen in tablets and smartphones). Based on this and also in the frequent advances of technologies related to computing, it is only natural to exist high interests in developing ways of interaction that are similar to those of common use between humans. Particularly, speech represents a vast part of the information exchanged during those interactions (JAIMES; SEBE, 2007). Therefore, once computers are able to efficiently comprehend human-like communication, human-computer interaction (HCI) becomes more convenient and effective.

However, differently from human-human interaction, HCI still presents many challenges. As an example, in automatic speech recognition (ASR), which is one of the main branches of HCI (THIRAN; MARQUÉS; BOURLARD, 2010), it is necessary to recognize words in audio signals that may have been corrupted by external environment factors, such as noise, reverberation and other competing speech sources. Therefore, to compensate for these degradations, user-level systems require front-end techniques to function robustly. In this context, Voice Activity Detection (VAD) and Sound Source Localization (SSL) are two of the most important preprocessing tools in speech-based HCI. In VAD, the main goal is to distinguish segments of a signal that contain speech from those that do not, so that any processing effort may be focused only on information that contain speech. In SSL, the main idea is to explore the spatial information of the acoustic signals through microphone array beamforming techniques in order to enhance the speech of a source of interest while supressing those of competing sources and lowering environment noise (BRANDSTEIN; WARD, 2001).

Such benefits of VAD and SSL are actually unconsciously performed and availed day-by-day by the human brain in a effortless and effective way. The so called "cocktail-party" problem is a good example of such situation (BENESTY; CHEN; HUANG, 2008): in short, if many people are having parallel conversations in a party, it may be desirable to focus attention on one person among the many that are simultaneously speaking. Thanks to our brain's ability of processing auditive and visual informations (as lip movements) conjunctionally, we are able to easily comprehend a given person's speech. However, the same is not true for digitally automated systems, where existing algorithms tend to fail in such scenario. In most cases, VAD and SLL are approached for a single speech source case, what might not be appropriate for a number of situations. In applications such as multi-conferences, gaming scenarios, and also HCI, it is most often desirable, as in the cocktail-party example, to be able to distinguish between different speakers that might overlap their speeches. This ends up extending both VAD and SSL to more complex problems. Some recent works have been proposed for simultaneous speakers VAD (MARABOINA et al., 2006; BERTRAND; MOONEN, 2010; LORENZO-TRUEBA; HAMADA, 2010), and for simultaneous speakers SSL (DO; SILVERMAN, 2010; ZHANG; RAO, 2010). However, those approaches were based solely on acoustic signal processing techniques and some used large-aperture microphone arrays.

On the other hand, some other researches have approached VAD and SSL using both image and sound signal processing techniques (ASOH et al., 2004; BUTKO et al., 2008; ALMAJAI; MILNER, 2008; PETSATODIS; PNEVMATIKAKIS; BOUKIS, 2009). The main idea is that, similarly to as done by the brain, by fusing more than one modality of data it is possible explore the correlation between them in a way that if either the audio or video provide unreliable information, one may compensate for the other's flaws, making the algorithm more robust to adverse situations. However, in most audiovisual-based works, such as the ones mentioned above, it has been only approached the single sound source problem. In this work, we explore different ways of fusing the audio and video modalities to perform VAD and SSL of simultaneous speech sources. Our work employs a microphone array for the audio processing, which allows us to exploit spatial information of the speech sources (process similar to what the human ears do), and a color camera, which allows visual information such as lip movements to be used. We aim, in general, at multi-user HCI systems, such as videoconferencing and gaming scenarios.

This dissertation is presented as a collection of previously written articles that have already been published (BLAUTH et al., 2012; MINOTTO et al., 2013) or that are under revision process. The referred works have dealt with VAD and SSL of one and multiple speakers in different ways. We have explored distinct methods for fusing audio and video data, achieving above 90% VAD accuracy in all works. Additionally, as a consequence of the necessity for real-time processing of HCI applications, we have also developed an efficient GPU version of the Steered Response Power (SRP), which is a key audio processing technique (MINOTTO et al., 2012) used in our algorithms.

For better understanding of the chronological progress of our works through the mentioned papers, Section 1.1 is devoted to explaining our contributions in each article, individually, as well as their interconnections.

1.1 Overview of the Papers

As mentioned, this dissertation has been developed as a collection of papers. Chapters 3 through 6 represent articles that have been published or are under revision process. In fact, we present three of the five published papers during this Masters course and an extra one that has already been submitted. The main contributions of the referred articles summarize to VAD and SSL of single and simultaneous speakers, and GPU implementations of the SRP-PHAT (SRP with Phase Transform) and the Cubic Splines Interpolation algorithms.

- Chapter 3 (BLAUTH; MINOTTO et al., 2012): we present an approach that performs single speaker VAD using a audio information only (video is included for SSL). We have developed a Hidden Markov Model (HMM) competition scheme, through which VAD is performed by analyzing the output of the SRP-PHAT microphone array beamforming technique. The SRP-PHAT is mainly an SSL algorithm, and is known to be robust in realistic conditions. We extend it to a VAD algorithm by assessing the spatio-temporal behavior of the dominating sound source against two HMMs, one that models speech situations and one that models silence. The dominant speaker is classified as active or inactive by comparing the output of both HMMs. For the SSL part of the algorithm, video cues are included from the results of a face-tracking algorithm. The output of the SRP-PHAT is weighted based on the location of the tracked faces, and a third speech-related HMM is used to produce the location of the main speech source. This work is a result of many collaborators, so we highlight that our main contribution is related to the development of the HMM competition scheme used for performing VAD, which is later extended in our other works.

- Chapter 4: in this article (MINOTTO et al., 2013), we expand the previously mentioned one to a multimodal technique. We combine our audio-based approach (BLAUTH; MINOTTO et al., 2012) to the video-based approach of (LOPES et al., 2011) through a decision fusion scheme. We study many supervised classification algorithms for combining the results of the individual unimodal classifiers. The well known Machine Learning software Weka (HALL et al., 2009) is used for exploring a variety of approaches, through which it is concluded that a C4.5 decision tree (QUINLAN, 1993) presents the best benefits in this scenario (trade-off between accuracy and speed, and also robust against overfitting). As another contribution of the work, we also analyze the robustness of our approach to adverse situations (intentionally generated), confirming that one modality in fact robustly compensates for the other's flaws by using the proposed fusion approach.

- Chapter 5: a multimodal approach for simultaneous speakers VAD and SSL is developed. In this work, we extend the ideas from both previously mentioned papers. The HMM competition scheme is reformulated in order to deal with multiple speakers in the scene. An optical-flow algorithm is used to assess lip movements of each user, which generates visual features as inputs to a Support Vector Machine (SVM) classifier. The SVM outputs a video-based VAD probability for each potential speaker, and the audio modality is processed by the multi-user HMM competition scheme, at which the video probability is incorporated. This characterizes the combination of both modalities, and is considered a mid-fusion approach, since the output of a classifier is applied to the input of another one. The final VAD decision is performed by analyzing the competing models, and its results are also reutilized for generating a final SSL position (recalling the HMM scheme uses the SRP-PHAT SSL method).

- Chapter 6 (MINOTTO et al., 2012): this chapter describes the GPU implementations we have developed to achieve real-time processing of our multimodal systems. As it may be observed, all our approaches employ the SRP-PHAT algorithm through a microphone array. Despite its known robustness against noise and reverberations situations, the SRP-PHAT has a high computational cost as a downside. For this reason, we have developed two Compute Unified Device Architecture (CUDA) versions of the algorithm, as well as one for the Cubic Splines Interpolation, which is commonly applied as a part of the SRP-PHAT itself. Using such implementations we are then able to achieve real-time processing in our previously mentioned VAD/SSL approaches.

From the referred papers we may notice the accomplished evolution of our techniques, from a single-speaker unimodal work to a multiple speakers multimodal VAD and SSL algorithm. Incorporating extra modalities of data into our methods was a natural path to be taken, given that the more realistic the scenario is, the more complex the problem becomes. While auditive information is more robust for performing VAD, visual information shows to indeed strengthen the audio modality (and enhance SSL) when an adequate fusion technique is applied, which by itself is a wide field of research (ATREY et al., 2010; THIRAN; MARQUÉS; BOURLARD, 2010). For this reason, we also consider the proposed fusion approaches as contributions of this dissertation (Section 2.5 provides review on the main aspects of multimodal fusion). More precisely, in Chapter 4 a decision fusion approach is employed (two unimodal classifiers are combined into a final multimodal one). In Chapter 5 a mid-fusion technique is used (the output of the video classifier is combined with the input of the audio one). And for a feature-fusion approach, in our ongoing/future work, we have been combining multiple features from both modalities into a single multimodal classifier. Preliminary results of this new approach also show to be promising by reaching an average VAD accuracy of 95% of up to three simultaneous speakers.

2 THEORY REVIEW

In this chapter we present some of the theory required for the comprehension of our proposed works. Given this dissertation is organized as a collection of previously produced papers, the theoretical overview of this chapter is restricted to the topics that were not covered in great details in the articles. In Sections 2.1 through 2.4 we unify some concepts regarding microphone arrays that are not common to all papers. We also explain some of the main aspects related to multimodal analysis in Section 2.5.

2.1 Microphone Arrays and the Time Difference of Arrival

A Microphone Array consists of a set of microphones properly positioned in a way that it becomes possible the extraction of spatial information from the captured acoustic signals (BENESTY; CHEN; HUANG, 2008). This ability is mainly explored for the tasks of speech enhancement and SSL, which are techniques that have been applied in the past for many practical purposes, such as videoconferencing (KELLERMANN, 1991; CHU, 1995; WANG; CHU, 1997), ASR (HUGHES et al., 1998, 1999; WEINSTEIN et al., 2004), military practices (SILVERMAN; PATTERSON W.R.; FLANAGAN, 1999), and echo cancellation (JOHNSON; DUDGEON, 1993a). Currently, some work on those areas are still being developed (BENESTY; CHEN; HUANG, 2008), but with a largest attention on techniques related to HCI, as in the focus of this work. For purpose of illustration, Figure 2.1 shows an example of an eight-elements microphone array, which is also the one that is going to be used for this work.

When an array of microphones is used as an acoustic capture system one is able to efficiently exploit the spatiality of the incoming sound waves, what is not achievable with only one microphone. This is possible due to something called Time Difference of Arrival (TDOA) that is provided by microphone arrays. The TDOA, thus, represents the difference in time that sound waves carrying same information take to travel from on microphone to another. In other words, it is the delay between two microphones receiving the same data. This is a consequence of the physical distance between the microphones, and is considered the main property of a microphone array that signal processing algorithms may avail from. Figure 2.1 helps the description of this process.

In the illustration, M is the number of microphones in the array system, and $r_m^{\mathbf{q}}$ is the distance from microphone m to the sound source located in $\mathbf{q}$, for $m = 1, 2, ..., M$. Therefore, we may denote $\tau_m^{\mathbf{q}}$ as being the travel time of a sound wave originated in $\mathbf{q}$ and propagated towards the microphone m:

$$\tau_m^{\mathbf{q}} = \frac{r_m^{\mathbf{q}}}{c} \tag{2.1}$$

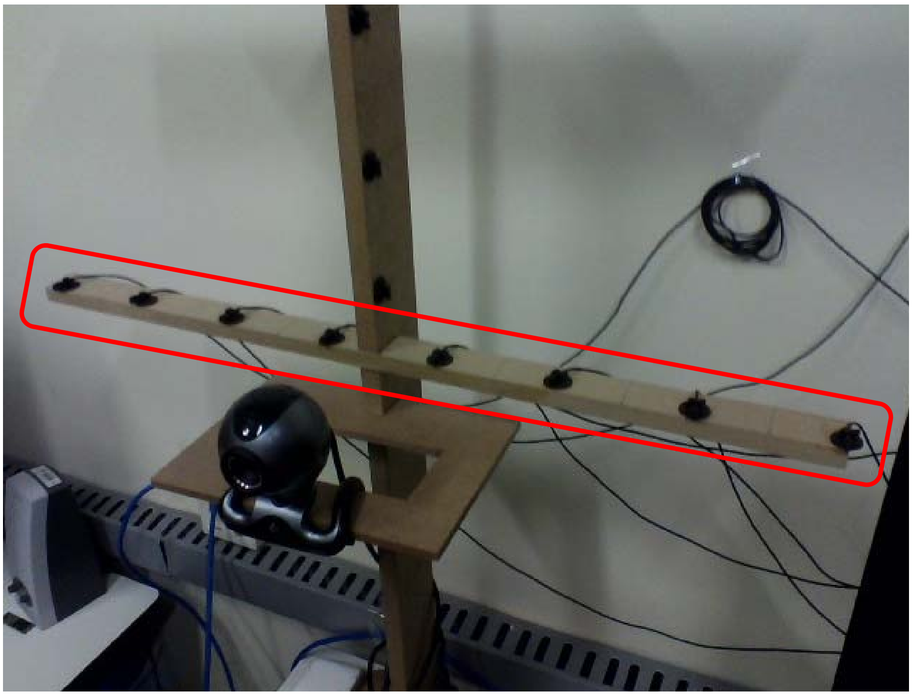


Figure 2.1: Example of a microphone array system.

where c represents the speed of sound $(343m/s)$, $\tau_m^{\mathbf{q}}$ is given in seconds, and $r_m^{\mathbf{q}}$ may be established using the simple Pythagorean theorem, since the position of the sound source and microphones (in this example) are known. From this equation, then, it is possible to define the TDOA between microphones m and l and a point $\mathbf{q}$ as being

$$\tau_{ml}^{\mathbf{q}} = \tau_m^{\mathbf{q}} - \tau_l^{\mathbf{q}} = \frac{r_m^{\mathbf{q}} - r_l^{\mathbf{q}}}{c} \tag{2.2}$$

From this definition of the TDOA, most techniques involving microphone arrays are derived. In the following sections we focus on the most important ones for the understanding of this work.

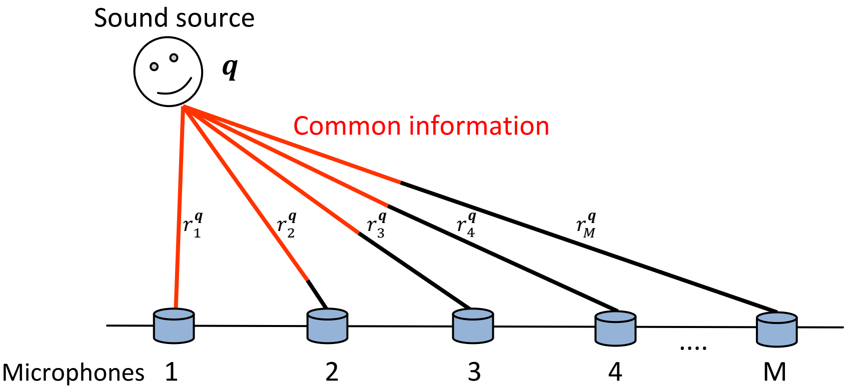


Figure 2.2: TDOA: Common information being received at different moments by the microphones.

2.2 Sound Wave Propagation

In this section we describe a model for the sound wave propagation that is consistent and suited for this work's practices. For that, we divided this section into two different topics: firstly it is explained assumptions that simplify the physical model for acoustic waves propagation, and secondly it is presented the mathematical equations that describe the sound signals emission and acquisition.

2.2.1 Simplified Acoustic Model

A sound source, either a human or a mechanic transducer are not ideal radiators of spherical waves (form of the acoustic wave). Additionally, in realist environments, sound waves are emitted with some degree of directionality, what is imposed by sound source, and suffer from spatial attenuation. Beyond that, they suffer from phenomena such as diffraction, reverberation, and also depend upon the medium the are being propagated through. For those reasons, for the sound wave model to be precisely correct, it would need a complex mathematical formulation dependent upon many variables. However, it has been already studied and experimented a simplified model that has been shown to be adequate enough for practical implementations (JOHNSON; DUDGEON, 1993a; BRANDSTEIN; WARD, 2001; BENESTY; CHEN; HUANG, 2008). The following assumptions summarize this model.

- **The sound source are modeled as points**. This simplifies the complex radiation patterns of human head models into a spherical wave propagation model.
- **The propagation medium is homogeneous**, which guarantees that the speed of sound c is constant everywhere. This implies that the acoustic propagation is non-refractive.
- **The medium is lossless**. This ensures that the sound waves do not lose energy during propagation.
- **The Doppler effect is negligible**. The sound signals' frequencies do not change if its source is moving.

Note that in practice c may change as a function of the room temperature. For that, it is taken into account that this does not occur during the course of a experiment, so that the same parametrization of c may be used (343.3 m/s). Additionally, for c to have a significant change, the temperature would also have to drastically change.

2.2.2 Acoustic Signals Mathematical Description

The mathematical definition of the microphone digital signals and sound source propagating signals is necessary for later describing all algorithms related the microphone arrays. Therefore, we may define the signal received by microphone m at time t as being

$$x_m(t) = s(t) + n_m(t) \tag{2.3}$$

where $s(t)$ is the sound wave emitted by the source, and $n_m(t)$ is the term due to noise present in each microphone m (still, $m = 1, 2, ..., M$). Note that this term is based in the microphones indexes, i.e., does not depend upon $s(t)$. Thus, it is a result of individual stochastic processes individual to each microphone (BENESTY; CHEN; HUANG, 2008), which is an important factor exploited by all the SSL algorithms that will be further presented.

It may be observed that Eq. (2.3) does not explicitly present a term describing environmental reverberations. According to Johnson and Dudgeon (JOHNSON; DUDGEON, 1993a), it is commonly omitted, since reverberations, differently from noises, depend on $s(t)$. Therefore, for sakes of simplicity, it is assumed that $s(t)$ embeds the reverberation information too.

Once the equation of the mics. received signals is described, it is then possible to derive them to a more specific version that is more suited for describing microphone array techniques (DIBIASE, 2000). Recalling Eq. (2.2) that describes the TDOA of a pair of microphones ml and a point $\mathbf{q}$ in space, we may incorporate it into $x_m(t)$, and describe the signal received by each microphone as a function of not only t, but $\mathbf{q}$ too:

$$x_m(t, \mathbf{q}) = s(t - \tau_{am}^{\mathbf{q}}) + n_m(t) \tag{2.4}$$

where a is a constant that represents a reference microphone, and $\tau_{am}^{\mathbf{q}}$ is the TDOA for between microphone m and the reference one.

Generally, a is taken to represent the microphone most distant from the sound source $\mathbf{q}$. This is convenient because, since $\tau_a^{\mathbf{q}} - \tau_m^{\mathbf{q}}$ is never negative, the function is casual, what is something needed for practical systems (DIBIASE, 2000). In other words, Eq. (2.4) shows that the signal of a microphone m corresponds to a time-advanced version of the signal of the reference microphone a. We may also notice that the delay term is not inserted in $n_m(t)$, due to the fact that the noise of different microphones are uncorrelated, thus, in practice, it is irrelevant to insert $\tau_{am}^{\mathbf{q}}$ into $n_m(t)$.

2.3 Beamforming

When one wants to perform tasks such as speech enhancement and/or SSL, it is often used algorithms based on beamforming. This class of techniques consist in virtually focusing the array of microphones in capturing the signals originating from a specific location in space, in such a way that informations originated elsewhere are attenuated. This process is often called spatial filtering (BENESTY; CHEN; HUANG, 2008) and its output is a signal often called the *steered response* of the beamforming process.

In Figure 2.3 an example of beamforming is given. A sound source of interest (a person), inside a room, emits sound waves (speech) that are corrupted by interfering noise sources. Besides the noise, the person's speech also suffer from reverberation caused by spatial aspects in the room, such as objects and walls. This picture shows a completely adverse (realistic) scenario where it may be hard to locate the sound source of interest due to the interfering noise and reverberations. However, using beamforming techniques, one could still be able to locate it and apply, for example speech enhancement algorithm to that source (BRANDSTEIN; WARD, 2001). Nevertheless, the precision with which the source is located is dependent upon the technique that is used (in Section 2.4.2 we present the SRP-PHAT - a state-of-the-art one).

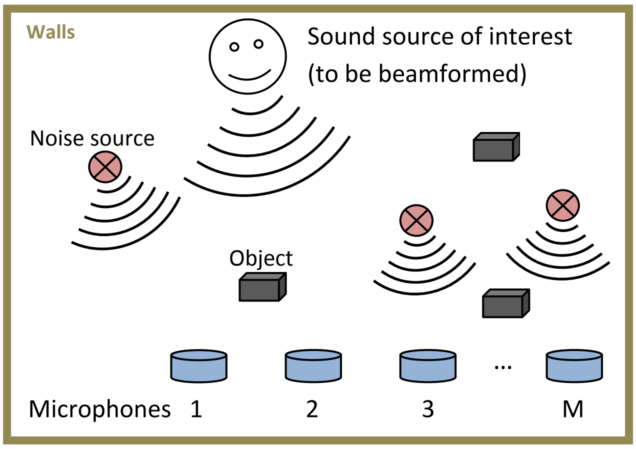


Figure 2.3: Example of a situation where spatial filtering is useful.

2.3.1 Delay-and-Sum

As explained before, beamforming is the process of finding the steered response of a certain point in space. For this purpose, different algorithms have been proposed (BENESTY; CHEN; HUANG, 2008), however, among them the delay-and-sum technique stands out due to being very simple and yet efficient.

As its name suggests, the delay-and-sum, in short, consists of applying delays to each microphone signal and summing them. Specifically, those delays (related to the TDOAs) are applied in order to compensate for the misalignment between the signals $x_m(t)$ (as explained in 2.1). Once each signal is time-aligned, they are summed to form a enhanced version of the original signals. Figure 2.3.1 illustrates this process, where $y(t)$ represents the steered response, i.e., the output of the delay-and-sum algorithm.

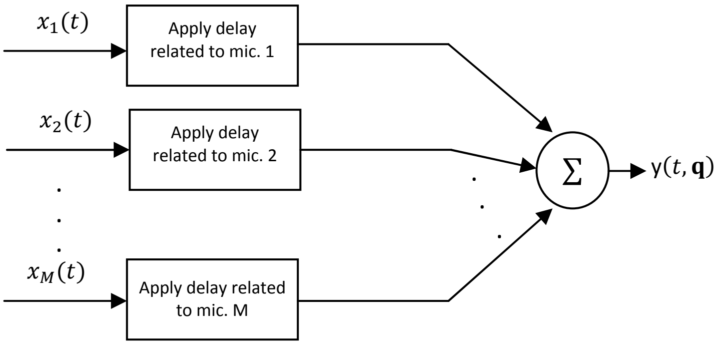

$x_1(t)$
Apply delay related to mic. 1
$x_2(t)$
Apply delay related to mic. 2
$x_M(t)$
Apply delay related to mic. M
Σ
y(t, **q**)

Figure 2.4: Delay-and-sum algorithm represented schematically.

Translating this schematic representation of the delay-and-sum into its mathematical representation, we may define it through the following equation (for an array of M microphones).

$$y(t, \mathbf{q}) = \sum_{m=1}^{M} x_m(t - \tau_{am}^{\mathbf{q}}) \tag{2.5}$$

Applying the Fourier Transform, it may be computed in the frequency domain as

$$Y(\omega, \mathbf{q}) = \sum_{m=1}^{M} X_m(\omega) e^{-j\omega\tau_{am}^{\mathbf{q}}} \tag{2.6}$$

where ω is the frequency parameter and the delay term is now represented as a complex exponential.

Analyzing the algorithm, we may note that $y(t, \mathbf{q})$ is not only a function of time, but also a function of a position $\mathbf{q}$ in space. From that, it may be concluded that, if $\mathbf{q}$ is in fact the position of the speaker, $y(t, \mathbf{q})$ will represent its speech in an enhanced form. Note that this happens as a result to the fact that the noise $n_m(t)$, as stated in Subsection 2.2.2, is uncorrelated between different microphones, making them sum in a destructive manner, no matter what value of $\mathbf{q}$ is passed as parameter. Consequently, the opposite happens to the speech signal $s_m(t)$ of the sound source: it will sum in a constructive manner, thus, causing the speech to be enhanced. On the other hand, if $\mathbf{q}$ is not the position of the speaker, $y(t, \mathbf{q})$ will represent an attenuated a and poorly audible version of the person's speech, because the whole signal (not only the noise) will sum in a destructive way.

To better visualize this process of signal attenuation and enhancement, we may observe Figure 2.3.1, where the signal of the fourth microphone is taken as reference ($a = 4$), and the delay-and-sum algorithm is applied. In this example, each signal individually received additive white Gaussian noise in order to simulate $n_m(t)$, and the sinusoid segment simulates the delayed common information between the microphones. We may observe that, as explained, the signal's information of interest was enhanced, while the noise was attenuated. Once again, this happens because the location that was beamformed contained a speaker.

2.3.2 Filter-and-Sum

The filter-and-sum algorithm may be seen as a more general case of the delay-and-sum. The algorithm's idea is very simple: it is the delay-and-sum with filtering. More specifically, in between its delaying and summation steps, there is a filtering process applied individually to each signal $x_m(t)$. Recalling the frequency-domain definition of the delay-and-sum in Eq. (2.6), we may define the filter-and-sum, also in the frequency-domain, by the following expression.

$$Y(\omega, \mathbf{q}) = \sum_{m=1}^{M} W_m(\omega) X_m(\omega) e^{-j\omega\tau_{am}^{\mathbf{q}}} \tag{2.7}$$

where $W_m(\omega)$ is a generic term representing a filter, $X_m(\omega)$ remains the Fourier Transform of $x_m(t)$, and $e^{-j\omega\tau_{am}^{\mathbf{q}}}$ is the delay term. The choice of an adequate filter is dependent upon the characteristics of the environment and the sound source (BRANDSTEIN; WARD, 2001), and may be chosen among many different already studied approaches. A good overview of some of those approaches may be found in (KWAK; KIM, 2008). However, in practice, the most famous one was proposed in (DIBIASE, 2000) for the purpose of sound source localization and will be further explained in Section 2.4.2.

2.4 Sound Source Localization

As seen in Figure 2.3.1, performing the beamforming of a location which contains a speaker will cause the output signal to have a high amount of energy, whereas of a

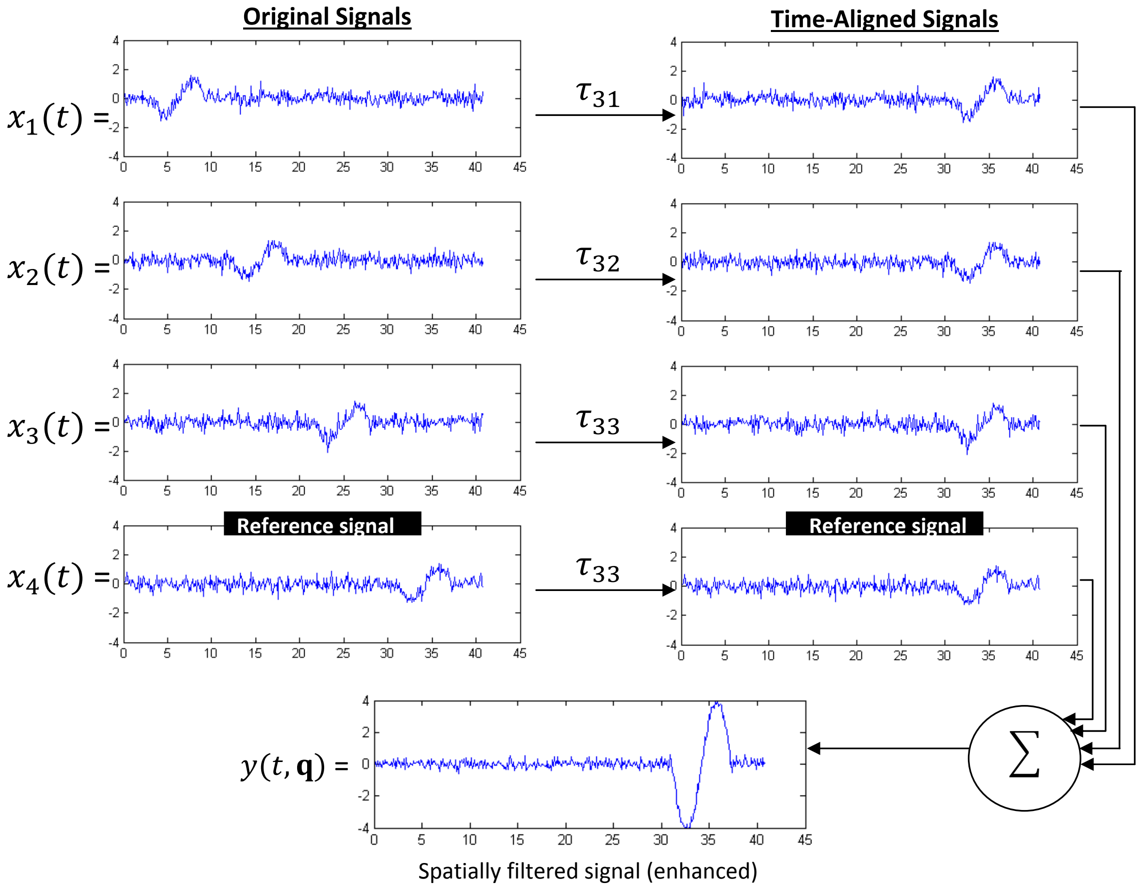


Figure 2.5: Example of delay-and-sum of a location containing an active sound source of interest (speaker).

position which does not contain a speaker will output a signal with low energy. This is an important aspect that is deeply exploited in the SSL algorithms. In other words, the SSL algorithms through beamforming are an extension, in some way, of the delay-and-sum and/or filter-and-sum techniques.

Next subsections explain how such extension is done by describing how the SRP-PHAT algorithm is derived from a filter-and-sum. We also show its two possible formulations, one in the frequency-domain, and another one in the time-domain.

2.4.1 Steered Response Power

As means of exploiting the energy characteristics of the output signal of a beamformed position, it is possible to compute the steered response power (SRP) of that position. The term SRP, as the name suggests, computes the power of a steered response signal, i.e., the energy of the output of a beamforming algorithm. The SRP may be expressed in terms of a point $\mathbf{q}$, through the following expression.

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} |y(t, \mathbf{q})|^2 \, dt \tag{2.8}$$

However, it is more interesting to represent it using the frequency-domain definition of $y(t, \mathbf{q})$. This may be done using Parseval's theorem, which states that the total energy contained in a waveform summed across all of time t is equal to the total energy of the waveform's Fourier Transform summed across all of its frequency components ω. Therefore, the SRP may be represented in the frequency-domain.

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} |Y(\omega, \mathbf{q})|^2 \, d\omega \tag{2.9}$$

Substituting $Y(\omega, \mathbf{q})$ by its complete expression, we get the expanded version of the SRP of a filter-and-sum:

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} \left| \sum_{m=0}^{M-1} W_m(\omega) X_m(\omega) e^{-j\omega\tau_{am}^{\mathbf{q}}} \right|^2 d\omega \tag{2.10}$$

where $W_m(\omega)$ still represents a filtering function, and will be further explained in Subsection 2.4.2.

Now that it is possible to measure the acoustic energy of a given point $\mathbf{q}$ in space, we may derive this to a sound source localization algorithm by simply recalling that a point which contains an active speaker will emanate a larger amount of acoustic energy than points which do not contain an active speaker. Therefore, given a set of candidate sound source locations, the one which potentially represents a speaker position will have the highest SRP among the set. Mathematically, we can describe the estimate $\hat{\mathbf{q}}$ of the sound source location as being

$$\hat{\mathbf{q}} = \underset{\mathbf{q} \in \mathcal{Q}}{\operatorname{argmax}} \, P(\mathbf{q}), \tag{2.11}$$

where $\mathcal{Q}$ is an user-defined set of real-world coordinates to be scanned. Generally, it is chosen in a way that its points compose a geometric form such as a line, square or cube (depending on the dimensionality of the search process).

2.4.2 Steered Response Power with Phase Transform

When the filtering function $W_m(\omega)$ is chosen to be the *phase transform* (PHAT) function, the SRP becomes the SRP-PHAT. This approach was proposed in (DIBIASE, 2000) and is still known as a state-of-the-art technique for SSL in noisy and reverberant environments.

The PHAT filter may be defined in the following way.

$$W_m(\omega) = \frac{1}{|X_m(\omega)|} \tag{2.12}$$

Therefore, substituting Eq. (2.12) into Eq. (2.9), the SRP becomes the SRP-PHAT:

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} \left| \sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{-j\omega\tau_{am}^{\mathbf{q}}} \right|^2 d\omega. \tag{2.13}$$

It may be observed that the effect of the PHAT filter is to simply remove the magnitude information of the sound signal, leaving only the phase information. In other words, it equally weights all the frequency components of the signal during the calculation of the SRP-PHAT, which is highly advantageous because it removes the contribution that noise and reverberation would have for the SRP. Moreover, since the TDOA information (delay) is only contained in the phase term of the signals, it is only normal that using the amplitude would increase the SRP-PHAT outputs based on the loudness of the sound and not on its delay information (which would enhance noise sources instead).

Now for the estimation of the sound source location, the same step described in Eq. (2.11) still holds. It summarizes in finding the position that generates the largest SRP-PHAT value among the set $\mathcal{Q}$ of candidate locations. To better visualize this process, Figure 2.4.2 illustrates the scanning process over an arbitrary $\mathcal{Q}$.

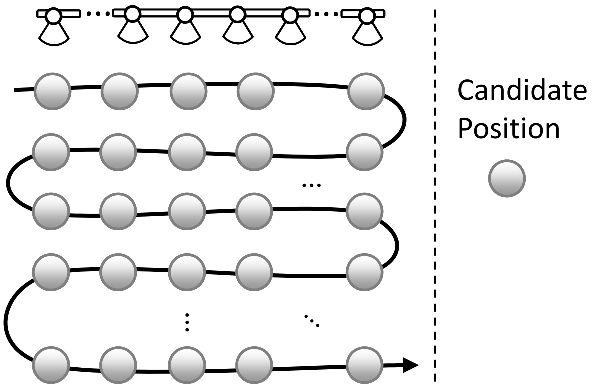


Figure 2.6: Example of the SRP-PHAT being used for SSL.

2.4.3 Alternate Formulation of the SRP-PHAT

The SRP-PHAT allows another practical implementation through which a lower computational complexity may be achieved at the cost of less localization accuracy. The approach described in Section 2.4.2 is entirely computed in the frequency-domain, so we refer to it as the frequency-domain version. The one described in this section has some steps done in the time-domain, so we refer to it as the time-domain version. These terms are extensively referred in Chapter 6 where we propose GPU implementations of both versions.

The energy of a signal in the frequency domain was described through Eq. (2.9). This equation may also be equivalently represented as,

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} Y(\omega, \mathbf{q}) Y^*(\omega, \mathbf{q}) d\omega \tag{2.14}$$

where * denotes the complex conjugate operator.

Substituting Eq. (2.7) in Eq. (2.14) and again using the PHAT filter, we obtain

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} \left(\sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{-j\omega\tau_{am}^{\mathbf{q}}} d\omega \right) \left(\sum_{l=1}^{M} \frac{X_l^*(\omega)}{|X_l^*(\omega)|} e^{j\omega\tau_{al}^{\mathbf{q}}} d\omega \right). \tag{2.15}$$

Rearranging this expression yields

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} \sum_{m=1}^{M} \sum_{l=1}^{M} \left(\frac{X_m(\omega)}{|X_m(\omega)|} e^{-j\omega\tau_{am}^{\mathbf{q}}} d\omega \right) \left(\frac{X_l^*(\omega)}{|X_l^*(\omega)|} e^{j\omega\tau_{al}^{\mathbf{q}}} d\omega \right). \tag{2.16}$$

Through Eq. (2.2) we may notice that

$$\tau_{al}^{\mathbf{q}} - \tau_{am}^{\mathbf{q}} = \tau_{m}^{\mathbf{q}} - \tau_{l}^{\mathbf{q}} = \tau_{ml}^{\mathbf{q}}. \tag{2.17}$$

Using such notion and organizing the multiplications in Eq. (2.16) we obtain

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} \sum_{m=1}^{M} \sum_{l=1}^{M} \frac{X_m(\omega)X_l^*(\omega)}{|X_m(\omega)X_l^*(\omega)|} e^{j\omega\tau_{ml}^{\mathbf{q}}} d\omega. \tag{2.18}$$

Since the SRP-PHAT of pair ml is the same as for pair lm, it is possible to reduce the number of iterations in the second summation. Furthermore, it is also possible to interchange the order of the integral and the summations, given that in practice the microphones' signals and filters contain finite energy. This way, we obtain the following expression for the SRP-PHAT.

$$P(\mathbf{q}) = \sum_{m=1}^{M} \sum_{l=m+1}^{M} \int_{-\infty}^{\infty} \frac{X_m(\omega)X_l^*(\omega)}{|X_m(\omega)X_l^*(\omega)|} e^{j\omega\tau_{ml}^{\mathbf{q}}} d\omega. \tag{2.19}$$

This integral in Eq. (2.19) is also know as the Generalized-Cross-Correlation with the Phase Transform (GCC-PHAT) of a microphone pair ml. It may be solved through the Inverse Fourier Transform (instead of an iteration process over the integral) to obtain its corresponding time-domain representation $R_{ml}(\tau)$. Therefore, we may describe this alternate formulation of the SRP-PHAT in terms of the summation of the GCC-PHATs of all unique microphone pairs:

$$P(\mathbf{q}) = \sum_{m=1}^{M} \sum_{l=m+1}^{M} R_{ml}(\tau_{ml}^{\mathbf{q}}). \tag{2.20}$$

This formulation yields a faster processing speed, for the integrals may be solved once for each pair of microphones prior to the search-space scanning step, but introduces an imprecision during the delaying process of the signals, which happens due to the truncation of $\tau_{ml}^{\mathbf{q}}$ when $R_{ml}(\tau_{ml}^{\mathbf{q}})$ is accessed. These topics are covered in more details in Chapter 6.

2.5 Multimodal Fusion

Applications that explore more than one modality of data have recently become more common in researches that deal with multimedia analysis tasks (ATREY et al., 2010). The combination of multiple modalities through a proper multimodal fusion approach may be highly beneficial for problems that can have multiple input streams of different nature. Such examples may include audiovisual speaker detection, human tracking, event detection, etc. In particular, our work deals with speakers detection/localization which implies in the use of video and audio as the input modalities. Therefore, the idea is that in situations where the audio might be corrupted by noise and/or reverberation, visual cues can compensate for it, making the final algorithm more robust. Conversely, if the visual modality is unreliable (as in moments where the users are too distant from the camera), auditive features may strengthen the final algorithm. Such benefits resulting from a multimodal fusion approach, however, demand that a more complex analysis of the input data is performed. For this reason some practical difficulties emerge in a multimodal processing scenario:

1. The media streams are asynchronous in most cases, which means their processing time are dissimilar. This may require a preprocessing synchronization step to be applied or require the fusion approach to deal with such problem.

2. The modalities may present varying confidences according to the scenario at which they are captured. As in the previous example, the video modality may be less reliable than the audio if the users are standing far away from the camera. Such characteristic may require some weighting approach of the streams.

3. What information (features) should be extracted from each modality. Audio-based and video-based applications, as in this work for example, may use a variety of algorithms for extracting speech related features for the classification process.

4. How the features extracted from each modality should be fused. Multimodal fusion may happen in different levels, such as at the feature level, the decision level or as a hybrid approach between those two. For this reason the combination approach is also an important aspect of multimodal analysis that should be chosen carefully according to the characteristics of the problem.

In our works, the first topic is basically covered through block-based processing of the audio modality, and for this reason, is not addressed during the multimodal fusion approaches. In more details, block-based processing of the audio means simply taking buffers composed of many audio samples as input for each image of the video modality. Since the audio sampling frequency is much higher than the video's (44100Hz vs 20fps, in our systems) we may synchronize both modalities by processing, e.g., 2205 audio samples for each input image.

The second topic is also something that may or may not be handled in the fusion approach. One case of modality weighting would be by doing it implicitly through an automated training process of a supervised classifier where failure cases (as those mentioned) are included in the training dataset, so that the classifier learns how to address such adverse situations. Another approach would be to assess the modality's reliability through some additional information extracted from its corresponding input stream and explicitly weight that modality's contribution to the fusion approach. One example of the first case is given in Chapter 4 where adverse situations are handled by a decision tree classifier; and the second case is approached in Chapter 5 by reducing the contribution of the visual cues based on the distance of the users from the camera.

The third topic is rather generalized to be categorized into different approaches. From the audio modality, for example, many different features may be extracted for speech detection purposes. Some examples include energy and entropy of the signal (LEE; MUHKERJEE, 2010a), coefficients of the Discrete Cosine Transform of the signals (GAZOR; ZHANG, 2003), and higher order statistics of the signal's linear predictive coding (LPC) residuals (NEMER; GOUBRAN; MAHMOUD, 2005). For the video modality, information regarding the user's mouth is usually extracted to explore its movement across time. For this purpose, some works, for example, propose the use of optical-flow (GURBAN; THIRAN, 2006; AUBREY; HICKS; CHAMBERS, 2010; TIAWONGSOMBAT et al., 2012), shape information (AUBREY et al., 2007), and color information (SCOTT et al., 2009; LOPES et al., 2011). Given such variety of methods for extracting information from the input modalities, the choice of the algorithm to use should be based upon what best suits the multimodal scenario. Sometimes, one of the influencing factors may also be the type of capture hardware available in the system, what may restrict or broaden the range of employable techniques.

The fourth matter related to multimodal analysis we raised is also very important for the final robustness of an algorithm. As mentioned, multiple modalities may be fused using many different approaches. To better understand the most common ones of the literature, Figure 2.7 illustrates early fusion (or feature-level fusion), mid fusion, and late fusion (or decision-level fusion). The first case, of early fusion, happens when the features of the modalities (in our case, audio and video) are unified into a larger set of features to be classified by some sort of classifier (either supervised or not). The third case, summarizes in individually classifying the features of each modality to later combine the separate results through a third classification technique. The second case is an intermediate approach between the previous others: the result of a unimodal classifier is combined with the input features of the other modality; the modified features are then classified by a final and unimodal method, and we name this approach as mid fusion. An example of mid-fusion is given in Chapter 5, where we propose to fuse the output of an SVM-based lip motion analyzer to the input of an HMM-based audio approach. In Chapter 4 a late-fusion method is proposed, where we use a decision tree technique to combine our algorithm of Chapter 3 to the work of (LOPES et al., 2011).

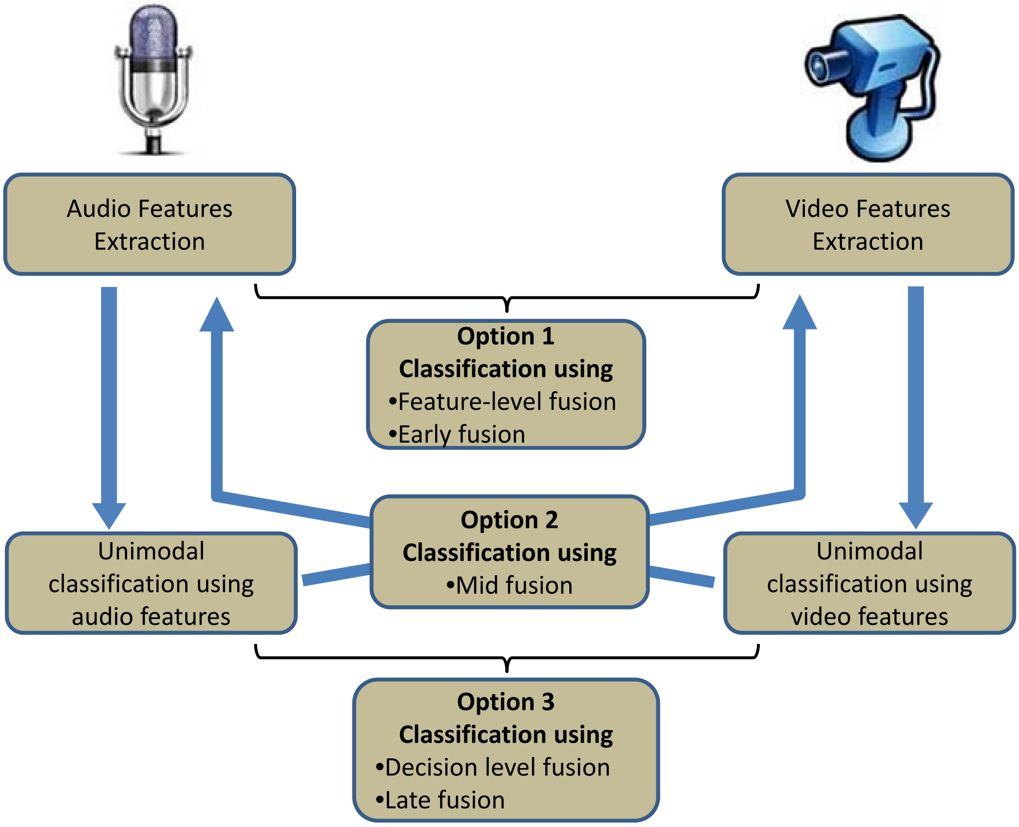


Figure 2.7: Three possible ways of performing multimodal fusion.

3 VOICE ACTIVITY DETECTION AND SPEAKER LOCALIZATION USING AUDIOVISUAL CUES

Dante A. Blauth, Vicente P. Minotto, Claudio R. Jung, Bowon Lee, Ton Kalker, Voice activity detection and speaker localization using audiovisual cues, Pattern Recognition Letters, Volume 33, Issue 4, March 2012, Pages 373-380, ISSN 0167-8655,
Keywords: User interfaces; Voice activity detection; Speaker localization; Multimodal analysis; Hidden Markov Models

Abstract

This paer proposes a multimodal approach to distinguish silence from speech situations, and to identify the location of the active speaker in the latter case. In our approach, a video camera is used to track the faces of the participants, and a microphone array is used to estimate the Sound Source Location (SSL) using the Steered Response Power with the phase transform (SRP-PHAT) method. The audiovisual cues are combined, and two competing Hidden Markov Models (HMMs) are used to detect silence or the presence of a person speaking. If speech is detected, the corresponding HMM also provides the spatio-temporally coherent location of the speaker. Experimental results show that incorporating the HMM improves the results over the unimodal SRP-PHAT, and the inclusion of video cues provides even further improvements.

3.1 Introduction

Nowadays, keyboard and mouse are the most popular devices for Human Computer Interaction (HCI), adopted by the vast majority of personal computers. Despite being intuitive and easy to use, they may not be adequate in a variety of applications. For instance, the manual annotation of multimedia data (photos, videos, and music clips, to name a few) into a set of tags using keyboard and mouse is a tiresome task, and other ways of interaction (such as audiovisual data) seem to be more natural. In particular, the analysis of speech and facial features appear to be promising in the development of multimodal HCI systems (JAIMES; SEBE, 2007).

There are several challenges when exploring facial cues and speech data to develop HCI systems. Firstly, the face must be detected and tracked robustly in time, which is a complex task in the presence of partial occlusions, head tilts and turns. Secondly, the audio analysis (mainly speech detection) is highly corrupted by background noise (e.g. an air-conditioning system), so that it is necessary to detect when a person is speaking or not (this problem is usually referred to as Voice Activity Detection - VAD). Finally, when more than one user is captured by the camera, it is important to determine which person is actually interacting with the computer (in the case of voice commands, that means finding the active speaker at a given time).

This chapter presents a new approach to estimate the location of the active speaker based on audiovisual cues. We propose a Hidden Markov Model (HMM) that characterizes the expected spatio-temporal properties of a typical speaker considering the input captured by an array of microphones, and its extension to include visual cues. This HMM imposes spatio-temporal constraints on the location of the active speaker, improving the results of audio-only localization. Another HMM to model silence periods is also presented, so that VAD can also be achieved by comparing the speech and silence HMMs.

The remainder of this chapter is organized as follows: Section 3.2 presents some related work, and the proposed approach is described in Section 3.3. Some experimental results are provided in Section 3.4, and the conclusions are drawn in Section 3.5.

3.2 Related Work

There are several approaches for VAD and for SSL, using mostly audio cues, video cues or a combination of both (multimodal processing). In general, VAD and SSL are considered as two separate problems, and a brief revision of both problems is presented below.

Most of existing approaches for VAD rely on audio cues, either relying on characteristics of voice patterns in the frequency domain or pre-determined (or estimated) levels of background noise. Sohn et al. (SOHN et al., 1999) presented a statistical method using the complex Gaussian assumption in the frequency domain for both speech and noise, based on the likelihood ratio test (LRT). Additionally, they also proposed an effective hang-over scheme based on an HMM. Ramirez and collaborators (RAMIREZ et al., 2005) proposed its extension, by employing multiple observations to include temporal smoothing. A further improvement was presented in (RAMIREZ et al., 2007), by using contextual multiple hypothesis testing.

Other authors have used different statistical models to characterize speech periods. For instance, a Laplacian distribution was used to model speech in the DCT domain in (GAZOR; ZHANG, 2003), which has shown to be a better model than a Gaussian. More recently, Lee and Muhkerjee (LEE; MUHKERJEE, 2010b) proposed a statistical algorithm for VAD, aiming to detect higher level speech activities (e.g. sentences instead of syllables, words, phrases, etc.). Their approach uses two distinct features for VAD, energy and entropy in the DCT domain, which are modeled as chi-square and Gaussian distributions respectively.

It should be noticed that the approaches described in (SOHN et al., 1999; GAZOR; ZHANG, 2003) aim to detect very short-time silence periods suitable for tasks such as speech coding or automatic speech recognition, while our approach aims to detect higher level speech activities, as in (LEE; MUHKERJEE, 2010b; RAMIREZ et al., 2005, 2007).

Regarding SSL (in our case, the sound source is the active speaker), most approaches rely only on audio information. It is necessary to have more than one microphone, and to analyze the relationships among the signals captured by different microphones (BRANDSTEIN; WARD, 2001). For a two-microphone case, we can find the time difference of arrival (TDOA) using the generalized cross-correlation (GCC) method, which involves a frequency weighting function (BRANDSTEIN; WARD, 2001). One of the most popular frequency weighting for GCC is the phase transform (PHAT), which effectively whitens the microphone signals to equally emphasize all frequencies before computing the cross correlation. Even with noise and reverberation, the GCC-PHAT has been reported to work reasonably well in many practical situations (BRANDSTEIN; WARD, 2001).

For multiple microphone cases, we can find the source location by triangulation given a set of TDOA's from different microphones pairs (BRANDSTEIN; ADCOCK; SILVERMAN, 1997). The TDOA-based method becomes unreliable when the individual TDOA estimates are inaccurate to begin with. Unfortunately, this is often the case in typical acoustic environments. Alternatively we can use the *Steered Response Power* (SRP) method (DIBIASE, 2000), which can be considered as an extension of the GCC method to multiple microphones. The main idea of the SRP is to steer the microphone array to all possible candidate source locations to find one with the maximum power, typically using some frequency weighting (filtering in the time domain). In particular, the SRP method with the PHAT frequency weighting (SRP-PHAT) has been reported to be more robust with respect to acoustic corruptions, such as background noise and reverberation compared with the TDOA-based methods (BRANDSTEIN; WARD, 2001; DIBIASE, 2000; DO; SILVERMAN; YU, 2007).

With an array consisting M microphones, the SRP-PHAT of the sound source at a position $\mathbf{q}$ is (DIBIASE, 2000)

$$P(\mathbf{q}) = \sum_{m=1}^{M} \sum_{l=1}^{M} \int \frac{X_m(\omega) X_l^*(\omega)}{|X_m(\omega) X_l^*(\omega)|} e^{j\omega(\Delta_m^{\mathbf{q}} - \Delta_l^{\mathbf{q}})} d\omega, \tag{3.1}$$

where $X_m(\omega)$ is the Fourier Transform of the signal at the m^{th} microphone, $\Delta_m^{\mathbf{q}}$ is the time delay computed from position $\mathbf{q}$ to the m^{th} microphone.

Since the SRP-PHAT requires significant amount of computation, i.e., double summation and one integration for each candidate source, an alternative expression was proposed in (ZHANG; ZHANG; FLORENCIO, 2007):

$$P(\mathbf{q}) = \int \left| \sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{j\omega\Delta_m^{\mathbf{q}}} \right|^2 d\omega. \tag{3.2}$$

SSL can also be done using statistical modeling of multichannel audio signals using e.g., multivariate complex Gaussian (ZHANG; ZHANG; FLORENCIO, 2007), or Laplacian (LEE; KALKER; SCHAFER, 2008) distributions whose computational cost is typically much higher than the SRP-PHAT and thus not quite suitable for real-time implementations. Furthermore, it is important to note that the audio-based approaches described so far work on sound buffers independently, not exploring temporal coherence.

Multimodal approaches for SSL explore different sensors, mostly focused on audio-visual integration. Wang and Brandstein (WANG; CHU, 1997) proposed a face tracking algorithm based on both sound and visual cues. Initial talker locations are estimated acoustically from microphone array data, based on the TDOA estimation followed by a triangulation procedure. The final location is obtained based on video cues (acquired with a single camera), using mostly motion and edge information. Their work was further extended in (WANG; GRIEBEL; BRANDSTEIN, 2000) by adding head pose estimation using multiple cameras and multi-channel speech enhancement techniques.

In (VERMAAK et al., 2001) and (PEREZ; VERMAAK; BLAKE, 2004), a particle filtering framework was employed for data fusion, since it deals better with non-Gaussian distributions (opposed to Kalman filtering). Both approaches explore only a pair of microphones and a single monocular camera, and obtain the sound localization by measuring the TDOA between signals arriving at the two microphones. In (VERMAAK et al., 2001), an active contour tracking approach was used to explore visual data, using monochromatic information, while color and motion cues were used in (PEREZ; VERMAAK; BLAKE, 2004).

Gatica-Perez and collaborators (GATICA-PEREZ et al., 2007) presented a probabilistic approach to jointly track the location and speaking activity of multiple speakers in meeting room equipped with a circular microphone array (with 8 microphones) on a table and multiple cameras, that capture frontal and top views of the participants. They used the SRP-PHAT approach for SSL, and the visual observations were based on models of the shape and spatial structure of human heads. The fusion was performed with a Markov Chain Monte Carlo particle filter. Despite the good results presented in the paper, a point to be improved is the initialization of the targets. Also, the computational cost was not discussed, and more than one camera is required.

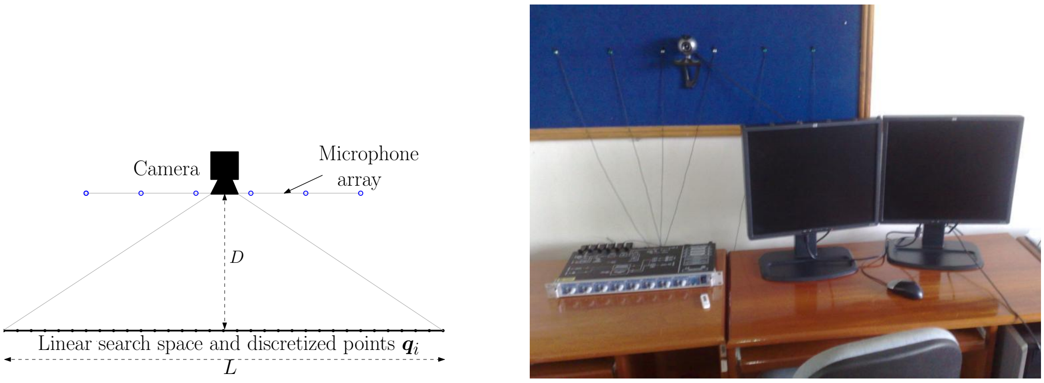
The group of Zhang (ZHANG et al., 2008) proposed a multimodal speaker identification approach that fuses audio and visual information at the feature level by using boosting to select features from a combined pool of both audio and visual features simultaneously, using a circular array with 6 microphones and a panoramic camera. The authors showed that their multimodal approach performed better than a unimodal sound source locator, but the motion cue may pose problems for stationary participants. Talantzis and collaborators (TALANTZIS; PNEVMATIKAKIS; CONSTANTINIDES, 2008) proposed an approach that estimates independently the position of the active speaker in cluttered and reverberant environments using audio and video information, and combined both outputs. Their method was tested with a large microphone array (80 microphones located in different locations inside the acoustic enclosure and organized in different topologies), and a set of five synchronized and calibrated cameras. One clear drawback of their approach is the hardware requirement (tens of microphones and multiple calibrated cameras).

Although VAD and SSL have been treated as independent problems in the literature, this chapter presents a new approach that explores spatio-temporal characteristics that arise in the SSL problem to solve the VAD problem. To further improve SSL results, we have also included visual information captured by a single camera. The proposed approach is described in the next section.

3.3 Our Approach

The proposed approach explores the expected spatio-temporal consistency of audio signals through two competing HMMs (one for silence and the other for speech) to distinguish silence from speech. When voice activity is detected, visual information is included in the third HMM to provide a robust estimate of the active speaker. To correlate audio and video signals, we assume a linear microphone array and a video camera placed right at the center of the array, arranged so that the camera plane is vertical and parallel to the array. The users are expected to be at an approximately constant distance D from the array, so that the search space required for the SRP-PHAT is linear, parallel to the array, and its length L is given by the Field of View (FOV) of the camera given D. A schematic representation of the required setup is provided in Figure 3.1(a), and our prototype room based on such setup is given in Figure 3.1(b).

The first step of our algorithm consists of initially discretizing the search space into N equally spaced positions $\mathbf{q}_i$ and applying Equation (3.2) to compute the SRP-PHAT values at such points, as illustrated in Figure 3.1(a). Since the distance between adjacent


Camera
Microphone array
D
Linear search space and discretized points $\boldsymbol{q}_i$
L

(a)

(b)

Figure 3.1: (a) Schematic representation of the proposed approach. (b) Our prototype system.

points is given by $L/(N-1)$, it is clear that parameter N relates to the precision of SSL: larger values for N lead to search points closer to each other providing higher spatial resolution at the cost of higher computational cost. Based on these N points, a 2D feature vector is computed (described in Section 3.3.1), and HMMs that describe both speech and silence situations are proposed.

One important thing to note here is that the spatial resolution is bounded by the choice of the sampling frequency. For example, at 44,100 Hz sampling frequency with the sound propagation speed of 343 m/s, one sample delay corresponds to $343/44100 = 0.78\,\text{cm}$ propagation distance. For a 6 element microphone array with the microphone spacing of 8 cm, we can achieve the minimum FOV of $\sin^{-1}\left(\frac{.78}{8\cdot(6-1)}\right) = 1.12°$ that gives $\tan(1.12°) = 1.96\,\text{cm}$ lateral distance (precision) for sources at 1 m away from the microphones. Therefore, N needs to be carefully chosen not to incur higher computational cost without gaining higher spatial resolution depending upon specific applications.

3.3.1 The Proposed HMMs

A HMM can be used to model dynamic systems that may change their states in time. A HMM with discrete observables is characterized by $\lambda = (A, B, \pi)$, where $A = [a_{ij}]$ for $1 \leq i, j \leq N$ is the transition matrix that contains the probabilities of state changes, $B = [b_i(\mathbf{O})]$ for $1 \leq i \leq N$ describes the observation probability for each state, and $\pi = [\pi_i]$ for $1 \leq i \leq N$ contains the initial probabilities of each state. Clearly, the choice of the parameters is crucial to characterize a given HMM.

In this work, we explore specific characteristics that are expected to arise in silence and speech situations to devise two HMMs that describe each situation. In the test stage, the HMM that best describes the captured observations is used to detect silence or speech periods. Furthermore, the speech HMM embeds spatio-temporal coherence, increasing the robustness of the SSL (when speech is detected).

The proposed HMMs consist of N hidden states, and each state $S(t)$ at time t relates to a discretized spatial position used to compute the SRP-PHAT. Another key issue is the definition of a set of observables that are adequate for silence/speech discrimination and SSL. In this work, the observables are two-dimensional vectors $\mathbf{O} = (O_1, O_2)$ that are computed based on the SRP-PHAT responses in all discretized positions, given by

$$O_1 = \underset{i}{\text{argmax}}\, P(\mathbf{q}_i), \tag{3.3}$$

$$O_2 = \frac{\max P(\mathbf{q}_i)}{\sum_{i=1}^{N} P(\mathbf{q}_i)}. \tag{3.4}$$

It is important to note that O_1 is exactly the discretized position that produces the maximal SRP-PHAT response. The second observable O_2 can be viewed as a confidence measure when estimating O_1. In speech situations, O_1 should provide the correct location of the speaker, and O_2 tends to be a large value (since the global maximum is expected to be considerably larger than the mean). On the other hand, in silence situations, all values of $P(\mathbf{q}_i)$ tend to be similar, and the largest one is selected as the location of the (inexistent) sound source. In this case, however, O_2 will be smaller (close to the lower bound), and this information is crucial when building the competing HMMs.

In theory, the lower bound for O_2 is $1/N$, and the upper bound is 1. We have observed in different experiments (with different speakers and varying background noise) that O_2 gets really close to the minimum theoretical value $1/N$ when no active speaker is present, and stays very far from the upper bound 1. In fact, when N increases so does the denominator in Equation (3.4), so that O_2 tends to be smaller. For a given value of N, an experimentally set maximum bound O_2^{max} is defined, and the values of O_2 are quantized into Q_2 possible values within the range $O_2^{\text{min}} = 1/N$ and O_2^{max} to obtain an HMM with discrete range observables.[1]

A widely used approach to select the parameters for a given HMM is the Baum-Welch algorithm (RABINER, 1989), which iteratively estimates the parameters $\lambda = (A, B, \pi)$ based on a set of training data. Our HMM, however, presents a relatively high number of states (N) and observables ($N \times Q_2$), which would require a large amount of training samples (comprising several situations such as speakers in different positions, alternation of speech and silence, presence/absence of background noise, etc.). Instead, we propose a parametric model for the HMM parameters based on the expected behavior of the user in audiovisual HCIs, as described next.

3.3.1.1 The Speech HMM

The joint probability distribution[2] $p_i^{\text{sp}}(O_1, O_2)$ for a given state $\mathbf{q}_i$ can be written as

$$p_i^{\text{sp}}(O_1, O_2) = p_i^{\text{sp}}(O_1|O_2)p^{\text{sp}}(O_2), \tag{3.5}$$

where $p^{\text{sp}}(O_2)$ is the distribution of O_2 during speech situations (which does not depend on the state $\mathbf{q}_i$), and $p_i^{\text{sp}}(O_1|O_2)$ is the conditional probability of O_1 given O_2, which is strongly affected by i. During speech situations, larger values for O_2 are expected to arise, since sharper peaks tend to occur in the SRP-PHAT. The proposed model is a discrete Gaussian distribution centered at O_2^{max}:

$$p^{\text{sp}}(O_2) = \exp\left\{-\frac{(O_2 - O_2^{\text{max}})^2}{2\sigma_{sp}^2}\right\}, \tag{3.6}$$

[1] Values of O_2 larger than O_2^{max} were quantized to the largest possible value. In all experiments, we quantized the values O_2 into $Q_2 = 7$.

[2] Since our observables are discrete, all PDFs should add up to unity. For the sake of simplicity, the normalization factor of each discrete PDF will be omitted.

where σ_{sp} relates to the variation of O_2 in speech situations.

In order to find an adequate model for $p_i^{\text{sp}}(O_1|O_2)$, it is first important to note the following. If a given state $\mathbf{q}_i$ is in fact the actual position of the active speaker (in speech situation), the observable with the highest probability of occurring is $O_1 = i$, so that $p_i^{\text{sp}}(O_1|O_2)$ is expected to present a peak at $O_1 = i$. If the confidence O_2 is large, the probability should decay rapidly as O_1 gets far from i. If O_2 is smaller, the decay should be slower, since the confidence is smaller and other values of O_1 are also expected to be encountered with higher probability. There are several functions that satisfy those criteria, and in this work an exponential function was chosen to model the decay from the peak:

$$p_i^{\text{sp}}(O_1|O_2) = \exp\left\{-\frac{|O_1 - i|}{f(O_2)}\right\}, \tag{3.7}$$

where $f(O_2)$ is a monotonically decreasing function that controls the probability decay for different values of O_2. Our choice for f based on empirical evaluations is another exponential function:

$$f(O_2) = 2N\exp\{-2N(O_2 - O_2^{\text{min}})\}, \tag{3.8}$$

so that a very sharp peak of $p_i^{\text{sp}}(O_1|O_2)$ appears when $O_2 \approx O_2^{\text{max}}$, and the decay is very slow when $O_2 \approx O_2^{\text{min}}$.

In order to define the state transition matrix, we first observe that the user typically does not present abrupt head movements in time in most HCIs for multimedia applications. Hence, during speech periods, the state transition probability $a_{ij}^{\text{sp}} \triangleq p^{\text{sp}}(S(t+1) = \mathbf{q}_j|S(t) = \mathbf{q}_i)$ should prioritize the maintenance of the current state or changes to neighboring states, and penalize changes to states far away from each other. In this work, an exponential function was used to build the transition matrix A_{sp}:

$$a_{ij}^{\text{sp}} = \nu_i \exp\left\{-\frac{|i-j|}{\beta_{\text{sp}}}\right\}, \tag{3.9}$$

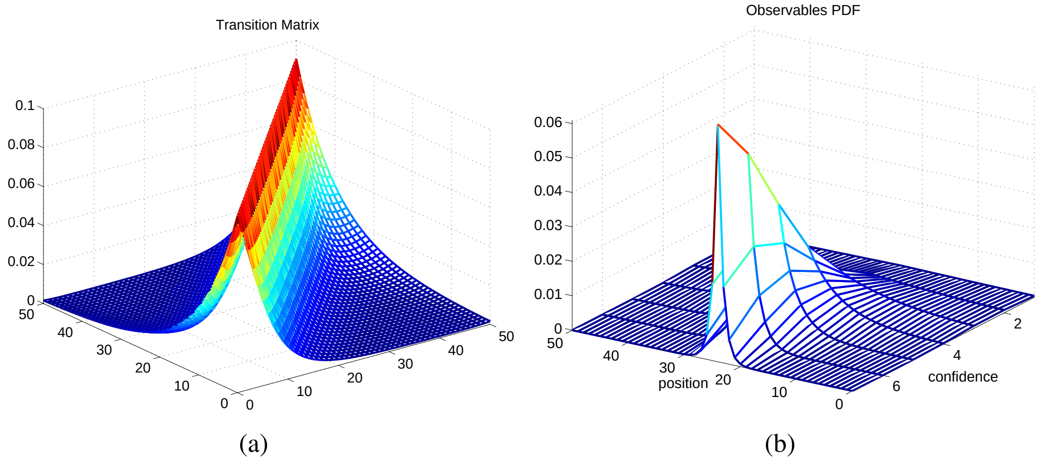
where β_{sp} controls the decay of the exponential and ν_i is a normalization factor to guarantee that each line of A_{sp} adds up to unity. Let us recall that the spacing between adjacent positions is $L/(N-1)$, and let v_m be the maximum lateral speed of a participant (in m/s), and F_s be the sampling frequency (in samples per second). Then, we select $\beta_{\text{sp}} = \frac{v_m(N-1)}{F_s L \ln 2}$, so that $a_{ij} = 0.5a_{ii}$ when $|i-j|L/(N-1) = v_m/F_s$, i.e., the probability of moving at the maximum speed within adjacent frames is half of the probability of staying at the same position. If v_m increases, so does β_{sp}, meaning that wider transitions should be allowed during speech situations.

For the sake of illustration, the transition matrix A_{sp} and the probability density function $p_{25}^{\text{sp}}(O_1, O_2)$ related to state $\mathbf{q}_{25}$ ($N = 51$) are shown graphically in Figure 3.2 (the parameters used to generate those plots are stated in Section 3.4).

3.3.1.2 The Silence HMM

The silence-related HMM can be viewed as the “dual” of the speech-related HMM. As it was already pointed out, during silence periods the response of the SRP-PHAT at each position (state) should be similar, so that observable O_2 is expected to be close to the smallest possible value, which is O_2^{min}. As in the speech situation, the joint probability function can be written as

$$p_i^{\text{si}}(O_1, O_2) = p_i^{\text{si}}(O_1|O_2)p^{\text{si}}(O_2). \tag{3.10}$$



(a) (b)

Figure 3.2: Illustration of (a) the transition matrix and (b) probability distribution function of observables ($i = 25$), for the speech-related HMM.

Function $p^{\text{si}}(O_2)$ was obtained similarly to its counterpart in speech situations:

$$p^{\text{si}}(O_2) = \exp\left\{-\frac{(O_2 - O_2^{\min})^2}{2\sigma_{si}^2}\right\}, \tag{3.11}$$

where σ_{si} relates to the variation of O_2 in silence situations.

For the conditional probability $p_i^{\text{si}}(O_1|O_2)$, there are two important things to be noted. First, such distribution should not depend on the state $\mathbf{q}_i$, since the position of the peak is related to noise, and not to an actual sound source at position $\mathbf{q}_i$. Secondly, all observables O_1 should be equally probable, for the same reason. Hence, the same uniform conditional probability function is chosen for all states:

$$p_i^{\text{si}}(O_1|O_2) = p^{\text{si}}(O_1|O_2) = \frac{1}{N}. \tag{3.12}$$

If in speech situations the peak of the SRP-PHAT is expected to be close in temporally adjacent observations, the same is not true for silence periods. Since all responses are usually similar, background noise may play a decisive role when retrieving the highest peak, which may be far from the one detected in the previous observation. In fact, the proposed state transition matrix for the silence-related HMM considers all transitions equally probable:

$$a_{ij}^{\text{si}} = \frac{1}{N}. \tag{3.13}$$

As for the initial distribution π for both speech and silence HMMs, we assumed that all states (i.e., positions) are initially equally probable.

3.3.1.3 Silence Detection and Location of the Active Speaker

Given the speech HMM λ_{sp}, the silence HMM λ_{si}, and a sequence of observables $\mathcal{O}_t = \left\{\mathbf{O}(t-T), \mathbf{O}(t-T+1), ..., \mathbf{O}(t)\right\}$ within a time window with size T, we can compute how well each HMM describes $\mathcal{O}_t$. More precisely, this can be done by computing $P(\mathcal{O}_t; \lambda_{\text{sp}})$ and $P(\mathcal{O}_t; \lambda_{\text{si}})$ using the forward-backward procedure (RABINER, 1989). Hence, based on a temporal window of size T, a given t is classified as silence if $P(\mathcal{O}_t; \lambda_{\text{sp}}) < P(\mathcal{O}_t; \lambda_{\text{si}})$, and classified as speech otherwise.

When the speech HMM wins, it is possible to apply the Viterbi algorithm (RABINER, 1989) to compute the most probable sequence of states $\{S(t-T+1), S(t-T+2), ..., S(t)\}$ based on the sequence of observations $\{O(t-T+1), O(t-T+2), ..., O(t)\}$. Since each state is related to a discretized position used in the SRP-PHAT algorithm, the sequence of states relate to the most probable trajectory of the active speaker considering the last T samples (and the last estimated state $S(t)$ is retrieved as the current position of the active speaker).

3.3.2 Inclusion of Video Information

Despite the improvement of the temporal coherence through the use of the HMM, noisy environments and severe reverberation may produce erroneous peaks in the SRP-PHAT computation, even when a participant is effectively speaking. Assuming that the participants are the possible sound sources, and they are usually facing the camera (which is a plausible hypothesis in HCI applications), the use of a face detection or tracking algorithm could guide the selection of the largest power location in the SRP-PHAT algorithm.

In this work, we used the face tracker algorithm described in (BINS et al., 2009), that is robust with respect to significant head tilts and turns and also to partial occlusions. In a nutshell, the approach in (BINS et al., 2009) is based on the individual tracking of KLT (Kanade-Lucas-Tomasi) features, which are combined in a robust manner using Weighted Vector Median Filters.

The face tracker provides, at each frame, the number k of identified faces, as well as the face centers $\mathbf{x}_1$, $\mathbf{x}_2$,... $\mathbf{x}_k$ in image coordinates. Since the camera is fixed and the depth of detected faces is assumed to be around 1 m, a simple projective mapping can be used to relate each face center in image coordinates (pixels) to its corresponding position in world coordinates (cm). Finally, this position in world coordinates can be easily mapped to the discretized positions $\mathbf{q}_i$ used in the computation of the SRP-PHAT, which are equivalent to the states used in the proposed HMMs. Let $f_j = h(\mathbf{x}_j)$ denote the mapping from the image coordinates of the j^{th} detected face to its correspondent state.

Although there may be several ways to combine audiovisual cues, in this work we employed a simple (but effective) technique to weight the output produced by the SRP-PHAT based on the detected faces, so that spatial locations around the detected faces are prioritized. More precisely, given the SRP-PHAT responses $P(\mathbf{q}_i)$, the weighted responses $P'(\mathbf{q}_i)$ are given by

$$P'(\mathbf{q}_i) = \sum_{j=1}^{k} P(\mathbf{q}_i) \left(1 + \gamma \exp\{-(i - f_j)^2/2\sigma^2\}\right), \qquad (3.14)$$

where $\gamma > 0$ controls the amplitude of the exponential, and σ the standard deviation. Experimentally, we set $\gamma = 0.25$ and $\sigma = N/30$.

One approach to improve the unimodal SRP-PHAT algorithm is to find the maximum of the weighted SRP-PHAT responses provided in Equation (3.14), which embed video information. To further include spatio-temporal coherence in the model, the HMM described in Section 3.3.1 is also included, using $P'(\mathbf{q}_i)$ instead of $P(\mathbf{q}_i)$ in Equations (3.3) and (3.4).

It is important to note the face tracker artificially increases the SRP-PHAT value at the face position, which could bias the silence detector. In this regard, for silence detection, described in the previous subsection, we use only audio information to compute $P(\mathcal{O}_t; \lambda_{\text{sp}})$ and $P(\mathcal{O}_t; \lambda_{\text{si}})$. When speech is detected, the audiovisual HMM (using the video-based weighted SRP-PHAT responses) is then used to improve the estimated location of the active speaker. In the next section, we present some results of the silence/speech discrimination, as well as comparisons of speaker localization using the unimodal SRP-PHAT algorithm and the proposed improvements presented in this chapter.

3.4 Experimental Results

All our experiments were conducted in our prototype room, which is equipped with a uniform array of six DPA 4060 omnidirectional microphones, placed 8 cm apart from each other, and a Logitech Quickam Pro 5000 webcam as depicted in Fig. 3.1(b). The participants are expected to be found approximately 1 meter away from the camera and the array (and approximately 50 cm from the monitor), and the field of view of our webcam considering such distance corresponds to a region approximately 94 cm wide. This linear search space was discretized into $N = 51$ points, yielding a distance of approximately 1.88 cm between neighboring points in the discretized search space based on our discussion on the spatial resolution in Section 3.3.

The audio signals were captured at 44,100 Hz, and 4096 samples were used to compute the SRP-PHAT (so that we update the SRP-PHAT localization approximately every 0.093 s). Video capture was synchronized with audio, so that approximately $F_s = 10$ audiovisual samples are captured per second. The size T of the temporal window described in Section 3.3.1.3 is related to temporal coherence. If a small value is chosen for T, speech hiatus between consecutive words may be detected as silence, which is usually not desirable for speech recognition. On the other hand, larger values for T provide better temporal consistency, but also lead to delays when detecting speech-silence or silence-speech changes. In this work we used $T = F_s = 10$, which corresponds to a window approximately 1 second long, and showed to be efficient to deal with speech hiatus and not present a long delay when the location of the speaker changes.

For our experimental setup, the selection of the required parameters is given as follows. Given a training sequence containing both speech and silence periods in approximately equal proportions (we used video sequence 3, described later in this work), we compute O_2 for all frames, and remove the largest 2% values (to eliminate possible outliers). Then, define $O_2^{\max}$ as the maximum of the remaining values (for our setup, we obtained $O_2^{\max} = 0.0603$). To avoid the need of having ground truthed data for training, we defined $\sigma_{si} = \sigma_{sp}$ required in Equations (3.6) and (3.11). They are defined as the standard deviation of O_2 in the training sequence (that contains both speech and silence periods), obtaining the value of 0.0133 for our experimental setup. Also, assuming that the maximum lateral speed of the participants is at most $v_m = 1.31$ m/s (that is the mean speed of pedestrians with unconstrained flow as reported in (ROBIN et al., 2009)), and the given values for L and N, we get $\beta_{sp} = 0.2011(N - 1)$ in Equation (3.9).

A total of seven ground-truthed video sequences, 1 minute long each, were generated to evaluate the accuracy of the proposed approach. The first three sequences contain a single speaker, that alternates speech with silence, and the other four sequences contain two participants, that alternate speech (and shorter periods of silence). Table 3.2 provides a very brief description of each video sequence used in this Section (number of participants, presence/type of background noise, number of speech-related frames), and the processed video sequences can be accessed at `http://www.inf.ufrgs.br/~crjung/multimodal_localization/`. For all sequences, the nine initial frames were discarded in the analysis, since the proposed HMM requires a time span of ten frames.

3.4.1 Voice Activity Detection

For VAD, only the first three sequences were used, since the other videos contain smaller periods of silence (and much more alternation between speakers and silence periods, which makes the manual markings for VAD/silence inaccurate). In total, 1914 frames were analyzed, and VAD results are summarized in Table 3.1, that shows the confusion matrix for both video sequences. Considering the results of all three sequences, the true positive rate was around 88.93%, and the true negative rate around 85.73%, yielding a total accuracy of 87.62%. It is also important to note that the temporal coherence imposed by the HMMs causes small delays in the silence/speech detection, which impacts the quantitative analysis. More precisely, from the 237 samples incorrectly classified, 35.87% occured at most five frames from a silence/speech or speech/silence transition. If we consider this 5-frames tolerance, the overall accuracy increases to 92.06%.

		Actual					
		Video 1		Video 2		Video 3	
		Speech	Silence	Speech	Silence	Speech	Silence
Detected	Speech	395	44	306	25	308	25
	Silence	28	171	74	218	23	284

Table 3.1: VAD results.

3.4.2 Active Speaker Location

To evaluate the accuracy of the proposed SSL algorithm, we compared the actual position of the active speaker with the SRP-PHAT approach, the combination of the SRP-PHAT with the speech HMM (called SRP-PHAT+HMM), and the complete approach combining the video-based weighting function, SRP-PHAT and HMM (called SRP-PHAT+video+HMM). The location of the active speaker was discretized into positions $\mathbf{q}_i$, each of which is considered as one of the hidden states of the HMM. Ground-truth data were manually generated for the video sequences, and the error E (absolute difference between the actual and the detected position) was evaluated for the three analyzed approaches, considering only speech-related frames from the seven video sequences.

Table 3.2 summarizes the quantitative analysis, providing the mean errors for each approach and video sequence. This table also presents the percentage of correct results (hits) achieved by each approach. A certain detection was considered a hit when $E \leq \tau$, where $\tau \geq 0$ is a tolerance that allows detected positions close to the ground truth to be considered correct (in Table 3.2 we used $\tau = 3$, which corresponds to a tolerance of approximately 5.64 cm). In most video sequences, the percentage of hits was improved when the HMM was employed, and in all sequences it was further improved when video information was included. The exception was video 2, that contains a person moving relatively fast while speaking. Since the HMM tends to preserve spatio-temporal coherence, it may also present small delays when the position changes rapidly, which explains the apparent loss of performance in this sequence. As shown in Figure 3.3, the introduction of the HMM indeed produces a smoother localization of the active speaker when compared to the direct use of SRP-PHAT, but introducing some delay when more abrupt motion happens.

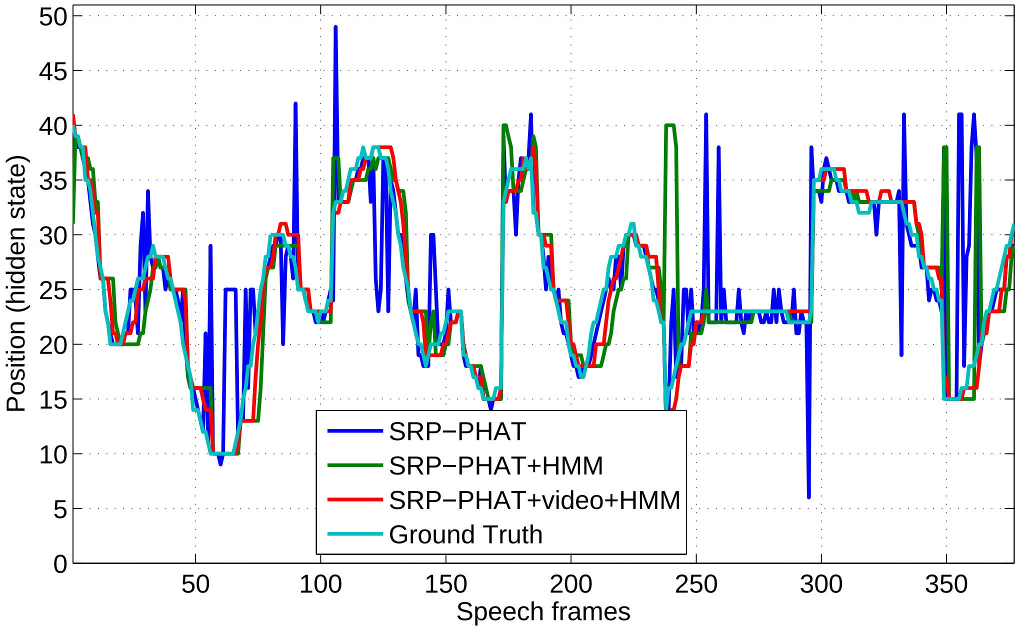


Figure 3.3: Results of active speaker localization for video 2.

A plot of the percentage of hits as a function of τ considering all video sequences is presented in Figure 3.4. As it can be observed, the accuracy of the multimodal approach presents the best results for all tolerance values τ.

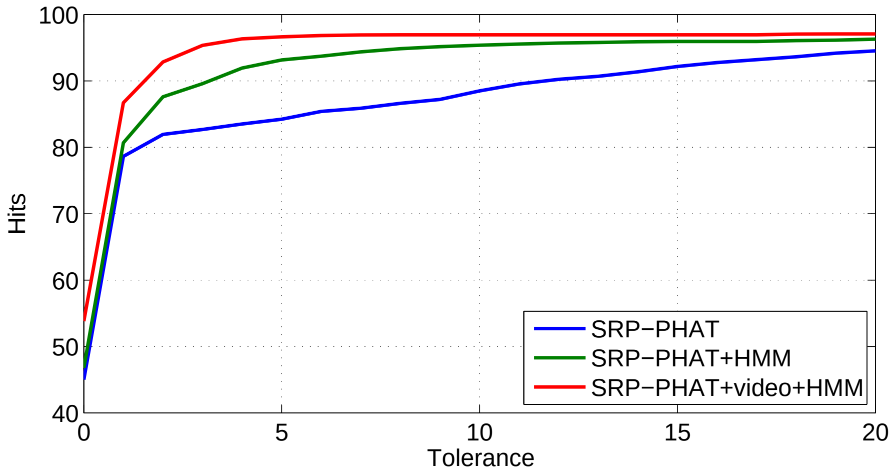


Figure 3.4: Percentage of correct localization results (hits) for different tolerance values.

As the number of microphones increases, the SRP-PHAT algorithm tends to further amplify the signal at the actual position of the sound source, so that better results for SSL are expected. An extra gain is also expected using the proposed approach, since increasing the number of microphones also tends to produce larger values for O_2 during speech situations, as shown in Figure 3.5. Table 3.3 shows the overall SSL accuracy considering all seven video sequences using SRP-PHAT, SRP-PHAT+HMM, and SRP-PHAT+video+HMM, using a different number of microphones. As expected, results produced by SRP-PHAT alone are considerably degraded when fewer microphones are used, but the proposed approach keeps the overall accuracy around 90% even when

Approach	SRP-PHAT	SRP-PHAT +HMM	SRP-PHAT +video+HMM
Video 1: One person, low background noise, 423 speech frames			
Mean error	1.53	0.98	0.57
Hits (%)	90.31	94.09	99.53
Video 2: One person, AC turned on in the middle, 377 speech frames			
Mean error	2.16	2.24	1.16
Hits (%)	86.74	81.43	94.96
Video 3: One person, music in background, 329 speech frames			
Mean error	1.52	1.93	0.24
Hits (%)	93.62	94.83	100.00
Video 4: two persons, low background noise, 500 speech frames			
Mean error	4.35	2.98	2.51
Hits (%)	76.20	90.20	92.20
Video 5: two persons, music in background, 452 speech frames			
Mean error	4.65	2.39	2.37
Hits (%)	73.45	88.05	93.36
Video 6: two persons, switching positions, 432 speech frames			
Mean error	4.06	2.15	1.69
Hits (%)	78.70	87.73	93.75
Video 7: two persons, 483 speech frames			
Mean error	4.23	1.12	1.06
Hits (%)	80.12	93.17	95.24

Table 3.2: Performance of the analyzed algorithms for locating the active speaker. Error relates to the absolute difference between the actual hidden state of the HMM and the detected state (1 unit $\approx$ 1.88 cm).

only two microphones are used. In fact, the gain of introducing the HMM to SRP-PHAT when using two microphones is around 12%, whereas the gain of further including video information rises to almost 50%.

	2 mics	4 mics	6 mics
SRP-PHAT	60.31%	74.67%	82.01%
SRP-PHAT+HMM	70.76%	86.85%	89.95%
SRP-PHAT+HMM+video	89.52%	94.46%	95.33%

Table 3.3: Accuracy of different SSL algorithms varying the number of microphones.

It is also important to notice that the distance D from the users to the camera should also impact the results, since it changes the relation between distances in image coordinates (pixels) and world coordinates (meters). However, as D usually does not vary significantly (particularly in HCIs that also involve keyboard typing), so we did not evaluate the effect of varying D in our experiments.

3.5 Conclusions

This chapter presented a new approach to simultaneously distinguish silence from speech, and to locate the active speaker in the latter case. Audio information is captured

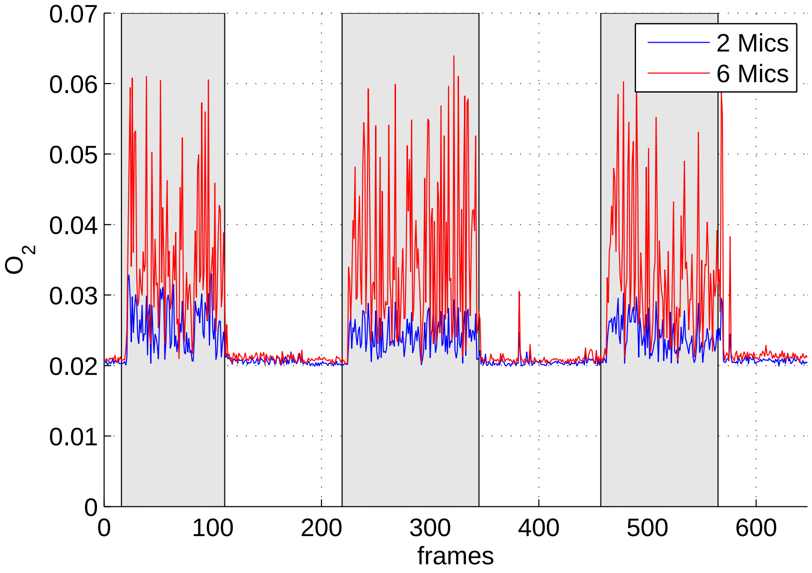


Figure 3.5: Plot of observable O_2 for several frames of video sequence 3. Shaded regions relate to speech, and the others to silence.

through an array of microphones, and the SRP-PHAT algorithm is used to provide an estimate location based solely on audio cues. A face tracking algorithm is then used to detect the participants across time, and the tracked faces are used to improve the results provided by the SRP-PHAT. Finally, two HMMs are built to model typical silence and speech situations, respectively. The HMMs are used to detect silence or speech situations, and in that case, the corresponding HMM embeds spatio-temporal coherence.

The experimental results showed that the proposed HMM increases the accuracy of the SSL when compared to the SRP-PHAT, and the combination with the face tracking algorithm presents even better results. The gain with respect to SRP-PHAT alone is even more considerable when few microphones are used, as shown in Table 3.3. A possible future research direction would be the formulation of a different HMM to allow the localization of more than one participant speaking simultaneously, and the use of programmable graphic processing units (GPUs), to reduce execution time. A more comprehensive study on other functions to model $p_i^{\mathrm{sp}}(O_1, O_2)$ can also be performed.

References

See the unified bibliography of the dissertation.

4 AUDIOVISUAL VOICE ACTIVITY DETECTION BASED ON MICROPHONE ARRAYS AND COLOR INFORMATION

Minotto, V.P.; Lopes, C.B.O.; Scharcanski, J.; Jung, C.R.; Bowon Lee, "Audiovisual Voice Activity Detection Based on Microphone Arrays and Color Information," Selected Topics in Signal Processing, IEEE Journal of , vol.7, no.1, pp.147,156, Feb. 2013 doi: 10.1109/JSTSP.2012.2237379
Keywords: Voice Activity Detection, Microphone Array, SRP-PHAT, Lips Detection, Hidden Markov Model

Abstract

Audiovisual voice activity detection is a necessary stage in several problems, such as advanced teleconferencing, speech recognition, and human-computer interaction. Lip motion and audio analysis provide a large amount of information that can be integrated to produce more robust audiovisual voice activity detection (VAD) schemes, as we discuss in this chapter. Lip motion is very useful for detecting the active speaker, and in this chapter we introduce a new approach for lips and visual VAD. First, the algorithm performs skin segmentation to reduce the search area for lip extraction, and the most likely lip and non-lip regions are detected using a Bayesian approach within the delimited area. Lip motion is then detected using Hidden Markov Models (HMMs) that estimate the likely occurrence of active speech within a temporal window. Audio information is captured by an array of microphones, and the sound-based VAD is related to finding spatio-temporally coherent sound sources through another set of HMMs. To increase the robustness of the proposed system, a late fusion approach is employed to combine the result of each modality (audio and video). Our experimental results indicate that the proposed audiovisual approach presents better results when compared to existing VAD algorithms.

4.1 Introduction

Voice activity detection is an important problem for various applications, such as video-conferencing (to identify silence periods and improve sound quality), speech recognition systems (VAD is crucial to determine which audio frames to process), and human-computer interaction systems for identifying human activities involving speech.

Speech is a bimodal signal, involving acoustic and visual information (SODOYER et al., 2009). Nevertheless, automatic speech recognition (ASR) systems often focus on acoustic information only. Typical audio-based VAD (AVAD) algorithms rely on some kind of estimate for background noise, and speech is detected when the signal intensity is higher than the noise level (TANYER; OZER, 2000). This focus on audio information makes these systems very sensitive to environmental and channel issues (e.g. noise), which has motivated research in audio preprocessing techniques and noise adaptation algorithms (SOHN; SUNG, 1998; SOON; KOH; YEO, 1999). Also, there are several challenging issues related to these approaches, such as non-stationary audio noise, reverberations, speech with low voice intensity etc.

Visual information like body and facial expressions, or lips and tongue movements, can complement to audio information and help in the voice activity recognition task. Actually, lip motion information is one of the best visual clues for recognizing when a person is speaking or is silent, since the lips move more than 80% of the time in human speech (WANG; WANG; XU, 2010). In fact, studies with controlled video information (SODOYER et al., 2009) demonstrate that lip movement is highly correlated to speech, but the extraction of visual cues using computer vision algorithms is still a challenging task.

Another class of approaches explores both audio and video cues for VAD, aiming to improve the robustness of individual modalities. Such techniques must deal with large amounts of data, particularly when multiple cameras and/or microphones are used. This work proposes a multimodal VAD (MVAD) approach based on a single monocular camera and an array of microphones, focusing on a videoconferencing or human-computer interaction setup where the participants are facing the camera at a roughly known distance. Video information is explored by temporally analyzing the region around the lips, and audio information is used to detect spatio-temporally coherent sound sources in a search space. Each modality is analyzed separately by individual classifiers that are both based on HMMs competition schemes. Using the output probability of each individual approach, a new classifier is built for MVAD. This way, our main contributions are the presented color-based visual voice activity detection (VVAD) algorithm and the late fusion scheme to the AVAD algorithm in (BLAUTH et al., 2012). We show that having good unimodal VAD algorithms, many possible supervised classifiers may be used to form a robust MVAD approach.

The remainder of this chapter is organized as follows. Section 4.2 revises some VAD approaches, and the proposed MVAD algorithm is described in Section 4.3, along with the color-based VVAD technique. Experimental results are presented in Section 4.4, and the conclusions drawn in the final section.

4.2 Related Work

Lip segmentation is an active research area, and several methods have been proposed. For example, Wang et al. (WANG; WANG; XU, 2010) proposed to extract Haar-Like features, local variances and train an SVM classifier for lip detection; then, a Kalman filter estimates the mouth center in the next video frame to track the lips in real time. However, if the Kalman filter fails to predict the next mouth and lips locations, and lip tracking is interrupted. Rohani et al. (ROHANI et al., 2008) used a fuzzy clustering approach for lip contour extraction. They preprocess the face images using local Successive Mean Quantization Transform (SMQT) features and a split up Snow classifier, and lips are estimated to be in the lower part of the frame partition. An RGB to pseudo hue transformation is performed in this smaller frame portion, and Fuzzy c-means (FCM) is used for clustering lip and non-lip pixels. Afterwards, they filter morphologically the detected blobs, model lips as ellipses to select and enhance the detected lip pixels. The method of Rohani et al. obtains good results, however it is dependent on the face detection quality for locating the lips correctly, i.e. the estimate of the face portion where lips are located before the lip are segmented, which is not detailed in (ROHANI et al., 2008). Skin information also has been proposed to delimit the lip search area. Yao and Gao (YAO; GAO, 2001) proposed to detect lips based on skin and lip chrominance transformations. We propose a Bayesian approach to model skin tones using Gaussian mixtures (YANG; AHUJA, 1999; FIGUEIREDO; JAIN, 2002; WEBB, 2002), and delimit a prior lip search area. Then, the chrominance information available on the CIELAB and on the HSV (Hue, Saturation, and Value) color spaces are used to discriminate lips and non-lips pixels within the skin delimited area. Finally, the lips are segmented by maximum-likelihood with the Otsu method (OTSU, 1979).

Once the lips are extracted, the problem of VVAD can be approached. Sodoyer et al. (SODOYER et al., 2006) proposed to exploit in controlled situations the smoothed temporal differences of inter-lips width and height. Despite the good results achieved by this approach, it is hard to assess the influence of errors in inter-lips width and height estimates. Aubrey and colleagues (AUBREY et al., 2007) proposed two algorithms for VVAD, one based on active appearance models (AAMs) to obtain the lips and a Hidden Markov Model (HMM) for VAD, and the other employing a retinal filter in a region around the lips, and the temporal difference and a metric for VAD. They concluded that the AAM-based method was suitable for the detection of the non-speech sections containing complex lip movements, and the retinal filter based method was better on the detection of non-speech where the lips move less.

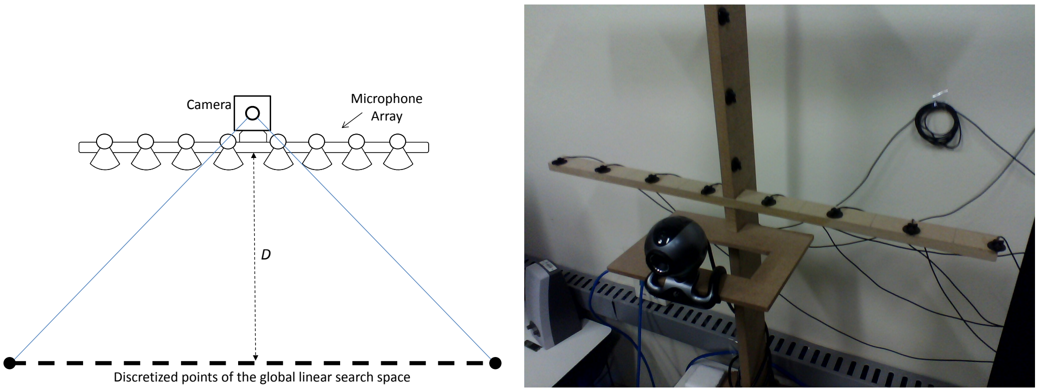
Aoki et al. (M. AOKI K. MASUDA; ARIKI, 2007) proposed an approach for VAD based on lip shape tracking using Elastic Bunch Graph Matching (EBGM), combined with audio cues. The visual analysis in this work employs the Gabor wavelet to extract feature points, and EBGM to match a generic face graph with the detected features. The lip aspect ratio (height over width) is used to measure mouth openness, and the aspect ratios temporal differences are thresholded to produce VAD in videos (which is combined with audio cues). However, the temporal differences can be noisy affecting the VAD results, and the authors used an infrared camera to better handle lighting changes. Sodoyer et al. (SODOYER et al., 2009) studied the existing relationship between lip movements and VAD, and concluded that the lips shapes can be similar during both voice activity or silence, and that dynamical parameters can provide enough separability if an adequate temporal window is used in the analysis.

For audio-based VAD, many different approaches exist. Sohn et al. (SOHN et al., 1999) proposed an AVAD method that uses frequency bands of the speech signal as input features for the likelihood ratio test, which is then followed by an HMM hang-over scheme to impose temporal coherence to the detector. Lee and Muhkerjee (LEE; MUHKERJEE, 2010a) proposed a statistical detector suitable for activities that contain high levels of speech. They model the entropy and the energy of the audio signals, in the decorrelated domain, as Gaussian and chi-square distributions, respectively. For classification they use the joint likelihood ratio estimate of the proposed features, for a given time interval, followed by an HMM-based smoothing technique. Gazor and Zhang (GAZOR; ZHANG, 2003) modeled clean and noisy speech, also in the decorrelated domain, as Laplacian and Gaussian distributions, also using HMM for preventing false negatives for weak speech. Despite performing well, as reported by the authors, statistical VAD algorithms, such as these just mentioned, require accurate estimation of each frame's signal-to-noise ratio (SNR). Although noise at each frame can be adaptively estimated (SOHN et al., 1999), signals that contain high levels of noise, and/or in which the noise is non-stationary, are very challenging for AVAD algorithms (LEE; HASEGAWA-JOHNSON, 2009).

In the work of Blauth and colleagues (BLAUTH et al., 2012), AVAD is treated as a spatio-temporal coherence problem through the use of a microphone array. HMMs are used as a tool to evaluate the behavior of the Steered Response Power using the Phase Transform (SRP-PHAT) method across time. If the SRP-PHAT behaves as the HMMs predict it should do in speech situations, speech is considered active, and inactive otherwise. This method does not require close speech capture and is robust against noise and reverberation. This owes to the SRP-PHAT being a robust sound source localization (SSL) method (DIBIASE, 2000), and to the HMMs' temporal analysis of the SRP-PHAT's coherence. High levels of noise (stationary or non-stationary) are also well handled as long as noise source is not within the SRP-PHAT's search region, given that this approach works under the assumption that the potential speaker is inside it.

In the context of multimodal approaches, many works are focused on the localization problem, using visual features mainly for finding faces and heads and leaving VAD mostly to audio cues. For instance, in the work of Wang and Brandstein (WANG; BRANDSTEIN, 1997) initial talker locations are estimated acoustically from microphone array data, based on the time-difference of arrival estimation followed by a triangulation procedure. The final location is obtained based on video cues (acquired with a single camera), using mostly motion and edge information. Their work was further extended in (WANG; GRIEBEL; BRANDSTEIN, 2000) by adding head pose estimation using multiple cameras and multi-channel speech enhancement techniques. In (GATICA-PEREZ et al., 2007), an array of microphones is combined with several cameras (frontal and top views of the participants), but visual information is used only to detect the shape and spatial structure of human heads. Talantzis and collaborators (TALANTZIS; PNEVMATIKAKIS; CONSTANTINIDES, 2009) proposed an approach that estimates independently the position of the active speaker in cluttered and reverberant environments using audio and video information, and combined both outputs. Their method was tested with a large microphone array (80 microphones located in different locations inside the acoustic enclosure and organized in different topologies), and a set of five synchronized and calibrated cameras. Blauth and colleagues (BLAUTH et al., 2012) explored the spatio-temporal coherence of the sound source for VAD, and used a face tracking to improve localization results.

In (ALMAJAI; MILNER, 2008) an MVAD algorithm has been proposed, where AVAD and VVAD are performed separetely using Gaussian Mixture Models and later fused by a SNR-based weighting approach. They use mono audio recordings with simulated white noise, showing good VAD accuracy, which is however affected when testing the algorithm on speakers different from those used for training it. Takeuchi et al. (TAKEUCHI S.; HAYAMIZU, 2009) use Mel-Frequency Cepstral Coefficients (MFCC) as audio features and optical flow of the speakers mouth as visual features through an multi-stream HMM system (approach similarly employed by speech recognition systems). Their algorithm shows good performance in a relatively controlled environment, that however decreases proportionally to noise levels in the audio modality. In (PETSATODIS; PNEVMATIKAKIS; BOUKIS, 2009) MFCCs are used in combination with a mouth opening measure to form the MVAD approach. Individual HMMs are used for each modality, generating confidence measures that are fused by a hierarchy scheme. If a face is recognized VVAD is performed to detect lip movements, which, if



(a) (b)

Figure 4.1: (a) Schematic representation of the proposed approach. (b) Our prototype system.

positive, is assisted by the AVAD. When lips are found to not be moving or no face is recognized, speech decision is delegated to the audio modality alone. While this approach presents good results, it may be highly degraded due to to false positives of the VVAD.

In this chapter, we present a new approach for lip motion extraction and VVAD. We study its accuracy when solely applied to the video data of multimodal recordings, as well as when combined with existing AVAD algorithm for performing MVAD on the same recordings. Next, we describe our approach for VVAD, we summarize the AVAD algorithm that is fused with our VVAD approach, and finally describe the MVAD integration approach.

4.3 Our Approach

Our setup is similar to the one described in (BLAUTH et al., 2012). We have a linear array of eight microphones and an off-the-shelf webcam placed at the center of the array. The participants are facing the camera and the array, and are expected to be encountered at approximate distance D from the array, as illustrated in Fig. 4.1.

A face detector/tracker (BINS et al., 2009) is used to extract the faces of the participants at each frame, and the lips region of each participant is extracted and used to infer voice activity, as explained next. The AVAD approach explores an array of microphones (BLAUTH et al., 2012) to find spatio-temporal coherent sound sources within the search region where the participants are expected. Both VVAD and AVAD procedures produce a fuzzy answer (value between 0 and 1), which can be used independently to detect voice activity. The output of these answers are then fed to a new classifier that produces the final MVAD result based on audiovisual cues. The proposed approach is detailed next.

4.3.1 Visual Voice Activity Detection (VVAD)

Our VVAD technique consists of lip detection based on color information, and the exploration of the temporal evolution of the extracted lips' movement through HMMs in order to detect voice activity or silence. It was inspired in (LOPES et al., 2011), including a color correction scheme to improve the detection of the lips region, and it is described next.

4.3.1.1 Skin Detection

In our approach, skin detection is performed using prior probability Gaussian mixture models for the skin (YANG; AHUJA, 1999), hair, and background classes, based on the 'a' and 'b' channels of the CIELAB color space. These models will be used later to classify pixels in the skin, hair, and background classes in face image sequences.

As mentioned above, the skin color probabilistic model is based on the 'a' and 'b' channels of CIELAB color space. The mixing parameters of the *a priori* probability representing the pixel colors in each class (i.e. mean vectors, covariance matrices, and priors of the skin, hair, and background classes) are estimated using the Gaussian Mixture Modeling method proposed in (FIGUEIREDO; JAIN, 2002). Given the class probability density function (PDF), the Bayes rule (WEBB, 2002) can be used for assigning the image pixels to the above mentioned classes:

$$p(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)p(\omega_j)}{p(\mathbf{x})}, \tag{4.1}$$

where ω_j is the j^{th} class, $j = 1, ..., n$; $p(\omega_j|\mathbf{x})$ is the posteriori probability; $p(\mathbf{x}|\omega_j)$ is the *a priori* pixel probability modeled by the Gaussian mixture; $p(\omega_j)$ is the class prior probability estimated from the number of samples per class in the training set; and $p(\mathbf{x})$ is the evidence probability based on the complete training data set. For a given class ω_j, the *a priori* probability of the Gaussian mixture is

$$p(\mathbf{x}|\omega_j) = \sum_{i=1}^{g} \pi_i \cdot p_i(\mathbf{x}; \mathbf{\Theta_i}), \tag{4.2}$$

where g is the number of mixing components; π_i is the mixing parameter i, where $\sum_{i=1}^{g} \pi_i = 1$ and $\pi_i \geq 0$; $p_i(\mathbf{x}; \mathbf{\Theta_i})$ is the i^{th} Gaussian PDF; and $\mathbf{\Theta_i}$ denotes the parameters of the i^{th} Gaussian, namely, bi-variate mean vectors μ_i and covariance matrices Σ_i for $i = 1, ..., g$ (estimated using (FIGUEIREDO; JAIN, 2002)). Finally, a pixel is classified as belonging to a skin region using the Bayes rule (WEBB, 2002):

$$p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}), k = 1, ..., n; k \neq j. \tag{4.3}$$

4.3.1.2 Lip Detection

Prior to the lip detection step, we correct the image colors aiming to obtain intensity levels similar to those used in the training data. This way, effects arising from the variation of illumination and camera-specific characteristics can be conpensated. Our approach is based on the diagonal model in (HORDLEY et al., 2005). More precisely, to perform the color correction to an input image, we first assume a reference one that is obtained from our training set. We treat the input image as a version of the reference one, having the same illuminant, but that has been previously subjected to intensity changes (e.g. due to motion within the room). This way, the color transformation $t_c = \frac{\mu_c'}{\mu_c}$ is applied to the input image, where c denotes the channels R, G and B, μ_c and μ_c' are the means of the RGB channels of the input and the reference images, respectively, and t_c is the illuminant transformation for each pixel, such that $c^{'} = ct_c$ is the corrected pixel color.

For detecting the lips, we may reduce the number of potential mouth-related pixels by priorly detecting the skin regions within the recognized speaker's face. Eq. (4.3) allows us to perform such skin detection, generating a binary mask for those pixels. In some cases, however, this proccess may end up not identifying some of the lip pixels (which

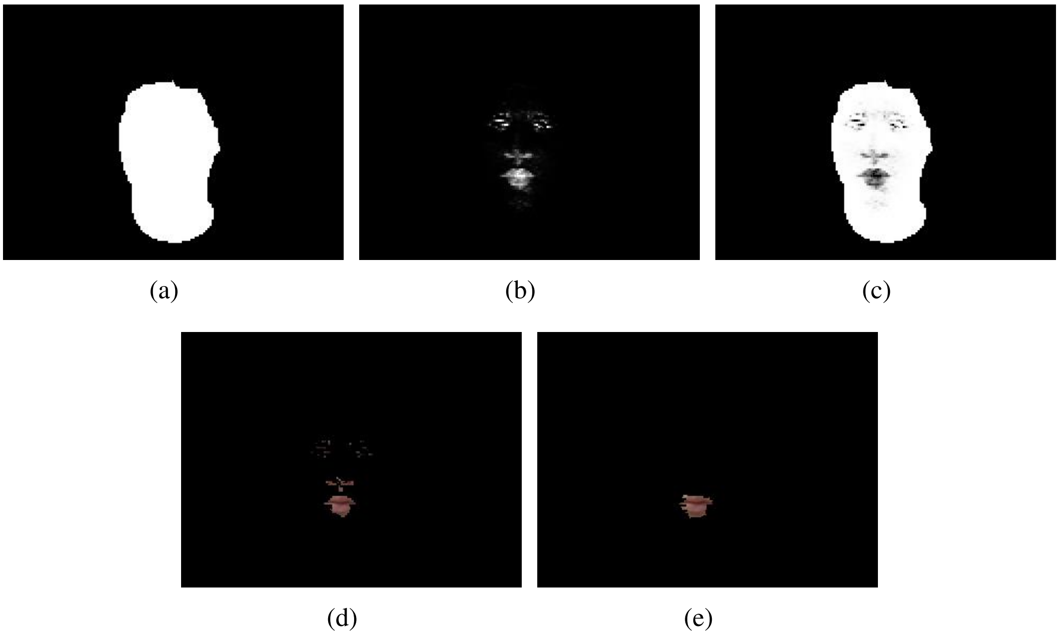


Figure 4.2: (a) binary mask corresponding to the skin regions (i.e. face) in *(b)*; (b) $p(\omega_{lip}|\mathbf{x})$; (c) $p(\omega_{non-lip}|\mathbf{x})$; (d) initial lip regions. (e) morphological post-processing

should be, since we treat them as skin too). For this reason, to ensure that mouth pixels are included in the mask, we apply a morphological closing with a circular dilatation operator of 15 pixels, and a circular erosion operator of 23 pixels within the mask (these parameters were determined experimentally for a 320×240 image). As an illustration, the binary mask corresponding to Fig. 4.2(b) is shown in Fig. 4.2(a).

Having this binary mask of skin pixels (within which lie the lip pixels), we use the Bayes rule in Eq. (4.3) to perform the lip detection. For this, we first create an *a priori* bivariate probability model for the lips using Eq. (4.2), similarly to the skin model. The parameters of the GMM are the normalized values, between $[0, 255]$, of the the Hue channel (HSV color space) and 'A' channel (CIELAB color space) of the lip-related pixels in our training samples, and the mixing parameters π_i are again estimated through the approach in (FIGUEIREDO; JAIN, 2002). We then assign the pixels within the skin binary mask to the lip class based on the values of $p(\omega_{lip}|\mathbf{x})$ and $p(\omega_{non-lip}|\mathbf{x}) = 1 - p(\omega_{lip}|\mathbf{x})$. The $p(\omega_{lip}|\mathbf{x})$ and $p(\omega_{non-lip}|\mathbf{x})$ values for the skin binary mask in Fig. 4.2(a) are illustrated in Figs. 4.2(b) and 4.2(c), respectively. Finally, $p(\omega_{lip}|\mathbf{x})$ is discretized in 255 values, and lip regions are segmented as extended objects using Otsu's technique (OTSU, 1979), as illustrated in Fig. 4.2(d). It can be observed that most of the lips regions are effectively segmented using the proposed color-based approach, but false positives often occur around the nose and eyes region. To refine the lip detection, some morphological post-processing steps are applied. A closing operator with a 2×5 rectangular structuring element (determined experimentally for frames with 320×240 pixels) is used to fill small gaps inside the mouth, and the largest remaining connected component is detected as the lips. The result of the post-processing step is shown in Fig. 4.2(e).

4.3.1.3 Visual Voice Activity Detection

The proposed approach explores the expected movement of the lips during speech periods to distinguish silence from speech, using two competing HMMs (one for silence and another for speech), as described next.

An HMM can be used to model dynamic systems that may change their states in time (RABINER, 1989). A model with N states and a discrete observable variable (with M possible values) is characterized by $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, where $\mathbf{A} = [a_{ij}]$ for $1 \leq i, j \leq N$ is the transition matrix that contains the probabilities of state changes, $\mathbf{B} = [b_{ij}]$ for $1 \leq i \leq N, 1 \leq j \leq M$ describes the discrete PDF of the observable for each state i, and $\pi = [\pi_i]$ for $1 \leq i \leq N$ contains the initial probabilities of each state.

In this work, we use two competing HMMs for the VVAD task, one for characterizing speech situations and other for silence situations. The speech HMM is described by $\lambda^{\text{v}}_{\text{sp}} = (\mathbf{A}^{\text{v}}_{\text{sp}}, \mathbf{B}^{\text{v}}_{\text{sp}}, \pi^{\text{v}}_{\text{sp}})$, and the silence HMM by $\lambda^{\text{v}}_{\text{si}} = (\mathbf{A}^{\text{v}}_{\text{si}}, \mathbf{B}^{\text{v}}_{\text{si}}, \pi^{\text{v}}_{\text{si}})$, where the $^{\text{sp}}$ and $^{\text{si}}$ subscripts denote speech and silence, respectively, and the $^{\text{v}}$ superscript denotes video, and is used to differ between other similar variables that are related to audio. For both HMMs, we use $N^{\text{v}} = 2$ hidden states, which are related to the current status of the mouth: open or closed. The observable variable $O^{\text{v}}(t)$ is the estimated height of the mouth at each frame t, computed using the color-based lip extraction algorithm previously described (Section 4.3.1.2) and is discretized into M^{v} values.

Given the speech HMM $\lambda^{\text{v}}_{\text{sp}}$ and the silence HMM $\lambda^{\text{v}}_{\text{si}}$, and given a sequence of observations $\mathcal{O}^{\text{v}}_t = \left\{O^{\text{v}}(t-T), O^{\text{v}}(t-T+1), ..., O^{\text{v}}(t)\right\}$ within a time window of size T, we can compute how well each HMM represents $\mathcal{O}^{\text{v}}_t$. More precisely, this can be done by computing $P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{sp}})$ and $P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{si}})$ using the forward-backward procedure (RABINER, 1989). Hence, based on this temporal window of size T, a given frame t may be classified as silence if $P(\mathcal{O}^{\text{v}}_t; \lambda_{\text{sp}}) < P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{si}})$, and classified as speech otherwise. Based on this approach, we can also compute a kind of posterior VVAD probability value between 0 and 1, instead of binary classification result:

$$P(\text{VVAD}|\mathcal{O}^{\text{v}}_t) = \frac{P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{sp}})}{P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{sp}}) + P(\mathcal{O}^{\text{v}}_t; \lambda^{\text{v}}_{\text{si}})}. \tag{4.4}$$

This allows us to use the video-based HMMs competition scheme to generate a normalized feature for a further multimodal classifier (process described in Section 4.3.3).

For choosing the appropriate values of $\lambda^{\text{v}}_{\text{sp}}$, $\lambda^{\text{v}}_{\text{si}}$, T and M^{v}, we have used the following process. For the size of the discrete time window T, larger values tend to increase the temporal coherence of the results, but also produce larger delays to detect silence-to-speech or speech-to-silence changes (since such transitions present observations used in both, speech and silence HMMs). On the other hand, smaller values for T lead to smaller delays in detecting transitions, but make the system more susceptible to noise. In this work, we used $T = 10$, which corresponds to 1 second for video sequences acquired at 10 frames per second. For choosing the number of possible observables, we set $M^{\text{v}} = 10$, which experimentally has shown to be enough for representing different levels of confidence of our mouth height measure. Larger M would increase the computational complexity of the algorithm at no advantage in accuracy, and smaller values would reduce the representativeness of our visual feature. To obtain the parameters of both HMMs, ground truth video sequences were used, where each frame was manually labeled as silence or speech. Each set of T adjacent frames marked as speech was used to build a training dataset for the speech HMM, and an analogous training dataset was created for the silence HMM. The Baum-Welch algorithm (RABINER, 1989) was then applied independently to each dataset to obtain matrices $\mathbf{A}$ and $\mathbf{B}$ for each model, while the initial probability vector π was considered uniform.

4.3.2 Audio Voice Activity Detection

In this work, AVAD is performed by evaluating the spatio-temporal coherence of the sound-source candidate through competing HMMs, similarly to (BLAUTH et al., 2012). The main rationale is that an active sound source should produce a peak in the SRP-PHAT that is both temporally and spatially coherent to the expected behavior of a human speaker, whereas sound absence should reflect as random global maxima in the SRP-PHAT as a result of the background audio noise and reverberations.

Given this idea, we start by computing the SRP-PHAT $P(\mathbf{q}_i)$, described in (DIBIASE, 2000), for a set of equidistant points $\mathbf{q}_i$ in a search space. The HMMs are modeled as having N^{a} hidden states, where each state at time t represents one of the discretized spatial position $\mathbf{q}_i$. The observable of the HMMs is a two-dimensional vector $\mathbf{O} = (O_1, O_2)$ given by

$$O_1 = \underset{i}{\text{argmax}}\, P(\mathbf{q}_i), \;\; O_2 = \frac{\max P(\mathbf{q}_i)}{\sum_{i=1}^{N} P(\mathbf{q}_i)}, \tag{4.5}$$

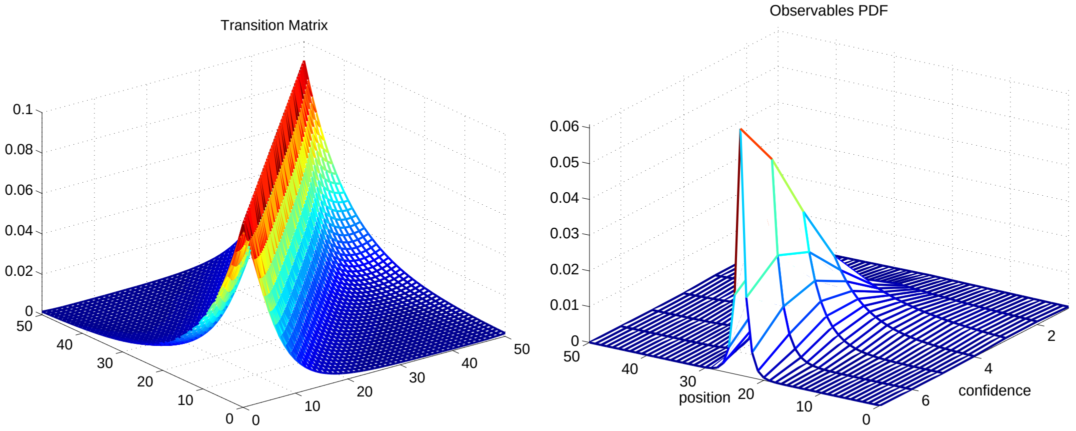
where O_1 is exactly the discretized position that produces the SRP-PHAT's maximum, and O_2 can be viewed as a confidence measure when estimating O_1, and is discretized into Q values, so that $M^{\text{a}} = QN^{\text{a}}$. For active speech, O_1 should provide the correct location of the speaker, and O_2 tends to be large (since the global maximum is expected to be considerably larger than the mean). On the other hand, for silence, all values of $P(\mathbf{q}_i)$ tend to be similar making O_2 smaller, and O_1 should arise in a random $\mathbf{q}_i$.

For the parameters defining $\lambda^{\text{a}}_{\text{sp}}$, we have chosen, as in (BLAUTH et al., 2012), parametric distributions instead of trained models (like the video approach). The chosen PDF for describing $\mathbf{B}^{\text{a}}_{\text{sp}}$ is the following:

$$p_i^{\{\text{sp,si}\}}(O_1, O_2) = p_i^{\{\text{sp,si}\}}(O_1|O_2)p^{\{\text{sp,si}\}}(O_2), \tag{4.6}$$

where $p(O_2)$ is the distribution of O_2 which is independent of the state i, and $p_i(O_1|O_2)$ is the conditional probability of O_1 given O_2, which is strongly affected by i. The choices for $p_i^{\{\text{sp,si}\}}(O_1|O_2)$ and $p^{\{\text{sp,si}\}}(O_2)$ are the same as in (BLAUTH et al., 2012), and their main characteristics are briefly described below.

When there is an active speaker, we expect a predominant peak of the SRP-PHAT values, implying in the observation of large values for O_2. For this reason, $p^{\text{sp}}(O_2)$ is responsible for implementing this effect in $\mathbf{B}^{\text{a}}_{\text{sp}}$, that is, increasing $p_i^{\{\text{sp,si}\}}(O_1, O_2)$ as function of O_2. This is done by modeling $p^{\text{sp}}(O_2)$ as discrete Gaussian centered at the maximum value O_2^{max}. Additionally, if a given state i is in fact the actual position of the active speaker (in speech situation), the observable with the highest probability of occurring is $O_1 = i$, so that $p_i^{\text{sp}}(O_1|O_2)$ is expected to present peak at $O_1 = i$. If the confidence O_2 is large, the probability should decay rapidly as O_1 gets far from i. If O_2 is smaller, the decay should be smoother, since the confidence is smaller and other values of O_1 are also expected to be encountered with higher probability. On the other hand, during silence periods we expect to find smaller values for O_2, since the SRP-PHAT values tend to be nearly homogeneous. Similarly to $p^{\text{sp}}(O_2)$, a discrete Gaussian was also used to model $p^{\text{si}}(O_2)$, but now centered at the smallest possible value O_2^{min} for observable O_2. Additionally, during silence situations the location of the SRP-PHAT peak is expected to be random, so that a uniform distribution is chosen for $p_i^{\text{si}}(O_1|O_2)$. For sake of illustration, the speech observation matrix is graphically shown in Figure 4.3(b), for $\mathbf{q}_{25}$, $N^{\text{a}} = 51$ and $Q = 7$.



(a) (b)

Figure 4.3: Illustration of (a) the transition matrix and (b) probability distribution function of observables ($i = 25$, $N^{\text{a}} = 51$ and $Q = 7$), for the speech-related HMM.

For the state transition matrix $\mathbf{A}^{\text{a}}_{\text{sp}}$, we prioritize the maintenance of the current state or changes to neighboring states, and penalizes changes to states far away from each other (recall that each state is a discretized SRP-PHAT position $\mathbf{q}_i$). This helps our model explore the fact that active and coherent speech sources are expected to be found, at each frame t, not too far from its previous location, in frame $t-1$. However, during silence periods, an opposite reasoning can be used (random located peaks, thus no spatio-temporal coherence, as previously mentioned). This allows us to define an uniform distribution of the transition matrix $\mathbf{A}^{\text{a}}_{\text{sp}}$, which is graphically shown in Figure 4.3(a), also for $N^{\text{a}} = 51$.

Finally, VAD is decided based on a time interval of T samples, similarly to the VVAD approach. More precisely, given a sequence of observations $\mathcal{O}^{\text{a}}_t = \left\{\mathbf{O}^{\text{a}}(t-T), \mathbf{O}^{\text{a}}(t-T+1), ..., \mathbf{O}^{\text{a}}(t)\right\}$ within a time window with size T containing the current observation $\mathbf{O}^{\text{a}}(t)$ and the previous $T-1$ observations, we compute how well each HMM describes $\mathcal{O}^{\text{a}}_t$. In other words, we compute $P(\mathcal{O}^{\text{a}}_t; \lambda^{\text{a}}_{\text{sp}})$ and $P(\mathcal{O}^{\text{a}}_t; \lambda^{\text{a}}_{\text{si}})$ and then a posterior audio-based VAD probability:

$$P(\text{AVAD}|\mathcal{O}^{\text{a}}_t) = \frac{P(\mathcal{O}^{\text{a}}_t; \lambda^{\text{a}}_{\text{sp}})}{P(\mathcal{O}^{\text{a}}_t; \lambda^{\text{a}}_{\text{sp}}) + P(\mathcal{O}^{\text{a}}_t; \lambda^{\text{a}}_{\text{si}})}. \tag{4.7}$$

It is important to note that visual data (information about the location of the faces) was used in (BLAUTH et al., 2012) to only improve sound source localization, leaving VAD for the audio modality, while in this work we use visual data to form our final MVAD algorithm. Next, we present our audiovisual approach for VAD.

4.3.3 Audiovisual VAD

Each of the approaches described above (VVAD and AVAD) may be used independently to detect voice activity, as previously mentioned. However, there are situations that are very challenging to each modality (audio or video), which would make an unimodal approach not robust to adverse situations. For instance, when the mouth is covered or when the lips moves without speech (e.g., smile or yawn), the VVAD approach tends to fail. When there is a localized sound source (e.g. a cell phone ringing) within the search range of the SRP-PHAT, the AVAD approach tends to fail. Hence, a joint audiovisual approach is expected to produce better results than using audio and video cues individually.

Fusion of multiple modalities can be performed at the feature level (early fusion) or at the classification level (late fusion) (ATREY et al., 2010), and in this work we employ the latter approach. The main reason for this choice is that each classifier (audio and video) provides a normalized response that can be related to a confidence of the detection. Therefore, having a later fusion approach can be easily modified to accommodate changes in either the VVAD or the AVAD algorithms, whereas the early fusion approach depends heavily on the chosen features.

To build our audiovisual voice activity detector, we initially created a set of video sequences acquired with a single monocular camera and an array of microphones. These sequences contain different participants and some conditions that may degrade the analysis of the audio or video information (this is detailed in Section 4.4). Audio information was captured at 44100 Hz, and video information captured at 10 frames per second (FPS) with a resolution of 640×480 pixels for some videos, and 960×720 for others. In total, 5112 multimodal frames were acquired, where each frame corresponds to one image (video frame) and to 4096 audio samples (amount used in the SRP-PHAT algorithm at each time t). All frames were manually ground truthed, so that we know if there is an active speaker or not, and also the location of the speaker when there is speech.

Having the ground truth of all 5112 multimodal frames, we then computed their audio and visual features (Eq. (4.7) and Eq. (4.4)) in order to train a supervised classifier. Since the literature is vast regarding supervised classifier, and many different techniques could suite our two-dimensional feature space, we have explored some possible algorithms using the machine learning software Weka, which is fully described in (HALL et al., 2009). We have tested decision trees such as the C4.5 (QUINLAN, 1993), Functional Trees (GAMA, 2004), Alternating Decision Trees (FREUND, 1999); neural networks, such as Multilayer Perceptron (RUSSELL; NORVIG, 2003), Voted Perceptron (FREUND; SCHAPIRE, 1998); two Support Vector Machine (SVM) algorithms, (PLATT, 1999) and (SHALEV-SHWARTZ; SINGER; SREBRO, 2007); decision rule algorithms, like decision tables (DT) (KOHAVI, 1995), hybrid DT/naive Bayes (HALL; FRANK, 2008) and one rule classifier (HOLTE, 1993).

Among all tested supervised classifiers, we chose the C4.5 decision tree (named J48 in Weka), due to showing good classification results, as some other algorithms, but also at the faster classification speed. In particular, the C4.5 algorithm uses the concept of information gain for choosing, at each tree level, the attribute that best splits the training set into successively more homogeneous sub-datasets. By assigning, to a decision node, an attribute with maximum information gain, the C4.5 algorithm will create a leaf (class label) when the entropy of a sub-dataset is zero (maximum homogeneity). After the final decision tree is built, it is pruned, decreasing the number of tests needed for classifying a sample. Reducing the tree's size also helps avoiding overfitting of the training set, which is a common problem when small a dataset is used for training.

Figure 4.4 shows an example of a C4.5 decision tree that was built (and pruned) using our labeled multimodal frames. The attributes of Eqs. (4.4) and (4.7) are respectively denoted as "vvad_spe_prob" and "avad_spe_prob", and the class labels are called "speech" and "silence".

4.4 Experimental Results

In this section we present a classification accuracy comparison of different algorithms. We use individually our VVAD approach, our AVAD approach, the multimodal approach, as well other state-of-the-art algorithms for AVAD. The multimodal recordings consist of

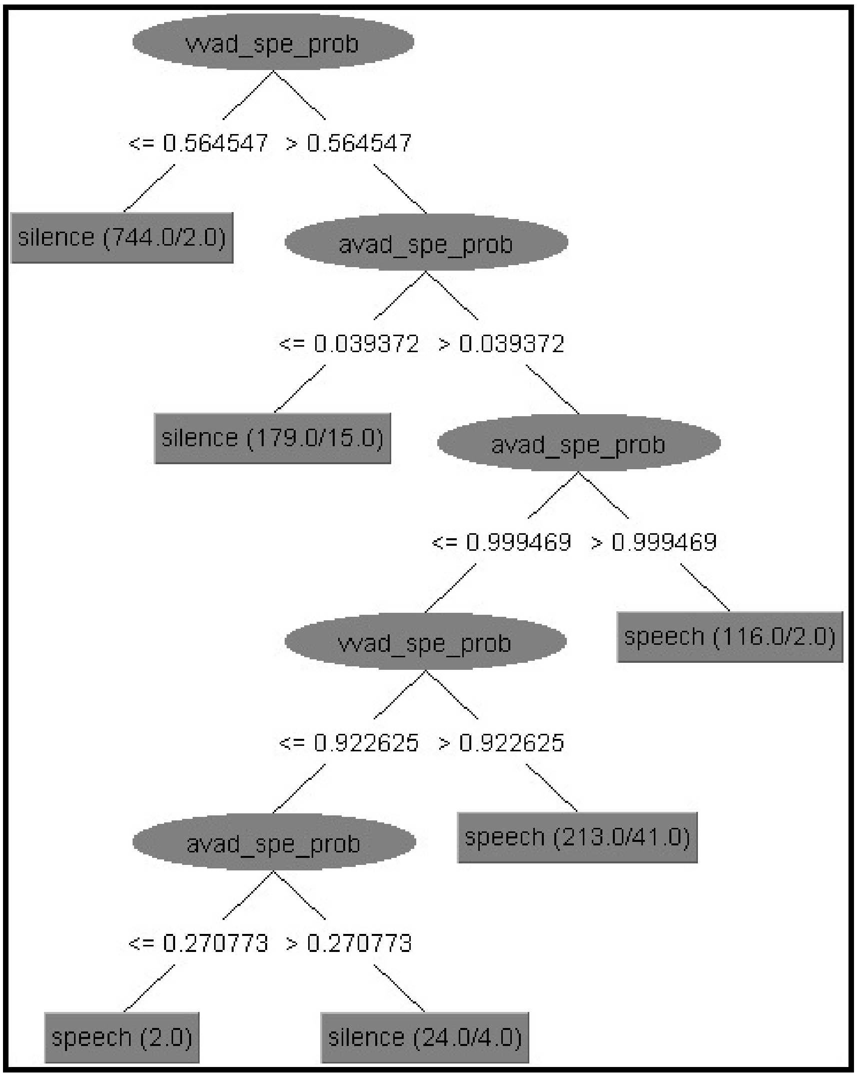


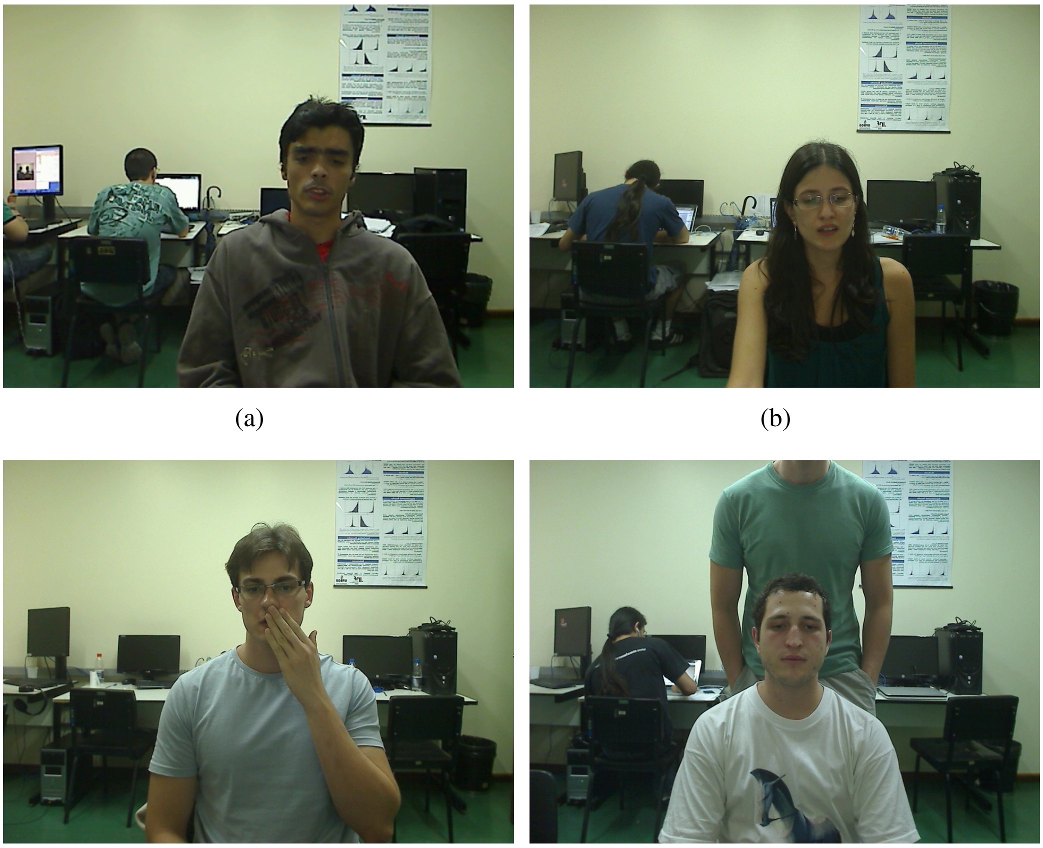
Figure 4.4: Pruned C4.5 decision tree classifier generated by supplying our labeled multimodal frames to the training process.

one-speaker scenarios, with varying audio background noise. Noise situations were generated by real-world sources such as competing speech (people talking in background), typing and clicking sounds, air-conditioner on, door slams etc. This makes our test scenario challenging and realistic given that the noise is non-stationary and in some cases reaches 0 dB SNR. All eight recordings are one minute long, with approximately equal amount of speech and non-speech moments[1]. The speakers alternate between 10 seconds of non-speech and 10 seconds of speech, starting with non-speech. During speech moments, they produce small speech hiatuses, which is common during normal voice activity, and they were marked as voice activity when generating ground truth data. Additionally, in two of those recordings, the speaker purposely imposes a scenario that would likely fool the AVAD and the VVAD (one recording for each detector), so that we can evaluate if the proposed MVAD fusion approach is in fact keeping an overall good result even under failure of one of the modalities.

Sequences called **Normal 1** through **Normal 4** were recorded by male individual

[1] The referred dataset (including the ground truths) can be found at `http://www.inf.ufrgs.br/~crjung/mvad/mvad.htm`.

speakers, and sequences **Normal 5** and **Normal 6** by female speakers. Normal sequences refer to the recordings where the participants behaved normally, as one would do in a regular conversation (but still having background noises, such as those previously mentioned). Recordings named **Fooling AVAD** and **Fooling VVAD** were created to purposely impose challenges to the audio-based and video-based approaches, respectively. For misleading the VVAD, the participant moved his mouth during non-speech moments (e.g. by smiling and yawning), and generating occlusions in the mouth region when speaking. For tricking the AVAD, the participant of interest remained in silence while another person nearby kept speaking (recall that our approach implicitly selects the participants of interest as a consequence of the SRP-PHAT's search region). Figure 4.5 illustrates, for four of the recordings, an image frame of the speakers' activity.



(a) (b)

(c) (d)

Figure 4.5: Illustration of the recorded sequences: (a) **Normal 1**, (b) **Normal 5**, (c) **Fooling VVAD**, (d) **Fooling AVAD**

Table 4.1 shows a comparison of different VAD approaches for the **Normal** recordings (individually and mean of all of them), to which we refer using the **Nor.** acronym. We have used our VVAD approach described in section 4.3.1.3, our AVAD approach described in section 4.3.2, the multimodal approach described in 4.3.3 and the AVAD approaches of some works previously mentioned: Sohn's algorithm (SOHN et al., 1999) and the Energy-Entropy (EE) approach (LEE; MUHKERJEE, 2010a). As it can be observed, AVAD approaches that rely mostly on signal/noise discrimination based on intensities (as (SOHN et al., 1999)) produced values as low as 55.87%. This is actually expected, since the SNR of some recordings is low, and the noise is non-stationary, making the estimated noise level to be inaccurate. On the other hand, our AVAD produces higher accuracy rates, since it relies on the spatio-temporal continuity of the sound source. Furthermore, the proposed MVAD increased the accuracy rates in most cases, achieving a rate above 85% in all videos.

Table 4.1: Classification accuracy of five unimodal approaches as well as the proposed multimodal fusion approach.

	Nor. 1	**Nor. 2**	**Nor. 3**	**Nor. 4**	**Nor. 5**	**Nor. 6**	**All**
AVAD (Ours)	94.37%	89.20%	81.69%	91.71%	88.26%	95.31%	90.09%
VVAD (Ours)	90.61%	80.13%	72.61%	81.85%	85.92%	69.80%	80.15%
MVAD (Ours)	94.05%	92.49%	85.45%	97.03%	89.67%	95.46%	92.07%
AVAD (EE)	90.61%	81.22%	65.10%	68.23%	85.76%	73.55%	77.41%
AVAD (Sohn)	91.71%	88.26%	56.18%	58.22%	84.98%	71.99%	75.22%

Table 4.2 shows the analysis for the two sequences created for fooling the AVAD and VVAD approaches. As expected, all AVAD approaches present low detection rates for recording **Fooling AVAD**, and the VVAD approach presented low accuracy for recording **Fooling VVAD**. However, the proposed MVAD present a high detection rate for both scenarios (above 93%), indicating that it indeed preserves good qualities of each modality. Table 4.2 also presents the average detection rates for all videos and for normal videos (repeating for easier comparison). As it can be observed, the average accuracy of our MVAD approach is above 91% when considering all videos.

Table 4.2: Classification accuracy for (1) the “fooling” recordings, (2) all normal recordings, (3) and all recordings.

	Fooling VVAD	**Fooling AVAD**	**All Videos**	**All Normal Videos**
AVAD (Ours)	91.71%	52.74%	85.62%	90.09%
VVAD (Ours)	58.93%	91.08%	80.61%	80.15%
MVAD (Ours)	93.87%	96.09%	91.82%	92.07%
AVAD (EE)	78.09%	64.48%	75.88%	77.41%
AVAD (Sohn)	75.12%	55.56%	72.75%	75.22%

Results shown so far were created by running a cross-validation method with ten folds, and taking the average of all folds. The pool used to extract training and test samples contain audiovisual frames for all the recordings, so one could claim that the trained classifier is biased to a specific scenario (since its training probably contains frames from the same speaker used for testing the approach). To evaluate the invariance of the MVAD classifier, we performed a second training procedure using only a subset of recordings to train the classifier, and tested using all of them. Table 4.3 shows the resulting accuracies using such procedures, where only videos shown in italic were used to train the model. As expected, accuracies for videos using both in training and testing were higher, but the MVAD results for recordings that were not present in the training procedure were also high (the lowest value accuracy was 85%). The average accuracy for all videos was

91.52%, and the average for the videos that were not present in the training dataset was 89.90%.

Table 4.3: Individual classification accuracy of our MVAD approach, for all recordings, and using the videos in italic for building the decision tree.

Sequence Name	MVAD (Our Approach)
Normal 1	94.84%
Normal 2	90.92%
Normal 3	85.13%
Normal 4	95.93%
Normal 5	88.89%
Normal 6	94.68%
Fooling VVAD	94.37%
Fooling AVAD	93.27%
All Videos	92.10%
All Normal Videos	91.52%

In an overview of our experiments, we may notice that the MVAD approach presents better classification accuracy than the AVAD or VVAD approach. In other words, fusing our proposed VVAD method to the AVAD work in (BLAUTH et al., 2012) has provided reasonable overall VAD accuracy. Additionally, the multimodal detector provides robustness when adverse conditions arise to one of the modalities (which are the case of most realistic environments). More important is that, although classification step we have used is simple, it earns good results, implying that the extracted audio and video features alone are robust, such that many kinds of classifiers may produce good results.

4.5 Conclusions

This work has presented a new approach for visual voice activity detection, which has indicated to be well suited for fusion with an AVAD technique, forming a robust multimodal VAD approach. We have run tests using multimodal sequences recorded with a monocular camera and an eight-sensor microphone array, which have shown good overall performance under normal realistic environments, and also when purposely corrupting either the audio or the video modality. We explored different algorithms using the well-known machine learning software Weka, finding that many supervised classifiers perform well using the proposed audio and video features, which, per se, also highlights the features' robustness. For our quantitative tests, we have manually ground truthed our recordings and chose the C4.5 decision tree algorithm for the classification step. Our VVAD approach consists of exploring the movement of the speaker's lips through the height of the mouth, which is found by our color-based segmentation approach; a competing HMMs scheme is then used for extracting a normalized VVAD probability. For the AVAD part, we also compute a normalized AVAD feature through a similar HMMs competition scheme that explores the spatio-temporal coherence of the sound source through the SRP-PHAT SSL method, with microphone arrays (BLAUTH et al., 2012).

References

See the unified bibliography of the dissertation.

5 SIMULTANEOUS SPEAKER VOICE ACTIVITY DETECTION AND LOCALIZATION USING MID FUSION OF SVM AND HMMS

Minotto, V.P.; Jung, C.R.; Lee, B.; "Simultaneous Speaker Voice Activity Detection and Localization Using Mid Fusion of SVM and HMMs", Submitted to IEEE Transactions on Multimedia on June 2013
Keywords:Multimodal Fusion, Voice Activity Detection, Sound Source Localization, Hidden Markov Model, Support Vector Machine, Optical-Flow, beamforming, SRP-PHAT

Abstract

Humans can extract speech signals that they need to understand from a mixture of background noise, interfering sound sources, and reverberation for effective communication. Voice Activity Detection (VAD) and Sound Source Localization (SSL) are the key signal processing components that humans can do by processing sound signals received at both ears, sometimes with the help of visual cues by localizing and observing the lip movements of the speaker that they listen to. Both VAD and SSL serve as the crucial design elements for building natural Human Computer Interface (HCI) applications involving human speech, such as speaker identification and speech recognition. The design and implementation of robust VAD and SSL algorithms in practical acoustic environments are still challenging problems, particularly when multiple simultaneous speakers exist in the same audiovisual scene. In this work we propose a multimodal approach that uses Support Vector Machines (SVMs) and Hidden Markov Models (HMMs) for assessing the video and audio modalities through an RGB camera and a microphone array. By analyzing the individual speakers' spatio-temporal activities and mouth movements, we propose a mid-fusion approach to perform both VAD and SSL for multiple active and inactive speakers. We tested the proposed algorithm in scenarios with up to three simultaneous speakers, showing an average VAD accuracy of 95.06% and average error of 10.9 cm when estimating the three-dimensional locations of the speakers.

5.1 Introduction

When a computer is used for a generic task, a mouse and a keyboard are commonly employed as the primary human-machine interfaces. Although being popular, they are not as natural as human-to-human interactions and may not be adequate for a variety of applications. Other kinds of interfaces can potentially be more promising such as touch or gestures. Additionally, thanks to the advancement of computing resources even on mobile platforms, more sophisticated human-computer interfaces (HCI) are becoming more viable and desired (JAIMES; SEBE, 2007), particularly those that allow human-to-human-like interactions, such as speech. One of the main problems with speech-based HCI, such as Automatic Speech Recognition (ASR), is that practical acoustic environments often include factors that significantly compromise the recognition accuracy of human speech, such as noise, reverberation, and competing sound sources. It is important to preprocess received signals to extract clean speech signals from such degradations as an input to ASR systems. Voice Activity Detection (VAD) and Sound Source Localization (SSL) are the most important examples of such front-end techniques. The main goal of VAD is to distinguish segments of a signal that contain speech from those that do not, such that a speech recognizer can process only the segments that contain voice information. In addition, speech enhancement algorithms (EPHRAIM, 1992) can benefit from the output of the VAD because accurate noise estimation is crucial and is often updated during noise only observations (EPHRAIM; MALAH, 1985). In SSL, the main goal is to identify the location of the active sound source, so that it is possible to enhance its speech signal using spatial filtering techniques such as beamforming with microphone arrays (BRANDSTEIN; WARD, 2001).

Most VAD and SSL approaches for HCI only consider single speaker scenarios. For applications such as videoconferencing or gaming, it is often desired to distinguish different speakers that may be speaking simultaneously, and algorithms designed for single speaker cases may not be suitable for such applications. Some recent works have proposed techniques for simultaneous speaker VAD (MARABOINA et al., 2006; BERTRAND; MOONEN, 2010; LORENZO-TRUEBA; HAMADA, 2010; DO; SILVERMAN, 2010; ZHANG; RAO, 2010) relying solely on the acoustic modality. While audio-only-based techniques might present promising results, leveraging from visual information is often beneficial when a video camera is available. In this context, other studies use the joint (multimodal) processing of both image and audio (ASOH et al., 2004; BUTKO et al., 2008; ALMAJAI; MILNER, 2008; PETSATODIS; PNEVMATIKAKIS; BOUKIS, 2009). The main idea is that by fusing more than one data modality it is possible to exploit the correlation among them in such a way that one modality compensates for the flaws from the others, making the algorithm more robust in adverse situations, especially with competing sources.

Our proposed work performs VAD and SSL for simultaneous speaker scenarios, using audio and video modalities. In order to process the visual information, we use a face tracker to identify potential sound sources (*speakers*) followed by optical-flow analysis of the users' lips with Support Vector Machines (SVMs) to determine whether or not that specific user is actively speaking. For the audio analysis, we use a Hidden Markov Model (HMM) competition scheme in conjunction with beamforming to individually evaluate the spatio-temporal behavior of potential speakers that are pre-identified by the face tracker. A mid fusion approach is proposed in between the visual VAD (VVAD) and audio VAD (AVAD) to construct our final multimodal VAD (MVAD). SSL is then performed for active speakers using information from the face tracker as well as the Steered Response Power with Phase Transform (SRP-PHAT) beamforming algorithm (BRANDSTEIN; WARD, 2001). Our experiments show an average accuracy of 95.06% and an average error of 10.9 cm when performing VAD and 3D SSL respectively, of up to three simultaneous speakers in a realistic environment with background noise and interfering sound sources.

The remainder of this chaper is organized as follows. Section 5.2 summarizes some of the microphone array techniques used in our work followed by the most recent research in the field of VAD and SSL. In Section 5.3, the proposed approach for multimodal VAD and SSL is presented. Section 5.4 presents the experimental evaluation of our technique, and conclusions are drawn in Section 5.5.

5.2 Related Work and Theoretical Overview

Typical existing VAD techniques are based on voice patterns in the frequency domain, pre-determined (or estimated) levels of background noise (SOHN et al., 1999), or zero crossing rate (TANYER; OZER, 2000). These approaches, however, do not tend to perform well in the multiple speaker scenario since simultaneous speech reflects as overlapped signals in the time-frequency plane. Most approaches (as in this work), therefore, use microphone arrays due to its capability of exploiting spatial characteristics of the acoustic signals such as through beamforming (TAGHIZADEH et al., 2011) or Independent Component Analysis (ICA) aided by beampattern analysis (MARABOINA et al., 2006). Furthermore, the addition of visual information is highly beneficial, since speech-related visual features are invariant to the number of simultaneous active sound sources.

Another consequence of the simultaneous sources case is that both VAD and SSL eventually become the same problem. When extending VAD from single to multiple sources, for example, the microphone array can be employed so that different speakers may be separately analyzed. In these cases, SSL is inevitable for not only identifying the number of active sources, but also to detect which are the active ones among all possible candidates. Reciprocally, for extending SSL from single to multiple sources, VAD must be used for validating the located sound events, so that noise sources are not accounted for as active speakers.

5.2.1 Related Work

In the work of Maraboina et al. (MARABOINA et al., 2006), frequency-domain ICA is used to separate the speech signals of different sound sources, and beampattern analysis is used to solve the permutation problem in the frequency components. Unmixed frequency bins are then separately classified using thresholding aided by K-means clustering. This approach, however, assumes that the number of sound sources is known, and it was only explored for two speakers scenario. Other approaches using ICA have also been proposed, such as (BERTRAND; MOONEN, 2010), where they claim reasonable VAD accuracy for simulated data.

In (YAMAMOTO et al., 2006), simultaneous VAD is performed for robot auditory system purposes as a preprocessing step for ASR. It is shown that by applying sound source localization through delay-and-sum far-field beamforming, it is possible to separate overlapping speech signals to the point that two sources are well detected. Their results are evaluated in terms of ASR accuracy, and are obtained in a fixed environment.

Gurban and Thiran (GURBAN; THIRAN, 2006) propose a supervised multimodal approach for VAD. They use the energy of the speech signal as audio feature and Optical Flow of the mouth region as visual feature. A Gaussian Mixture Model is trained from labeled data, and the Maximum Likelihood (ML) is applied to classify each data frame. Their approach does not deal with simultaneous speakers and it was tested in a controlled environment. The authors also mention that scenes having background movements may degrade the algorithm's performance, since no face detection/tracking algorithm is used. In (ASOH et al., 2004), background subtraction using stereo cameras is combined with expectation maximization (EM) using microphone array; a Bayesian network trained with a particle filter is used to estimate the direction of the sound sources.

In (MOHSEN NAQVI et al., 2012), multimodal analysis of multiple speakers is performed to tackle the similar problem of blind source separation. Multiple cameras are used by a three-dimensional (3D) face tracker to provide *a priori* information to a least-squares-based beamforming algorithm with a circular microphone array to isolate different sources. The separated audio sources are further enhanced by applying a binary time-frequency masking as a post-filtering process in the cepstral domain. Results are not shown in terms of VAD or SSL accuracy, for only speech separation is attempted, assuming the speakers are always active.

A different approach for joint VAD and SSL has been proposed in (TAGHIZADEH et al., 2011), where the steered response power with phase transform (SRP-PHAT) method is used to iteratively detect the speakers' locations. For distinguishing the sources, the SRP-PHAT's gradient is used to separate different speakers' regions in its power map. VAD is then automatically performed when the iterative algorithm finds the last speaker, which happens when the maximum's position corresponds to the SRP-PHAT's null point. This approach, however, is based on the assumption of diffusive noise and may cause false positives for directional noise such as door slams.

Other works also leverage from the proven robust SRP-PHAT algorithm (DIBIASE, 2000), such as by integrating it with clustering techniques in attempt to separate the maxima that belong to different speakers. Do et al. (DO; SILVERMAN, 2008) use Agglomerative Clustering (AC) with the Stochastic Region Contraction optimization method; they later propose Region Zeroing and Gaussian Mixture Models (DO; SILVERMAN, 2010), achieving up to 80% correct classification rate for two speakers case using a large-aperture microphone array; Cai et al. also explore AC, but with spectral sub-band SRP-PHATs (CAI; ZHAO; WU, 2010). Alternatively, in (BRUTTI; OMOLOGO; SVAIZER, 2008) (where the SRP-PHAT is named Global Coherence Field - GCF), multiple speakers are located by a de-emphasis approach of the GCF; the dominant speaker is localized and then the GCF map is modified by compensating for the effects from the first speaker and the position of the second speaker is detected. While this method can considerably increase the localization rate of the second speaker, its computational cost is very high, and the localization accuracy of the second speaker depends on that of the first one.

From the mentioned works, we may observe that many of the existing multi-source SSL/VAD algorithms extended the SRP-PHAT in a way that speakers other than the dominant one are localized and identified. In our work, we also use the SRP-PHAT for beamforming due to its robustness. Therefore, we briefly describe SSL using the SRP-PHAT algorithm in the next section.

5.2.2 Sound Source Localization using the SRP-PHAT

For an array of N_{mic} microphones, the signal $x_m(t)$ captured at the m^{th} microphone can be described using a simplified acoustic model (DIBIASE, 2000),

$$x_m(t) = \alpha_m s(t - t_m^{\mathbf{q}}) + u_m(t), \tag{5.1}$$

where $s(t)$ is the source signal, $u_m(t)$ represents the combination of reverberation, interferences, and background noise, and α_m and $t_m^{\mathbf{q}}$ denote the propagation attenuation and delay of the signal $s(t)$ from a source location $\mathbf{q}$ to the m^{th} microphone, respectively. Equivalently, Eq. (5.1) can be represented in the frequency domain as

$$X_m(\omega) = \alpha_m S(\omega) e^{-j\omega\tau_m^{\mathbf{q}}} + U_m(\omega), \tag{5.2}$$

where $\omega = \frac{2\pi f}{F_s}$ is the normalized frequency in radians corresponding to the frequency f (in Hz) of the continuous-time signal $x_m(t)$ that is sampled with the sampling rate of F_s Hz, and $\tau_m^{\mathbf{q}} = t_m^{\mathbf{q}} F_s$. We assume that the signal is sampled above the Nyquist rate.

Therefore, given a vector of Fourier transforms of observed signals, $[X_1(\omega), X_2(\omega), \cdots, X_{N_{\text{mic}}}(\omega)]$ for the normalized frequency ω, SSL may be seen as the problem of finding a source location $\mathbf{q}$ that satisfies some optimality criteria such as Maximum-Likelihood (ZHANG; ZHANG; FLORENCIO, 2007; LEE; KALKER, 2010) or maximum power of the filter-and-sum beamformer like the SRP-PHAT method (DIBIASE, 2000).

As previously mentioned, the SRP-PHAT is currently one of the state-of-the-art algorithms for SSL due to its robustness against noise and reverberation. It finds a source location by comparing, for a frame of data, the output energies of PHAT-weighted filter-and-sum beamformers of different potential sound source locations in a search region. The filter-and-sum beamformer steered at location $\mathbf{q}$ may be represented in the frequency domain as

$$Y(\omega, \mathbf{q}) = \sum_{m=1}^{N_{\text{mic}}} W_m(\omega) X_m(\omega) e^{-j\omega\tau_m^{\mathbf{q}}}, \tag{5.3}$$

where $W_m(\omega)$ denotes a generic weighting function applied to the m^{th} microphone's signal. When this weighting function is chosen to be the PHAT, that is, $W_m(\omega) = |X_m(\omega)|^{-1}$ we may define the SRP-PHAT of a point $\mathbf{q}$ by computing the energy of the PHAT-weighted filter-and-sum of that point. Using Parseval's theorem and ignoring the constant scaling factor $\frac{1}{2\pi}$, this energy may be described as

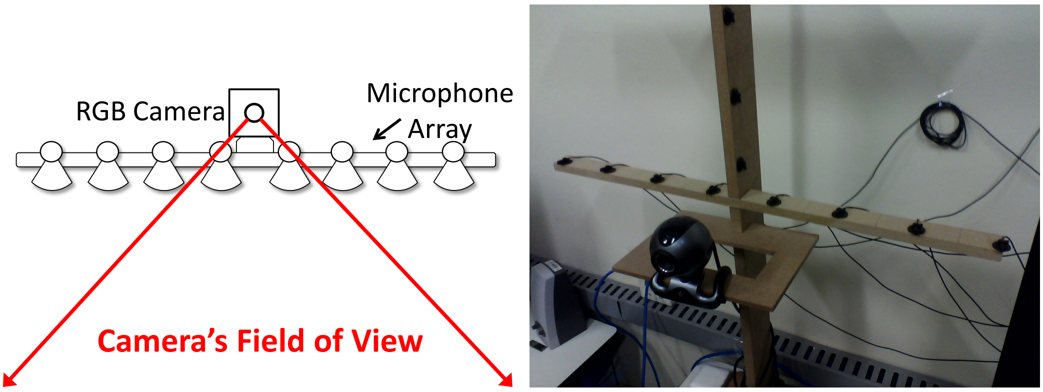
$$P(\mathbf{q}) = \int_0^{2\pi} |Y(\omega, \mathbf{q})|^2 \, d\omega = \int_0^{2\pi} \left| \sum_{m=1}^{N_{\text{mic}}} \frac{X_m(\omega)}{|X_m(\omega)|} e^{-j\omega\tau_m^{\mathbf{q}}} \right|^2 d\omega. \tag{5.4}$$

Once $P(\mathbf{q})$ has been computed for all candidate positions using Eq. (5.4), we can estimate the sound source location as

$$\hat{\mathbf{q}} = \underset{\mathbf{q} \in \mathcal{Q}}{\text{argmax}} \, P(\mathbf{q}), \tag{5.5}$$

where $\mathcal{Q}$ denotes a set of points in space that represent all candidate locations. This maximization approach robustly finds the dominant sound source given a relatively short time window (e.g. 50ms). While the SRP-PHAT may be implemented different ways (DIBIASE, 2000), this description in Eq. (5.4) has been shown to be suitable for GPU implementation (MINOTTO et al., 2012).

In general, one drawback of using the SRP-PHAT's global maxima to localize potential speakers is that the precision of such approaches tend to drop as the number of simultaneous speech sources increases. This is a rather common problem with beamforming techniques, since one speaker's voice acts as noise to the others' (BENESTY; CHEN; HUANG, 2008). Moreover, to assume that a set of largest $P(\mathbf{q})$ values represents the speakers' positions is somewhat inaccurate, given that the SRP-PHAT's power map contains many local maxima due to noise and reverberation. Therefore, for evaluating which $P(\mathbf{q})$ values truly characterizes a speaker, proper VAD technique must be



(a) (b)

Figure 5.1: (a) Schematic representation of the proposed approach. (b) Our prototype system.

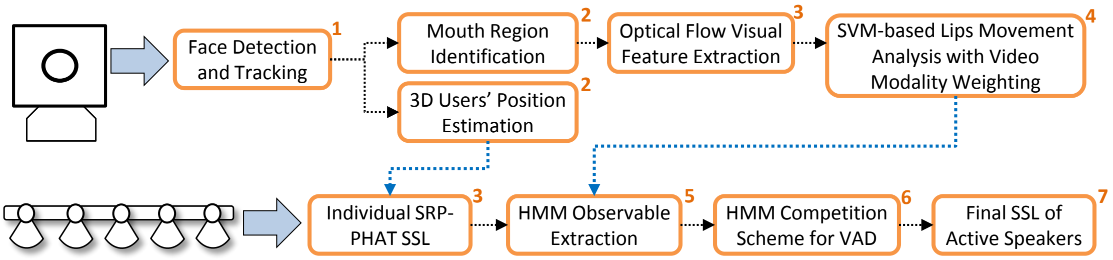

Face Detection and Tracking
1
Mouth Region Identification
2
Optical Flow Visual Feature Extraction
3
SVM-based Lips Movement Analysis with Video Modality Weighting
4
3D Users' Position Estimation
2
Individual SRP-PHAT SSL
3
HMM Observable Extraction
5
HMM Competition Scheme for VAD
6
Final SSL of Active Speakers
7

Figure 5.2: Schematic representation of our algorithm's flow. The indexes at the upper-right corner of the boxes represent the order at which the individual steps are processed. The blue arrows represent the moments where information between audio and video are exchanged where the multimodal mid fusion happens.

employed using some kind of *a priori* knowledge (or assumptions) about the acoustic scenario, e.g., known number of speakers (BRUTTI; OMOLOGO; SVAIZER, 2008), noise is diffuse (TAGHIZADEH et al., 2011), or speakers' $P(\mathbf{q})$ are above a minimum noise level (DO; SILVERMAN, 2010).

5.3 The Proposed Approach

Our work approaches multiple speaker VAD and SSL as joint problems. We employ a linear microphone array and a conventional RGB camera. Our setup expects the users to be facing the capture sensors and to be within the Field of View (FOV) of the camera, as it is the case of most HCI systems (JAIMES; SEBE, 2007). A schematic representation of the required setup is provided in Figure 5.1(a), and our prototype room based on such setup is given in Figure 5.1(b). It is important to mention that for the entire work, we adopt a Cartesian coordinate system, where the width dimension (x) is parallel to the array, and positive to the right; the height dimension (y) is positive upwards; and the depth dimension (z) is positive towards the camera's FOV. This convention is also applied for image coordinates' x and y dimensions. Figure 5.2 summarizes the pipeline for the proposed VAD and SSL approach, which is described next.

5.3.1 Visual Analysis

The processing related to the the video modality may be summarized in two steps: extracting proper visual features from different potential speakers, and then evaluate them using some sort classification technique. Next subsections detail these two steps.

5.3.1.1 Visual Feature Extraction

In order to extract a reliable visual feature for VVAD, we exploit the fact that anyone who has intention to speak moves the lips. As previously mentioned in Section 5.2.1 this has been approached in other works (TAKEUCHI S.; HAYAMIZU, 2009; ATREY et al., 2010; AUBREY; HICKS; CHAMBERS, 2010; TIAWONGSOMBAT et al., 2012): computing the optical flow of a region enveloping the speakers' mouths. In this work, we chose the Lucas-Kanade (LK) (LUCAS; KANADE, 1981) algorithm for such task, due to its good compromise between computational cost and accuracy.

The first step in this process is to use a face tracker/detector algorithm to identify the potential speakers in the captured image. We opted to use the face tracker in (BINS et al., 2009) given its low computation complexity and robustness to light changes and head rotation. After running the tracker for the current frame t, K faces in the scene are detected/tracked, and we may then find the bounding rectangle of each speaker's mouth using the following anthropometric relations (FARKAS, 1994):

$$\mathbf{x}_{\text{lips}}^{\text{tl}} = \mathbf{x}_{\text{mid}} + (-0.4r, 0.25r) \tag{5.6}$$

$$\mathbf{x}_{\text{lips}}^{\text{br}} = \mathbf{x}_{\text{mid}} + (0.3r, 0.65r) \tag{5.7}$$

where $\mathbf{x}_{\text{lips}}^{\text{tl}}$ and $\mathbf{x}_{\text{lips}}^{\text{br}}$ respectively represent the coordinates of the top-left and bottom-right corners of the lips' bounding rectangle; $\mathbf{x}_{\text{mid}}$ is the 2D location of the face's center, and r is the radius of the face (distance from $\mathbf{x}_{\text{mid}}$ to any corner of the face's bounding box). All these values are expressed in image coordinates.

After the mouth region has been defined for each speaker, we then populate that region with punctual LK features that will be tracked between adjacent image frames using the algorithm described in (BOUGUET, 2000). More precisely, for every new image frame (at time t) we distribute N_{LK} features inside each of the detected mouth regions of frame $t-1$, in a regular grid manner, and track them to their corresponding new position at t. An illustration of this process is shown in Figure 5.3. Additionally, it is important to notice that for defining the points to track, a feature selection algorithm such as in (SHI; TOMASI, 1994) could be used instead. However, for our case, they provide no extra tracking accuracy while increasing the overall computational cost of the proposed approach.

For the feature extraction process, we leverage from the fact that, oppositely to silence situations, the computed optical flow vectors tend to show large magnitudes during speech. However, we also observe that since we analyze a region larger than the actual lip area (which is necessary for not losing track of the mouths during head translations and mainly rotations), not all optical flow vectors have large magnitudes during speech. For this reason, we not only extract a measure of energy but also the standard deviation of the magnitudes as our visual features.

Denoting $\mathbf{x}_i(t)$ as the position of the i^{th} LK feature, with relation to its face center, and frame at time t (that has been tracked from $t-1$), we may define the magnitude of its resulting optical flow vector as $V_i(t) = ||\mathbf{x}_i(t) - \mathbf{x}_i(t-1)||$, where $||\cdot||$ denotes the Euclidean norm. Therefore, the extracted visual features from the optical flow process of

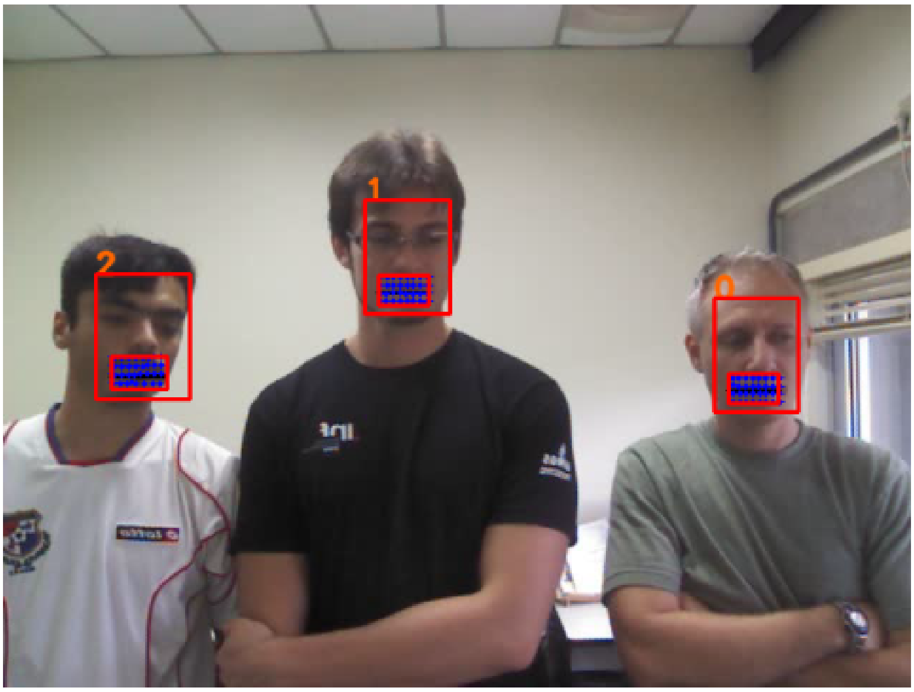


Figure 5.3: Example of LK features distributed as a regular grid inside the mouths' bounding rectangle.

a given speaker are defined as

$$\mu_V(t) = \frac{1}{N_{\text{LK}}} \sum_{i=1}^{N_{\text{LK}}} \frac{V_i(t)}{r}, \tag{5.8}$$

$$\sigma_V(t) = \sqrt{\frac{1}{N_{\text{LK}}-1} \sum_{i=1}^{N_{\text{LK}}} \left(\frac{V_i(t)}{r} - \mu_V(t)\right)^2}, \tag{5.9}$$

where the division by r is used to normalize the features with respect to the image dimensions and the distance of the users from the camera. We may also note the users' lateral velocity is automatically compensated by computing $\mathbf{x}_i(t)$ with respect to the face center. Therefore, $\mu_V(t)$ and $\sigma_V(t)$ respectively represent the mean and standard deviation of $V_i(t)$, and are expected to be higher during speech than during silence. It is important to notice that, although we describe these measures without a k index (for simplicity), they are computed K times, once for each speaker.

Using these features for describing lip movements has two main advantages. They represent well the movements of the lips even if the mouth region is not precisely estimated, which is the case of the used anthropometry-based approach, and they do not require prior knowledge about the shape of the mouth, such as the contour of the lips. However, despite such advantages, these features are not robust against small pauses during speech, since they are computed between consecutive frames only, and in HCI applications (such as ASR) it is often desired that speech hiatuses are detected as part of the spoken sentences instead of as silence moments (RABINER; SCHAFER, 1978). Therefore, we analyze $\mu_V(t)$ and $\sigma_V(t)$ over a longer time window of T frames. We propose four new features, extracted from Eqs. (5.8) and (5.9):

$$\begin{aligned} f_1(t) &= \frac{1}{T}\sum_{i=0}^{T-1}\mu_V(t-i), && (5.10)\\ f_2(t) &= \sqrt{\frac{1}{T-1}\sum_{i=0}^{T-1}(\mu_V(t-i)-f_1(t))^2}, && (5.11)\\ f_3(t) &= \frac{1}{T}\sum_{i=0}^{T-1}\sigma_V(t-i), && (5.12)\\ f_4(t) &= \sqrt{\frac{1}{T-1}\sum_{i=0}^{T-1}(\sigma_V(t-i)-f_3(t))^2}. && (5.13)\end{aligned}$$

This imposes temporal coherence to the visual features to a point that speech hiatuses do not become problems for a further classifier. By contrast, this approach may also introduce a detection lag between speech-to-silence and silence-to-speech transitions.

5.3.1.2 Video-related Probability Estimation using SVM

The next step used for extracting a probability measure from our final visual features is to use some sort of supervised classifier. In this work, we chose the SVM algorithm implemented in (CHANG; LIN, 2011) for it provides the known robustness of SVM techniques (SCHAPIRE; FREUND, 1998) and allows probability estimation (instead of binary labeling) using the approach described in (WU; LIN; WENG, 2004).

For training the SVM model, we perform a grid-search at the unknown parameters running successive turns of 5-fold cross-validation, since employing this method is known for avoiding overfitting problems (CHANG; LIN, 2011). The training data used during this procedure are extracted from our labeled multimodal database (described in Section 5.4). Finally, once the best set of parameters are found, the SVM model Φ is trained (through the algorithm in (FAN; CHEN; LIN, 2005)), and a posterior speech probability v for the video modality is extracted as

$$v = P(\text{speech}|\mathbf{f}_{\text{vid}};\Phi), \qquad (5.14)$$

where $\mathbf{f}_{\text{vid}} = [f_1(t), f_2(t), f_3(t), f_4(t)]$ is a vector composed of the previously described visual features, and intuitively $P(\text{silence}|\mathbf{f}_{\text{vid}};\Phi) = 1 - v$.

At this point, it is important to notice that v could be directly used for the final decision of a video-only VAD approach. However, its accuracy is highly dependent on the distance of the speakers from the camera: as the user move far from the camera, the mouth region appears smaller in image coordinates, and the optical flow tends to become noisier. Also, participants moving at high lateral velocities may also corrupt the extracted visual features. Despite the implicit compensation for lateral movements when computing $\mathbf{x}_i(t)$, abrupt translations may blur the faces, also corrupting the optical flow estimate. For this reason, we propose a weighting factor w_v for Eq. (5.14), that is monotonically decreasing with respect to both distance of the user from the camera and his/her lateral speed:

$$w_v = \exp\{-z'_{\text{vid}} - v'_x\}, \qquad (5.15)$$

where $z'_{\text{vid}} = z_{\text{vid}}/z^{\max}_{\text{vid}}$ and $v'_x = v_x/v^{\max}_x$ are respectively the normalized depth and lateral velocity of the user (measured in world coordinates), and $z^{\min}_{\text{vid}} \le z_{\text{vid}} \le z^{\max}_{\text{vid}}$, and $0 \le$

$v_x \leq v_x^{\max}$. Furthermore the k^{th} participant's z_{vid} is the depth component of the estimated 3D video-based position q_k^{vid} (such estimation process is described in Section 5.3.2).

In other words, the VVAD is expected to have maximum effect for the multimodal fusion when $z_{\text{vid}} = z_{\text{vid}}^{\min}$ and $v_x = 0$, and exponentially lose its effect as the users' depths and velocities reach $z_{\text{vid}}^{\max}$ and $v_x^{\max}$ respectively, having no effect at all when $y_{\text{vid}} > z_{\text{vid}}^{\max}$ and $v_x > v_x^{\max}$. An appropriate value for $z_{\text{vid}}^{\min}$ is chosen to be the minimum distance two users may comfortably participate in a camera-equipped HCI system while not leaving its FOV. As for $z_{\text{vid}}^{\max}$, we chose the value at which the Lucas-Kanade optical flow algorithm is not able to track the movements of the lips. Finally, for finding a reasonable value for $v_x^{\max}$ we extracted the maximum velocity a user has reached in our multimodal recordings, finding $v_x^{\max} = 0.25m/s$. In our setup we used a Logitech Quickcam Pro 5000 camera, and experimentally found $z_{\text{vid}}^{\min} = 0.5m$ and $z_{\text{vid}}^{\max} = 1.8m$ for VGA (640×480) video sequences.

5.3.2 Audio Analysis

Computing the SRP-PHAT for identifying competing sound sources is known to be a hard task. As mentioned in Section 5.2.1, many works have approached this using clustering techniques (DO; SILVERMAN, 2008, 2010; CAI; ZHAO; WU, 2010) or some iterative isolation criteria (BRUTTI; OMOLOGO; SVAIZER, 2008; TAGHIZADEH et al., 2011). These approaches, however, may be rather complex and also fail under high noise conditions. We therefore propose a simple and effective process for isolating different regions around potential speakers in the SRP-PHAT's global search space $\mathcal{Q}$, which shows to be robust for the simultaneous speakers scenario, even under noisy and reverberant conditions.

For the k^{th} participant, we define an 1D ROI $\mathcal{Q}_k$ as a subset of the global search region $\mathcal{Q}$. Each ROI is treated as individual space regions having their own, bounded, coordinates system, that is centered around each participant's 3D video-based location $\mathbf{q}_k^{\text{vid}}$, as illustrated in Figure 5.4. This way, given a fixed length ℓ for $\mathcal{Q}_k$ (in meters), all ROIs may form equally sized horizontal line segments (parallel to the microphone array) centered at each tracked face. Finally, Eqs. (5.4) and (5.5) can be calculated for the k^{th} speaker using $\mathcal{Q}_k$ instead of $\mathcal{Q}$, which allows us to later separately analyze the SRP-PHAT's behavior of each user through our HMM approach.

It is important to notice that an 1D ROI is chosen (instead of 2D or 3D) as a consequence of our linear array configuration, since microphone arrays best discriminate locations parallel to the same direction most microphones are distributed along (JOHNSON; DUDGEON, 1993b). Therefore, in the case of our linear array (previously depicted in Figure 5.1), the SRP-PHAT is more accurate along the horizontal dimension, making an 1D ROI enough for our VAD approach, at low computational cost in the search process of Eq. (5.5). As for the choice of a linear configuration, we base on the fact that in a multimodal multi-user HCI applications, the users tend to stand side-by-side in order to remain within the camera's FOV, emphasizing the need for better localization along x, the width dimension.

Before running the SRP-PHAT, $\mathbf{q}_k^{\text{vid}}$ must be computed so the ROIs may be properly centered around each person's 3D position. This is done by estimating $\mathbf{q}_k^{\text{vid}}$ from the 2D face-tracking results (the centering process must be repeated every frame), using an inverse projective mapping. Assuming a pinhole camera model and that the camera is aligned with the microphone array, the relation between image coordinates

$\mathbf{x}_{\text{mid}} = (x_{\text{pix}}, y_{\text{pix}})$ and world coordinates $\mathbf{q}^{\text{vid}} = (x_{\text{vid}}, y_{\text{vid}}, z_{\text{vid}})$ is given by

$$x_{\text{pix}} = f_{\text{len}} \frac{x_{\text{vid}}}{z_{\text{vid}}}, \quad y_{\text{pix}} = f_{\text{len}} \frac{y_{\text{vid}}}{z_{\text{vid}}}, \tag{5.16}$$

where f_{len} is the focal length of the camera.

Therefore, given the mean radius $r_{1\text{m}}$ (in pixels) of a face placed at one meter from the camera (which can be estimated experimentally or based on the projection of the anthropometric average face radius (FARKAS, 1994)), the z component (depth) of a detected face can be estimated through

$$z_{\text{vid}} = r/r_{1\text{m}}, \tag{5.17}$$

where r is the radius (in pixels) of the tracked face. This way, given z_{vid} and the image-related face central position $\mathbf{x}_{\text{mid}}$, it is possible to obtain the width and height world components of the k^{th} speaker by isolating x_{vid} and y_{vid} in Eq. (5.16). This allows the horizontal search region $\mathcal{Q}_k$ to be centered at the k^{th} speaker's video-based world position $\mathbf{q}_k^{\text{vid}}$, so that his/her audio-based location may be computed using the SRP-PHAT as

$$\mathbf{q}_k^{\text{aud}} = \underset{\mathbf{q} \in \mathcal{Q}_k}{\text{argmax}}\, P(\mathbf{q}). \tag{5.18}$$

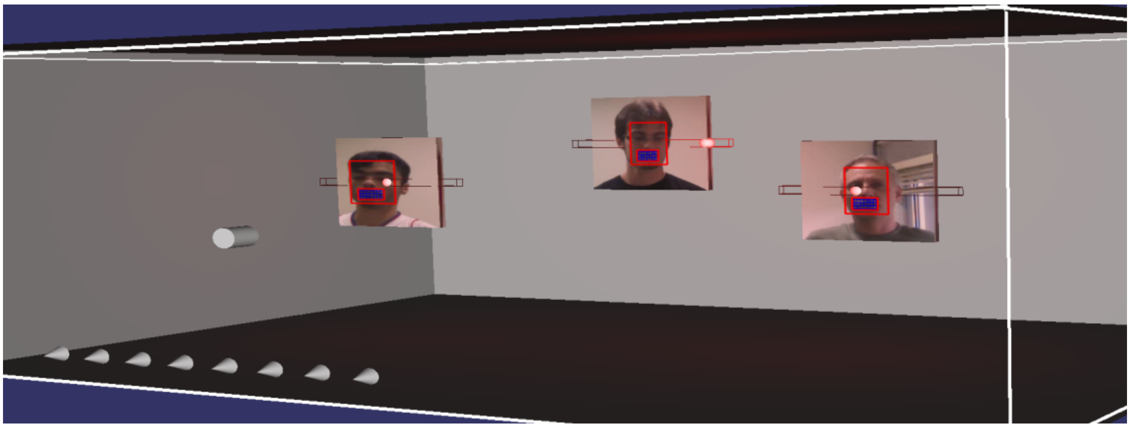


Figure 5.4: Example of ROI-based SRP-PHAT search being performed for each user in the scene. The 3D model is rendered from the scene's information: the cylinder represents the camera; the cones represent the microphones; the gray planes form the global search region $\mathcal{Q}$; the long parallelepipeds are the 1D ROIs $\mathcal{Q}_k$; and the red spheres are the locations estimated through Eq. (5.18).

At this point, it is important to notice that neither $\mathbf{q}_k^{\text{aud}}$ nor $\mathbf{q}_k^{\text{vid}}$ are the final location that is computed by our SSL approach. These estimate are used by the HMMs for spatio-temporal coherence analysis to perform both the final MVAD and SSL. These topics are covered in the next subsections.

5.3.3 Multimodal Mid-Fusion using HMMs

Given the results of the video and audio analyses of each speaker, υ and $P(\mathbf{q}_k^{\text{aud}})$, respectively, we develop a fusion scheme by using an HMM competition scheme, which is inspired in (BLAUTH et al., 2012; MINOTTO et al., 2013). We extend such works to the multiple speaker scenario, also weighing the importance of υ by w_υ.

In summary, two HMMs that model the expected behavior of the SRP-PHAT peak for the multiple speaker scenario are defined. One HMM describes speech situations, and the other, silence situations. By extracting proper observations from the SRP-PHAT, it is possible to use a competition scheme between both models in order to evaluate the SRP-PHAT's spatio-temporal behavior for different speakers; separate scores for the same set of observations may be computed for each HMM through approaches such as the Viterbi algorithm (RABINER, 1989), and then compared to form a final MVAD decision. Therefore, Section 5.3.3.1 explains the general idea of our competition approach; in 5.3.3.2 and 5.3.3.3 the speech and silence HMMs are described respectively; Section 5.3.3.4 presents our MVAD approach, and 5.3.3.5 the SSL one; finally, in 5.3.3.6 we explain how the parameter estimation of the HMMs is performed.

5.3.3.1 The Proposed HMMs

An HMM can be used to model dynamic systems that may change their states in time. An HMM with discrete observables is characterized by $\lambda = (A, B, \pi)$, where $A = [a_{ij}]$ for $1 \leq i, j \leq N$ is the transition matrix that contains the probabilities of state changes, $B = [b_n(\mathbf{O})]$ for $1 \leq n \leq N$ describes the observation probability for each state, and $\pi = [\pi_i]$ for $1 \leq i \leq N$ contains the initial probabilities of each state. Clearly, the choice of the parameters is crucial to characterize a given HMM.

In (BLAUTH et al., 2012), competing HMMs were used for single-speaker VAD by exploring the expected spatio-temporal location of the sound source when the speaker is active. In this work we adopt a similar approach, but instead we build $2K$ competing HMMs (two for each detected face), also including the video-based VAD cue v. More precisely, each candidate sound source location in $\mathcal{Q}_k$ is a state of the HMMs, so that the number N of states depends on the size of the search region. We denote $\mathbf{S}_k = \left\{S_1^k, S_2^k, ..., S_N^k\right\}$ such N states for k^{th}, with N given by

$$N = \left\lfloor \frac{\ell}{spa} \right\rfloor + 1, \tag{5.19}$$

where spa is the real-world spacing between neighboring points in $\mathcal{Q}$, and should be chosen (along with ℓ) in a way that N is odd, allowing $\mathbf{S}_k$ to have a middle state.

In our approach, we determine an observable that is to able carry information about the estimated speaker's position as well as some sort of confidence measure of that estimate. That is, recalling the HMMs' states are the candidate positions of the SRP-PHAT, the observation extracted for user k is a two-dimensional vector $\mathbf{O}_k = (O_1^k, O_2^k)$ computed based on $P(\mathbf{q})$. It is given by

$$O_1^k = \mathbf{q}_k^{\text{aud}}, \tag{5.20}$$

$$O_2^k = \frac{P(\mathbf{q}_k^{\text{aud}})}{\min\limits_{\mathbf{q} \in \mathcal{Q}_k} P(\mathbf{q})}. \tag{5.21}$$

The rationale of this approach is that, in speech situations, O_1^k should provide the correct location of the speaker, and O_2^k tends to be a large value (since the maximum is expected to be considerably larger than the minimum). On the other hand, in silence situations, all values of $P(\mathbf{q})$ tend to be similar, and $\mathbf{q}_k^{\text{aud}}$ should represent the location of the inexistent sound source. Additonally, in this later case, O_2^k will be smaller, since the maximum and mininum tend to be simillar. These informations are used when bulding the competing models.

In theory, the lower bound for O_2^k is 1, and the upper bound $O_2^{\max}$ is ∞. We have observed in different experiments (with different speakers and varying background noise) that O_2 gets really close to 1 during non-speech, and reaches a maximum value during speech. Therefore, we experimentally find the upper bound $O_2^{\max}$, and the values of O_2 are quantized into L possible values within the range $[1\ O_2^{\max}]$ to obtain an HMM with discrete range of observables. Values of O_2^k larger than $O_2^{\max}$ are quantized into $O_2^{\max}$, and we choose $L = 8$ (higher values show no extra representativeness for O_2^k). Variable O_1^k represents the position at which the SRP-PHAT peak is located in $\mathcal{Q}_k$, and is therefore discretized into N values.

The next step for defining the speech and silence HMMs is then to define the probabilities of A, B and π in a way that, during true speech situations, the described observables and states behave as modeled by the speech HMM, and during silence situations, as modeled by the silence HMM. Next section details this matter.

5.3.3.2 The Speech HMM

For determining the parameters $\lambda = (A, B, \pi)$ of an HMM, a widely used estimation approach is the Baum-Welch algorithm (RABINER, 1989). For our HMM, however, such approach is impracticable. The used models present a relatively high number of states (N) and observables ($M = NL$), which would require a large amount of training samples (comprising several situations such as speakers in different positions, alternation of speech and silence, presence/absence of background noise etc.). Instead, we propose parametric probability density functions (PDF) for the HMM matrices based on the expected behavior of users in an audiovisual HCI situation. We also highlight that the normalization process of the hereafter described PDFs are omitted for better readability, although it is important to notice that π and the rows of matrices A and B must sum up to unity.

Recalling the observation $\mathbf{O}_k$ is a two-element vector, we may define the distribution of the observation in the n^{th} state S_n^k using the following joint PDF (here we omit the k affix for readability and for the fact the speech HMM λ^{sp} is the same to any user):

$$b_n^{\text{sp}}(\mathbf{O}) = b_n^{\text{sp}}(O_1, O_2) = b_n^{\text{sp}}(O_1|O_2)b^{\text{sp}}(O_2), \tag{5.22}$$

where the superscript $^{\text{sp}}$ stands for "speech", $b^{\text{sp}}(O_2)$ is the distribution of O_2 during speech situations (which does not depend on the state S_n), and $b_n^{\text{sp}}(O_1|O_2)$ is the conditional probability of O_1 given O_2, which is strongly affected by n.

Since sharp peaks tend to occur in the SRP-PHAT during speech situations, O_2 is expected to be large, and this fact should be availed by the speech HMM. Therefore, $b^{\text{sp}}(O_2)$ should be a monotonically increasing function, and the following exponential function was chosen:

$$b^{\text{sp}}(O_2) = \exp\left\{c_1 \frac{O_2}{O_2^{\max}}\right\}, \tag{5.23}$$

where c_1 is an estimated auxiliary constant (see Section 5.3.3.6) that controls the decay of $b^{\text{sp}}(O_2)$.

Function $b_n^{\text{sp}}(O_1|O_2)$ describes the conditional density of O_1 given O_2. Since each state n relates to a position in the search space $\mathcal{O}$, the value of O_1 (which is the position of the largest SRP-PHAT value) should be close to n. Furthermore, if the confidence O_2 is large, the probabilities should decay abruptly around this peak; if O_2 is small, though, the decay around the peak should be smoother, allowing other O_1 to be encountered with higher probabilities as well (even if the SRP-PHAT matches the actual position of the user with a low O_2, it might be a coincidence). Inspired in (BLAUTH et al., 2012), we used an exponential function to model this behavior:

$$b_n^{\text{sp}}(O_1|O_2) = \exp\left\{-g(O_2)|O_1 - n|\right\}, \tag{5.24}$$

where $g(O_2)$ controls the speed at which $b_n^{\text{sp}}(O_1|O_2)$ decays due to changes in O_2. This means that, as the confidence O_2 gets larger, the decay around n should be faster, reducing the chances of neighboring O_1 to happen. Therefore, $g(O_2)$ is chosen as

$$g(O_2) = \exp\left\{-c_2 O_2 - c_3\right\}, \tag{5.25}$$

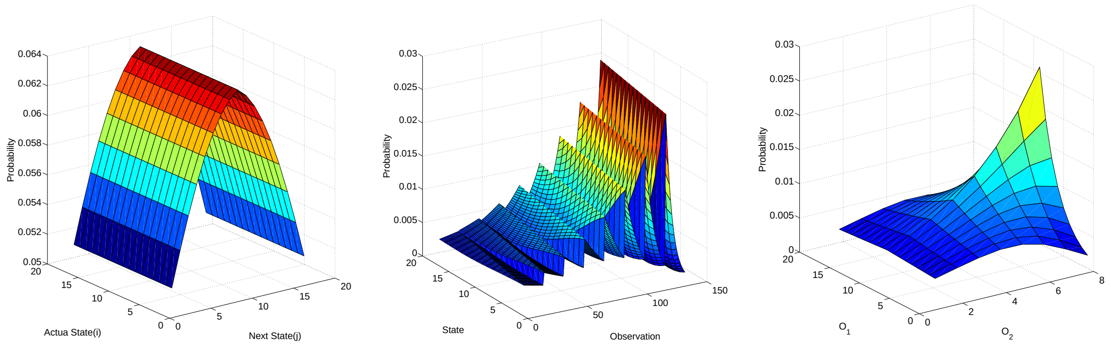
where c_2 and c_3 are also auxiliary constants that are estimated using the approach described in Section 5.3.3.6.

In order to find an adequate configuration for the state transition matrix $A_{\text{sp}} = a_{ij}^{\text{sp}}$, it is first important to observe the following. Given that the ROIs are always centered at each speaker's position, we must expect O_1 to move toward the central state during speech situations, even if the SRP-PHAT peaks at neighboring states with high confidence. Therefore, A_{sp} is configured in such a way that $a_{\frac{N}{2}j}^{sp}$ is maximum (for $1 \leq j \leq N$), and the probabilities decay as j distances from $N/2$. This way, A_{sp} is defined as

$$A_{\text{sp}} = a_{ij}^{\text{sp}} = \exp\left\{\frac{|N/2 - j| + 1}{2\sigma^2}\right\}, \tag{5.26}$$

where σ is constant used for controlling the decay. As we may notice, a_{ij}^{sp} depends only upon j, which is the state being transited to. In other words, regardless of which state the speaker is located at, the one with the highest transition probability is the middle one.

For the sake of illustration, the transition matrix A_{sp}, the observation matrix B_{sp} and the probability density function $p_8^{\text{sp}}(O_1, O_2)$ related to state S_8^k ($N = 17$) are depicted in Figure 5.5.



(a) (b) (c)

Figure 5.5: Speech HMM matrices plots for $N = 17$ and $L = 8$. (a) Transition matrix A_{sp}, (b) Observation matrix B_{sp} and (c) Slice of observation matrix, for $n = 8$.

5.3.3.3 *The Silence HMM*

The silence-related HMM is characterized by $\lambda^{\text{si}} = (A^{\text{si}}, B^{\text{si}}, \pi^{\text{si}})$. As it was already pointed out, during silence periods the response $P(\mathbf{q})$ of the SRP-PHAT at each position (state) should be similar, so that observable O_2^k is expected to be close to the smallest possible value, which is 1. Furthermore, $\mathbf{q}_k^{\text{aud}}$ will correspond to random positions inside $\mathcal{Q}_k$, owing to background noise and reverberations. Therefore, similarly to Eq. (5.22), the joint probability function of the observables, for state S_n^k, can be written as[1]

$$b_n^{\text{si}}(\mathbf{O}) = b_n^{\text{si}}(O_1, O_2) = b_n^{\text{si}}(O_1|O_2)b^{\text{si}}(O_2), \tag{5.27}$$

where function $b^{\text{si}}(O_2)$ was obtained similarly to its counterpart in speech situations, except that higher probabilities should occur for smaller values of O_2:

$$b^{\text{si}}(O_2) = \exp\left\{c_1 \frac{O_2^{\max} - O_2 + 1}{O_2^{\max}}\right\}, \tag{5.28}$$

where c_1 has the same value and role as in Eq. (5.23).

For the conditional probability $p_n^{\text{si}}(O_1|O_2)$, there are two important things to be noted. First, such distribution should not depend on the state S_n^k, since the position of the peak is related to noise, and not to an actual sound source at the discrete position n. Secondly, all observables O_1 should be equally probable, for the same reason. Hence, an uniform conditional probability function is chosen:

$$b_k^{\text{si}}(O_1|O_2) = \frac{1}{N}. \tag{5.29}$$

If in speech situations the peak of the SRP-PHAT is expected to be close in temporally adjacent observations, the same is not true for silence periods. Since all responses are usually similar, background noise plays a decisive role when retrieving the highest peak, which may be far from the one detected in the previous observation. In fact, the proposed state transition matrix for the silence-related HMM considers all transitions equally probable:

$$A_{\text{si}} = a_{ij}^{\text{si}} = \frac{1}{N}. \tag{5.30}$$

As for the initial distribution π for both speech and silence HMMs, we assumed that all states (i.e., positions) are initially equally probable.

5.3.3.4 Multimodal VAD using the HMMs

Given the speech HMM λ^{sp}, the silence HMM λ^{si}, and a sequence of observables $\mathcal{O}_t^k = \{\mathbf{O}_k(t-T), \mathbf{O}_k(t-T+1), ..., \mathbf{O}_k(t)\}$ for speaker k within a time window of size T, we can compute how well both HMM describe $\mathcal{O}_t^k$ (this is the same time window used in Section 5.3.1.1). If $\mathcal{O}_t^k$ was generated during a speech situation, then it should present a higher adherence to λ^{sp} than λ^{si}, and the opposite for silence situations. More precisely, this can be done by computing likelihoods $P(\mathcal{O}_t^k; \lambda^{\text{sp}})$ and $P(\mathcal{O}_t^k; \lambda^{\text{si}})$ using the forward-backward procedure (RABINER, 1989). This way, an AVAD-only decision could be performed such that the k^{th} user at frame t is considered to be active if $P(\mathcal{O}_t^k; \lambda^{\text{sp}}) > P(\mathcal{O}_t^k; \lambda^{\text{si}})$.

However, as the number of simultaneous speakers increase, the height of the SRP-PHAT peaks at the actual speaker positions is lowered, since one person's voice acts as noise to the others'. As a consequence, O_2^k might not always be as large as expected, so that false negatives may occur when there is speech. For this reason, we propose a mid-fusion technique that attempts to boost the values of O_2 based on visual cues, particularly in simultaneous speech situations, by using the confidence v of the VVAD

[1] We again omit the superscript ^{k} for the sake of readability. The PDFs are equal to all users.

algorithm. More precisely, we propose an enhanced observable $\bar{\mathbf{O}}_k = (O_1^k, \bar{O}_2^k)$ for the HMM competition scheme, with

$$\bar{O}_2^k = O_2^k \left(1 + \frac{\upsilon_k w_{\upsilon_k}}{c_4}\right), \tag{5.31}$$

where υ_k and w_{υ_k} are computed for the k^{th} user through Eqs. (5.14) and (5.15), respectively, and $c_4 > 1$ controls the contribution of the video modality to the multimodal fusion. The value of c_4 must be carefully chosen so that O_2^k is effectively enhanced during simultaneous speech situations, but not overly amplified to avoid false VAD. Our procedure for setting c_4 is presented in Section 5.3.3.6.

Finally, according to our final MVAD approach, a given user is considered to be active if $P(\bar{\mathcal{O}}_t^k; \lambda^{\text{sp}}) > P(\bar{\mathcal{O}}_t^k; \lambda^{\text{si}})$. As for the time window T, it must be properly chosen. If it receives a small value, speech hiatus between consecutive words may be detected as silence, which is usually not desirable for speech recognition. On the other hand, larger values for T provide better temporal consistency, but also lead to delays when detecting speech-silence or silence-speech changes. In this work we chose T in a way it corresponds to a window approximately 1 second long, since it showed to be efficient to deal with speech hiatus and not present a long delay when the location of the speaker changes

5.3.3.5 Multimodal SSL

As previously mentioned in Section 5.2, in order to perform multiple speaker VAD, one implicitly needs to perform localization (either in 3D or 2D in image coordinates), so that active speakers may be differentiated from inactive ones. For this reason, despite the main difficulty of a competing sources scenario being the VAD part itself, we also implement SSL as a part of our algorithm. While we do not consider this to be the main contribution of our work, we show that we may easily avail from our spatio-temporal-based HMM formulation to locate the active speakers.

As a requirement for our MVAD approach, two location estimates are initially produced for each speaker, from the audio and video modalities, $\mathbf{q}_k^{\text{aud}}$ and $\mathbf{q}_k^{\text{vid}}$, respectively. Either one of them could be used as a final SSL decision for the speakers. However, both estimates present inaccuracies due to practical issues. The audio location $\mathbf{q}_k^{\text{aud}}$ is highly corrupted by noise and reverberation, especially during simultaneous speech situations. The video location $\mathbf{q}_k^{\text{vid}}$ is affected by the depth estimation in Eq. (5.17), since the faces radii r present small variations across time (and for different users). Therefore, we propose a more robust approach by reusing the speech HHM.

Recalling that our HMMs are based on the spatial locations $\mathcal{Q}_k$ of the SRP-PHAT, it is possible to use a decoding algorithm that finds the state sequence with length T that best corresponds (according to some optimality criterion) to the sequence of observables $\mathcal{O}_t^k$ evaluated using a given model λ. In our case, each state of the T decoded states would correspond to the speaker location at each time frame, and such decoding process could be performed using the Viterbi algorithm (RABINER, 1989). Therefore, by decoding the active speakers' $\mathcal{O}_t^k$ (same observables used for VAD) against the speech model λ^{sp}, the last state in the computed sequence represent the most recent location of the k^{th} speaker. This approach introduces both spatial and temporal coherence to the SRP-PHAT's location estimates, due to the time-window analysis of the Viterbi algorithm and to the characteristics of A_{sp} and B_{sp}.

However, we must recall our HMM approach is applied only to the width dimension of the search region, meaning only the horizontal position of each active speaker is brought from our MVAD approach. For this reason, for SSL purposes only, we separately decode λ^{sp} using observables obtained from other two 1D ROIs, one spanning along the depth dimension (z), and the other one along the height (y). They are also centered at $\mathbf{q}_k^{\text{vid}}$, such that the three ROIs are orthogonal to each other, allowing each 1D search region to retrieve one component of the speakers' 3D location. Therefore, denoting x_{HMM}^k, y_{HMM}^k and z_{HMM}^k as the locations found by decoding the above 1D HMMs, we define the final 3D position of the k^{th} speaker as

$$\hat{\mathbf{q}}_k = \left(x_{\text{HMM}}^k, y_{\text{HMM}}^k, z_{\text{HMM}}^k\right). \tag{5.32}$$

Finally, it is important to notice this SSL approach still keeps our algorithm at a low computational cost, since only $3N$ states are evaluated with the Viterbi method, oppositely to N^3 as would happen if a 3D cuboid-like ROI was used.

5.3.3.6 Parameter Estimation

As previously mentioned, due to the fact that matrices A_{sp} and B_{sp} of the speech HMM are composed by a large set of states and observables, we opted to use parametric models. However, the chosen PDFs present crucial parameters to which values must be assigned for the HMMs to work properly. These are the case of c_1 and $O_2^{\max}$ in Eq. (5.23), c_2 and c_3 in Eq. (5.25), σ in Eq. (5.26), and c_4 in Eq. (5.31).

Based on manually labeled data from our multimodal sequences, we are able to extract the true observable occurrence count and transition count for each state of the speech HMM, thus allowing us to compute the histograms A'_{sp} and B'_{sp}, corresponding to the transition and observation matrices, respectively. Therefore, by using A'_{sp}, B'_{sp}, A_{sp} and B_{sp}, we may define residual functions that upon minimization allow the mentioned constants to be estimated.

However, there are two main practical difficulties in such minimization problem. First, the equations that describe A_{sp} and B_{sp} are not linear, and no direct solution exist. Secondly, the computed histograms may present outliers due to errors in the manual labeling process, compromising the estimation process through overfitting, specially in the case where the training dataset has limited size. To ensure the first problem is avoided, a trust region minimization approach (BYRD; SCHNABEL; SHULTZ, 1988) is applied, which is a robust technique for solving non-linear ill-conditioned minimization problems (CONN; GOULD; TOINT, 2000). For the second issue, we assign an M-estimator as our residue function, which is a robust statistics method for reducing the effect of outliers during parameter estimation problems (SMALL; WANG, 2003). Among the many existing possible M-estimators, we have chosen the Huber function (HUBER, 1964), which has been a popular choice since then (HUBER, 2005).

Finally, the last parameter to be estimated is c_4 for the mid fusion approach in Eq. (5.31). For this, we randomly select some multimodal recordings in our database, and perform a linear search for possible values for c_4 within the range of $(1, 10]$ (using a step of 0.1), and select c_4 as the value that maximizes the total MVAD accuracy for those recordings.

5.4 Experimental Evaluation

All our experiments were conducted in our prototype room, which is a computer lab with the dimension of 4.5 m × 4 m × 3 m and the reverberation time of 0.6 seconds.

Our data acquisition hardware is composed by an uniform array of eight DPA 4060 omnidirectional microphones, placed 8 cm apart from each other, and a Logitech Quickcam Pro 5000 webcam positioned in the middle, as depicted in Fig. 5.1(b)[2]. The 1D ROIs $\mathcal{Q}_k$ were set to have $N = 17$ discrete locations, spaced 2 cm apart from each other, so that $\boldsymbol{\ell} = 34$ cm. The audio signals were captured at $F_s = 44,100$ Hz, and the frame size of $B = 4096$ samples were used to compute the SRP-PHAT at each frame. Video capture was synchronized with audio, so that one image corresponds to one audio frame, implying in an approximate frame rate of 10 images per second. Consequently the time window T mentioned in Section 5.3.3.4 is chosen as $T = 10$.

To evaluate our VAD and SSL approaches, we have recorded a total of 24 multimodal sequences ranging from 40 to 60 seconds of duration each. Eight of them were randomly picked for the training processes of the SVM and HMM parameters, and 16 used for testing. Of the 16 used for testing, six have one speaker in the scene, and are named `One1` to `One6`. Other six contain two speakers, and are named `Two1` to `Two6`, and four having three speakers, named `Three1` to `Three4`. In all recordings the users randomly chat in Portuguese, alternating between speech and silence moments, and for the `Two` and `Three` ones, they intentionally overlap their voices at times. Furthermore, all recordings have some sort of natural noise, such as people talking in background, air-conditioning functioning, door slams, and fan from other computers.

For measuring the VAD accuracy, we have manually labeled each speaker at each frame as active or inactive, and run three experiments for each sequence. One testing the precision of the video modality alone, by using Eq. (5.14), the audio modality, by using $P(\mathcal{O}_t^k; \lambda^{\text{sp}})$ (without the fusion), and the combined modalities, by using $P(\bar{\mathcal{O}}_t^k; \lambda^{\text{sp}})$. Table 5.1 shows the obtained results, from which we may observe that the MVAD outperforms the unimodal classifiers in all experiments, suggesting our fusion technique indeed promotes improvements over the audio or video alone. It is also important to note that accuracy rates for the multimodal version were over 90% for all video sequences.

Another key point to be observed is that most existing approaches that use video information for VAD work under a close capture range (GURBAN; THIRAN, 2006; AUBREY; HICKS; CHAMBERS, 2010; PETSATODIS; PNEVMATIKAKIS; BOUKIS, 2009; TIAWONGSOMBAT et al., 2012), which makes the scenario unrealistic for multiple user HCI applications. In our recorded sequences, however, speakers stand at distances between 0.9m and 1.4m from the camera, which is enough to accommodate up to three side-by-side participants more realistically. As a consequence, such large distances from the camera, as previously mentioned, may considerably degrade any video based technique. Our video weighting approach, however, is able to balance such effect, increasing the overall accuracy of the final MVAD. Table 5.2 shows the VAD results for our two sequences with the highest capture range, with and without Eq. (5.15), the video weighting. We may observe that proper weigthing of the video modality is required so it does not corrupt the final multimodal algorithm. This also suggests that the exsiting video-only/multimodal techniques (as the ones just mentioned) would likely fail on our dataset.

For assessing our SSL approach, a different labeling process had to be performed, since it is rather complex to manually define the precise 3D position of each speaker in each frame. One could stipulate the locations of each user before the recordings, but

[2] Some sequences use a different color camera. More details on the setup may be found in `http://www.inf.ufrgs.br/~crjung/MVAD-data/mvadsimult.htm`, where the multimodal recordings (with the ground truths) are also made available.

Table 5.1: VAD accuracy for all recorded sequences, using our proposed algorithms.

Sequence	Audio	Video	Multimodal
One1	94.55%	89.72%	97.55%
One2	91.12%	79.91%	96.74%
One3	82.24%	81.62%	92.83%
One4	92.52%	81.78%	96.42%
One5	87.23%	85.83%	93.79%
Two1	93.67%	86.30%	96.72%
Two2	94.03%	91.10%	98.32%
Two3	90.40%	89.93%	96.00%
Two4	92.49%	87.47%	96.60%
Two5	87.47%	81.15%	93.04%
Three1	81.58%	84.07%	93.13%
Three2	82.51%	83.14%	92.28%
Three3	82.44%	89.51%	95.76%
Three4	83.96%	84.01%	92.26%
Three5	88.21%	87.74%	94.48%
Average	**88.29%**	**85.55%**	**95.06%**

Table 5.2: VAD accuracy for two distant capture sequences with and without the video weighting approach.

	MVAD w/o weighting	MVAD w/ weighting
Three6	77.06%	93.23%
Three7	76.45%	91.26%

movements would not be allowed, making the scenario unrealistic. For this reason, some sort of automatic labeling had to be employed. We ended up using an RGB-D camera (namely, Microsoft's Kinect sensor) for finding the actual position of the speakers. While this approach may also present some imprecision, it shows to be accurate enough to the point our algorithms may be compared to. Another detail is that not all our multimodal sequences have SSL ground truth, given the Kinect device was not present in all of the recordings. For this reason, so we only assess the SSL accuracy for a portion of our sequences.

To evaluate the SSL performance of our algorithm, we have computed the Euclidean distance between the found locations and the labeled locations, as an error measure (results are shown in Table 5.3). We repeated this process for the video and audio modalities alone as well as for of our multimodal SSL approach. For the video modality, we have used Eq. (5.16). For the audio modality, Eq. (5.18) was applied, with the difference that a cuboid-like ROI was used instead of $\mathcal{Q}_k$ (such 3D search was used only for performing the localization comparisons). For our multimodal HMM approach we used $\hat{\mathrm{q}}_k$, which is estimated by Eq. (5.32). By observing the SSL results, our multimodal SSL approach demonstrates an accuracy gain over the audio and video modalities alone. An average error of 10.9 cm is present when estimating the speakers' 3D position using the multimodal approach, which is about twice the length of human mouth, in average. This means that even if such relatively low error is present, no speaker is confused as being another one.

Table 5.3: Average Euclidean distance, in meters, between the speakers' locations found by our SSL algorithms and the true location found by the Kinect's depths stream.

Sequence	Audio	Video	Multimodal
`Two1`	0.1479	0.1826	0.1275
`Two2`	0.1365	0.0963	0.0950
`Two3`	0.1422	0.1358	0.0998
`Two4`	0.2033	0.1522	0.1020
`Two5`	0.1381	0.1278	0.1148
`Two6`	0.1577	0.1784	0.1157
`Three1`	0.2007	0.1643	0.1096
`Three2`	0.1887	0.1311	0.1172
`Three3`	0.2040	0.1218	0.1022
Average	**0.1688**	**0.1434**	**0.1093**

5.5 Conclusions

We have presented a multimodal VAD and SSL algorithm for simultaneous speaker scenarios. The proposed approach fuses the video and audio modalities through an HMM competition scheme. An SVM-based classifier is first used to extract a visual voice-activity score from the optical-flow algorithm run on the users' mouths, and the SRP-PHAT is separately computed for each speaker, to extract their location estimate. A mid-fusion technique is proposed by combining the output of the video-based score with the audio-based feature, that is later evaluated by the HMMs to make a final VAD decision. The final position of the users that are found as active are later estimated by reutilizing the HMMs outputs. Results showed an average 95.06% accuracy for VAD in scenarios with up to three simultaneous speakers, and our SSL approach present an average error of 10.9 cm when localizing such speakers (using a microphone array and a color camera only). Our approach also works for relatively long capture distances (1.4m was the longest tested one) and using a compact microphone array (56cm linear aperture), whereas most works use close capture scenes and/or large-aperture arrays. Additionally, our method presents a higher accuracy than the ones reported in Section 5.2.1, which run on a controlled/simulated environment.

References

See the unified bibliography of the dissertation.

6 GPU-BASED APPROACHES FOR REAL-TIME SOUND SOURCE LOCALIZATION USING THE SRP-PHAT ALGORITHM

V. Peruffo Minotto, C. Rosito Jung, L. Gonzaga da Silveira, and B. Lee, "GPU-based Approaches for Real-Time Sound Source Localization using the SRP-PHAT Algorithm", International Journal of High Performance Computing Applications, first published on August 1, 2012, doi:10.1177/1094342012452166.
Keywords: GPGPU, CUDA, SRP-PHAT, Audio Processing, Cubic Splines Interpolation, High Performance

Abstract

In most microphone array applications, it is necessary to localize sound sources in a noisy and reverberant environment. For that purpose, many different Sound Source Localization (SSL) algorithms have been proposed, where the SRP-PHAT (Steered Response Power using the Phase Transform) has been known as one of the state-of-the-art methods. Its original formulation allows two different practical implementations, one that is computed in the frequency-domain (FDSP), and another in the time-domain (TDSP), which can be enhanced by interpolation. However, the main problem of this algorithm is its high computational cost due to intensive grid scan in search for the sound source. Considering the power of GPUs (Graphics Processing Units) for working with massively parallelizable compute-intensive algorithms, we present two highly scalable GPU-based versions of the SRP-PHAT, one for each formulation, and also an implementation of the cubic splines interpolation in the GPU. These approaches exploit the parallel aspects of the SRP-PHAT, allowing real-time execution for large search grids. Comparing our GPU approaches against traditional multithreaded CPU approaches, results show a speed up of $275\times$ for the FDSP, and $70\times$ for the TDSP with interpolation, when comparing high-end GPUs to high-end CPUs.

6.1 Introduction

Sound source localization (SSL) is an important topic in microphone array signal processing applications, such as beamforming for speech capture in teleconferencing (BRANDSTEIN; WARD, 2001), distant speech recognition (WÖLFEL; MCDONOUGH, 2009), or human computer interaction (DEY; SELVARAJ; LEE, 2011), most of which require real-time processing of the signals. With two microphones, we can find the time difference of arrival (TDOA) using the generalized cross-correlation (GCC) method, which typically involves a frequency weighting function (KNAPP; CARTER, 1976). Given a single TDOA estimate determined by two microphone signals, we can then find a hyperboloid in a three-dimensional space of points that share the same TDOA.

For systems with more than two microphones, we can find the source location from a set of TDOA's from different microphone pairs by finding the optimal intersection of the hyperboloids using the maximum-likelihood (BRANDSTEIN; ADCOCK; SILVERMAN, 1995) or least-squares (BRANDSTEIN; ADCOCK; SILVERMAN, 1997) criteria. These methods rely on the assumption that the individual TDOA estimates are accurate enough to determine the sound source location. Unfortunately, in typical acoustic environments with reverberation, background noise, and interfering sound sources, TDOA estimates can become unreliable making the TDOA-based SSL methods inaccurate (BRANDSTEIN; WARD, 2001).

Alternatively, we can use the *Steered Response Power* (SRP) method (DIBIASE, 2000; OMOLOGO; SVAIZER, 1997) as an extension of the GCC method to multiple microphones. The main idea of the SRP is to steer the microphone array to all possible candidate source locations to find the one with the maximum power, typically using some frequency weighting. In particular, the SRP method with the PHAT frequency weighting (SRP-PHAT) has been popular for its robustness against background noise and reverberation (BRANDSTEIN; WARD, 2001; DIBIASE, 2000; DO; SILVERMAN; YU, 2007). The SRP-PHAT can be computed either in the time domain (DIBIASE, 2000; OMOLOGO; SVAIZER, 1997) (TDSP) or in the frequency domain (ZHANG; ZHANG; FLORENCIO, 2007) (FDSP).

The SRP-based methods belong to the category of the search space-based methods such as Maximum-Likelihood approaches (ZHANG; ZHANG; FLORENCIO, 2007; LEE; KALKER; SCHAFER, 2008) and wideband MUSIC algorithm (TUNG et al., 1999). In general, search space-based techniques have a computational complexity proportional to the number of microphones in the array, and mainly the number of candidate source locations in the discretized search space, making them almost impracticable for real-time speech processing applications.

Solutions to this problem have been proposed by some authors. Lee and Kalker (LEE; KALKER, 2010), showed that by using Intel's Integrated Performance Primitives (IPP, a multi-threaded library) to couple common operations of the FDSP together, one can reduce the CPU usage from 39.2% to 14.4%. Alternatively, Do and Silverman (DO; SILVERMAN, 2007) proposed an iterative hierarchical method called Coarse-to-Fine Region Contraction (CFRC) to reduce the effective number of candidate locations for the SRP-PHAT. Their approach may achieve a runtime reduction up to three orders of magnitude for low signal-to-noise ratio (SNR) scenario, but the speedup decreases as the SNR increases. Furthermore, the sound source is assumed to be stationary. In our previous work (SILVEIRA et al., 2010) we proposed a GPU implementation of the FDSP which exploited a level of parallelism of the algorithm.

Using GPUs for audio signal processing has been a growing practice since the advent of general purpose programmable devices. As an example, many works have been devoted to the one-dimensional Discrete Fourier Transform (DFT) (GOVINDARAJU et al., 2008; CUDA CUFFT LIBRARY, 2011), audio synthesis and Finite Impulse Response (FIR) filtering (SAVIOJA LAURI; VÃd'LIMÃd'KI, 2011), and audio rendering (TSINGOS, 2009). In this chapter we present two approaches for sound source localization by computing the SRP-PHAT using CUDA (CUDA PROGRAMMING GUIDE, 2011): the TDSP with interpolation (TDISP) and the FDSP, which is an improved version of the work in (SILVEIRA et al., 2010). When comparing our GPU-based approaches to CPU-based multithreaded implementations using OpenMP, the FDSP shows a speed-up of $275\times$, and the TDISP shows a speed-up of $70\times$. Additionally, our implementations also provide the advantage of leaving the CPU available for any other kind of processes, and the advantage of achieving the speed-ups without modifying the algorithms formulations, which could allow further speed improvements using methods for search space reduction (DO; SILVERMAN, 2007).

The remainder of this chapter is organized as follows. Section 6.2 reviews the mathematical foundations of the SRP-PHAT algorithm and the interpolation that can be applied to the time domain version. A theoretical analysis of the computational cost of both versions is presented in Section 6.3. Section 6.4 describes the main concepts of programming GPUs using NVIDIA's CUDA model. In Section 6.5 we present our approach for computing the SRP-PHATs using the GPU. Finally, Section 6.6 shows the experimental evaluation that demonstrates the efficiency of our GPU algorithms and Section 6.7 draws some conclusions about the presented work.

6.2 Sound Source Localization

For an array of M microphones, the signal $x_m(t)$ captured at the m^{th} microphone can be modeled as

$$x_m(t) = \alpha_m s(t - t_m^{\mathbf{q}}) + v_m(t), \tag{6.1}$$

where $s(t)$ is the source signal, $v_m(t)$ represents the combination of reverberation, interferences, and background noise, and α_m and $t_m^{\mathbf{q}}$ respectively denote the propagation attenuation and delay of the signal $s(t)$ from a source location $\mathbf{q}$ to the m^{th} microphone. Equivalently, Eq. (6.1) can be represented in the frequency domain as

$$X_m(\omega) = \alpha_m S(\omega) e^{-j\omega\tau_m^{\mathbf{q}}} + V_m(\omega), \tag{6.2}$$

where $\omega = 2\pi f T$ is the normalized frequency in radians corresponding to the frequency in f Hz of the continuous-time signal $x_m(t)$ that is sampled with the sampling period of T seconds, i.e., $x_m[n] = x_m(nT)$ and $\tau_m^{\mathbf{q}} = \frac{t_m^{\mathbf{q}}}{T}$. We assume that the signal is sampled above the Nyquist rate, i.e., $T < \frac{1}{2f_{\text{max}}}$ where f_{max} is the maximum frequency of the signal.

Given a vector of Fourier transforms of observed signals, $\{X_1(\omega), X_2(\omega), \cdots, X_M(\omega)\}$, SSL therefore may be seen as the problem of finding a source location $\mathbf{q}$ that satisfies some optimality criteria such as Maximum-Likelihood (ZHANG; ZHANG; FLORENCIO, 2007; LEE; KALKER, 2010) or maximum power of the filter-and-sum beamformer like the SRP-PHAT method (DIBIASE, 2000).

Next we present the two different versions of the SRP-PHAT that, although being mathematically equivalent, differ in practice. Section 6.2.1 presents the frequency-domain version adapted by Zhang et al. in (ZHANG; ZHANG; FLORENCIO, 2007). Section 6.2.2 shows the time domain version, which was first introduced as the Global Coherence Field (GCF) by Omologo et al. in (OMOLOGO; SVAIZER, 1997).

6.2.1 Frequency Domain SRP-PHAT

The SRP-PHAT method finds a source location by comparing the output powers of PHAT-weighted filter-and-sum beamformers of different potential sound source locations in a search region. In the frequency domain (DIBIASE, 2000), the SRP-PHAT of a point

$\mathbf{q}$ in space is defined as

$$P(\mathbf{q}) = \sum_{m=1}^{M}\sum_{l=1}^{M}\int_{0}^{2\pi} \frac{X_m(\omega)X_l^*(\omega)}{|X_m(\omega)X_l^*(\omega)|} e^{j\omega\tau_{ml}^{\mathbf{q}}} d\omega, \tag{6.3}$$

where $\tau_{ml}^{\mathbf{q}} = \tau_m^{\mathbf{q}} - \tau_l^{\mathbf{q}}$ is the term representing the TDOA between microphones m and l and point $\mathbf{q}$.

By interchanging the order of integration and summation and using its symmetry, Zhang et al. (ZHANG; ZHANG; FLORENCIO, 2007) showed that Eq. (6.3) is mathematically equivalent to

$$P(\mathbf{q}) = \int_{0}^{2\pi} \left| \sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{j\omega\tau_m^{\mathbf{q}}} \right|^2 d\omega, \tag{6.4}$$

which reduces the number of computations of the SRP-PHAT by a factor of M. After $P(\mathbf{q})$ has been computed for all candidate positions using either Eq. (6.3) or Eq. (6.4), we can estimate the sound source location as

$$\hat{q} = \underset{\mathbf{q}\in\mathcal{Q}}{\text{argmax}}\, P(\mathbf{q}), \tag{6.5}$$

where $\mathcal{Q}$ denotes a set of points in space that represent all candidate locations.

6.2.2 Time Domain SRP-PHAT

According to the original proposal in (DIBIASE, 2000), the SRP-PHAT can also be computed by summing the PHAT-weighted Generalized Cross Correlations (GCC-PHAT) of all possible pairs of the set of microphones. The GCC-PHAT (KNAPP; CARTER, 1976) between two microphones m and l may be defined as

$$R_{ml}(\tau) = \frac{1}{2\pi}\int_{0}^{2\pi} \frac{X_m(\omega)X_l^*(\omega)}{|X_k(\omega)X_l^*(\omega)|} e^{j\omega\tau} d\omega, \tag{6.6}$$

We can notice that the SRP-PHAT in Eq. (6.3) is equivalent to the summation of Eq. (6.6) over all microphone pairs except for the scale factor. This allows us to represent the SRP-PHAT in terms of Eq. (6.6)

$$P(\mathbf{q}) = \sum_{m=1}^{M}\sum_{l=m+1}^{M} R_{ml}(\tau_{ml}^{\mathbf{q}}), \tag{6.7}$$

where the number of summations has been further reduced from M^2 to $M(M-1)/2$, without affecting the SSL results, due to the symmetry of the GCC-PHAT in Eq. (6.6)

Although Eqs. (6.3), (6.4), and (6.7) are mathematically equivalent, the computational complexity of a discrete implementation of Eq. (6.7) costs less than the corresponding discrete versions of Eqs. (6.3) or (6.4). This is due to the fact that the GCC-PHAT in Eq. (6.6) is independent of the source location $\mathbf{q}$. Thus, for the double summation in Eq. (6.7), all we need is to access memory positions of $R_{ml}(\tau)$ corresponding to the TDOA $\tau_{ml}^{\mathbf{q}}$ for a proposed source location $\mathbf{q}$ once $R_{ml}(\tau)$ is computed for all microphone pairs. In contrast, the frequency-domain representations in Eqs. (6.3) and (6.4) require complex multiplications and evaluation of an integral for each source location $\mathbf{q}$. The computational complexity for each of these methods will be described in more detail in Section 6.3.

6.2.3 Cubic Spline Interpolation for TDSP

Although the computational complexity of the TDSP algorithm is much less than that of the FDSP, it suffers the loss of its accuracy caused by a quantization process in the TDOAs, i.e., τ in Eq. (6.6) should be an integer as a sample index for the discrete-time domain signal whereas $\tau_m^{\mathbf{q}}$ in Eq. (6.3) can have any precision that the computing environment allows. Therefore, it is often interesting to use some kind of interpolation technique to increase its accuracy. For instance, Tervo and Lokki (TERVO; LOKKI, 2008) compared Parabolic Fitting, Exponential Fitting and Fourier-interpolation methods, concluding that the Exponential Fitting was the best among the three. In (SILVERMAN et al., 2005), Silverman et al. claimed that using 64:1 FIR filter interpolation, the precision loss in the TDSP becomes negligible according to their experiments. However, they did not provide any mathematical proof or analysis for satisfying this condition. Furthermore, in (DO; SILVERMAN, 2007), Do and Silverman used the Cubic Splines Interpolation (CSI) algorithm, which is shown to be as precise as the FIR filter interpolation but faster. In this work we have also chosen the CSI for interpolating the SRP-PHAT for two reasons: it provides reasonably accurate results and is well suited for GPU implementation due to already existing fast GPU-based tridiagonal system solvers, which is a crucial step of the CSI and is detailed next.

First, one may choose among some variations of the CSI, depending on the boundary conditions. We found that varying these conditions has negligible impact on the final result of the TDISP, so we chose the Natural Splines conditions, since it simplifies the GPU implementation as it will shown next. Additionally, from this point on, we treat the GCC-PHAT $\mathbf{R}_{ml}$ as a discrete-time vector instead of a continuous variable.

For the CSI implementation, despite the fact that it is a simple process itself, there are some details when it comes to applying interpolation to the SRP-PHAT algorithm. Since the delay is quantized through the sampling of the continuous signal prior to accessing the vector $\mathbf{R}_{ml}$, the GCC-PHAT values are the candidates for the interpolation. Furthermore, we may notice that when summing the GCC-PHATs using Eq. (6.7), the range of TDOAs that will actually be used to access $\mathbf{R}_{ml}$ is smaller than range of values it represents. In addition, if we compute the GCC-PHAT using the Discrete Fourier Transform (DFT) as in Eq. (6.6), then the values for negative TDOAs are wrapped around and appear at the end of the vector $\mathbf{R}_{ml}$. Therefore, we may chose only a set of its values for the interpolation and properly adjust the values for the negative TDOAs, that is

$$R'_{ml}[i] = R_{ml}[j], \text{ where } \begin{cases} j = i, \text{ if } 0 \leq i \leq V \\ j = N + i, \text{ if } -V \leq i \leq -1 \end{cases} \tag{6.8}$$

for $i = -V, -V+1, \ldots, 0, \ldots, 1, \ldots, V-1, V$, where V is set to the largest possible absolute value of TDOA given the microphone positions and search grid coordinates and N denotes the DFT length.

In particular, it can be noticed that the largest possible TDOA value occurs for the points at the so-called "end-fire" configuration where the points are located on the line that connects the microphone pair with the largest spacing, and that are also not in between them. In this case, the largest possible TDOA can be calculated as

$$V_{\text{max}} = \frac{\|\mathbf{x}_1 - \mathbf{x}_2\|}{CT}, \tag{6.9}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the positions of microphones with the largest spacing, $\|\cdot\|$ denotes the Euclidean distance and C is the speed of sound. Given this, $V = V_{\text{max}}$ is a plausible choice especially when the search region is not static along the entire application. Notice, however, that setting $V = V_{\text{max}}$ requires higher computational cost of the interpolation process especially when the range of the actual TDOA values for the given microphone configuration and search space are much smaller than V_{max}. Considering this we can further decrease the value of V by selecting

$$V = \left\lceil \max_{\substack{\mathbf{q}\in\mathcal{Q}\\ 1\leq m\leq M\\ m+1\leq l\leq M}} \left(\left| \frac{\tau_{ml}^{\mathbf{q}}}{T} \right| \right) \right\rceil, \tag{6.10}$$

which can be computed pre-runtime. However, since it depends on the search region $\mathcal{Q}$, it must be re-computed if the search region is changed dynamically.

Once $\mathbf{R}'_{ml}$ has been computed, it is then possible to interpolate between its values using a CSI algorithm in order to obtain $\mathbf{R}_{ml}^{CSI}$, a length $(2V-1)E+1$ interpolated GCC-PHAT for each pair of microphones with E being the interpolation factor. This can be done by solving a $(2V-2)$ by $(2V-2)$ tridiagonal system, as in Eq. (6.11) and doing some algebraic operations.

$$\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \times \begin{bmatrix} \Phi[1] \\ \Phi[2] \\ \Phi[3] \\ \vdots \\ \Phi[V-3] \\ \Phi[V-2] \\ \Phi[V-1] \end{bmatrix} = \begin{bmatrix} Y[0] \\ Y[1] \\ Y[2] \\ \vdots \\ Y[V-4] \\ Y[V-3] \\ Y[V-2] \end{bmatrix} \tag{6.11}$$

The right-hand side vector $\mathbf{Y}$ can be determined through some operations on $\mathbf{R}'_{ml}$, and the unknown vector $\mathbf{\Phi}$ is used to compute the splines coefficients through other operations. Both proccess involving these vectors are explained in Subsection 6.5.4. Once the interpolation is done, we may use $\mathbf{R}_{ml}^{CSI}$ instead of $\mathbf{R}_{ml}$ in Eq.(6.7) to compute the TDISP.

6.3 Computational Complexity Analysis

With a fixed microphone array and search region, we can compute some parts of the algorithm pre-runtime (LEE; KALKER, 2010). In particular, given the search region $\mathcal{Q}$ and the position of all M microphones, we can precompute all the TDOAs $\tau_{ml}^{\mathbf{q}}$ that will be used along the entire application, in Eq. (6.4) or Eq. (6.7). For this reason, we only consider runtime computations in computational complexity analysis. Additionally, we assume the use of the Fast Fourier Transform (FFT) for the DFT computation, and denote N as the initial number of audio samples used for computing a single frame of the SRP-PHAT, and K as the number of frequency bins for computing the integrals in Eq.(6.4) or in Eq.(6.6), where $K \leq N/2+1$ as a result of the FFT.

6.3.1 FDSP Cost

Recalling that the algorithm depends on N (the FFT size), K, Q (the number of SSL candidates) and the M (the number microphones), we may define the asymptotic computational complexity of the FDSP as being $O(NQM)$. This usually implies a very large computational cost mostly due to Q because $Q \gg N, K, M$ in typical scenarios. In a detailed analysis, a discrete-time implementation of the SRP-PHAT requires the following number of arithmetic operations (additions and multiplications) at runtime (LEE; KALKER, 2010).

- FFT: $\frac{5}{2}MN\log_2 N$ operations
- PHAT: $10MK$ operations ((DO; SILVERMAN; YU, 2007))
- SRP: $8MKQ + 4KQ$ operations.

For example, if search grid points are in a two-dimensional space of 2 m × 2 m with a uniform grid spacing of 0.02 m, we have the total number of $Q = 10000$ candidate locations. With this search space, for a system with $M = 8$ microphones with an equidistant spacing of 0.08 m, $(N = 2048)$-point FFT, and $K = 1024$, the FDSP requires 6.968×10^8 operations per frame.

6.3.2 TDISP Cost

Remembering that for the TDSP the computation of $\mathbf{R}_{ml}$ in Eq.(6.6) is done prior to $P(\mathbf{q})$ in Eq.(6.7), the asymptotic computational cost of the TDSP may be seen as $O(M^2(K+Q))$, and of the TDISP as $O(M^2(K+Q+V))$. Now for a practical example, we first recall that $R_{lm}[n] = R^*_{ml}[N-n]$, and thus, most M^2 terms become $\frac{M(M-1)}{2}$. Additionally, we first analyze separately the TDSP and the CSI. For the TDSP, we have the following.

- FFT: $\frac{5}{2}MN\log_2 N$ operations;
- PHAT: $5NM(M-1)$ operations ((DO; SILVERMAN; YU, 2007));
- IFFT: $\frac{M(M-1)}{2} \cdot \frac{5}{2}N\log_2 N$ operations;
- SRP: $\frac{M(M-1)}{2} \cdot Q$ operations.

For the NCSI algorithm, we have to find the splines coefficients for all the $\frac{M(M-1)}{2}$ GCC-PHATs, and compute the new points using those coefficients. This is done in three separate processes, with the following computational costs:

- Thomas Tridiagonal Solver: $\frac{M(M-1)}{2}(8V-23)$ operations;
- Find Splines Coefficients: $\frac{M(M-1)}{2}(16V-8)$ operations;
- Generate Interpolated points: $\frac{M(M-1)}{2}(12V-6)E$ operations.

Using $E = 10$ (10:1 interpolation), $T = \frac{1}{44100\text{Hz}}$, $V = V_{\text{max}}$, and the same parameters as those in Subsection 6.3.1, the TDSP needs a total of $2,88 \times 10^6$ operations, and the TDISP a total of 3.168×10^6 operations. Notice that even though the TDSP costs about two orders of magnitudes less than the FDSP, it may still be impracticable for real-time applications if large values for Q and/or M are used. Furthermore, as mentioned before, this gain in computational complexity has the trade-off of reducing the algorithm's accuracy (SILVERMAN et al., 2005), and therefore, using the TDISP helps alleviate this effect.

6.4 GPGPU using CUDA

In the last years, the increasing processing power of GPUs led the scientific community to explore non-graphic related computations onto these highly parallel architectures, firstly through vertex and fragment shaders, and more recently programming the device. This practice became even more feasible when NVIDIA released in 2006 their G80 chipset series (CORPORATION, 2006). This new architecture leveraged the first dedicated GPUs in the market that could be used for general purpose computing through NVIDIA's also newly created CUDA programming model (KIRK; HWU, 2010; LINDHOLM et al., 2008). Later in 2010, NVIDIA launched their Fermi family cards (GLASKOWSKY, 2009; CORPORATION, 2011), a successor of the GT200 series GPUs, that brought many advantages over its predecessors and now represent NVIDIA's most powerful cards. For this reason, we will use the Fermi architecture for the explanations in this section.

However, it is important to mention that ATI/AMD also has their programming model, the Stream SDK (SSDK). Furthermore, there is also the OpenCL, an open industry standard that abstracts both CUDA and the SSDK, facilitating heterogeneous computing due to portability and vendor-independence. Given these options, we chose CUDA over SSDK due to implementation practicality and over OpenCL due to performance reasons. Nevertheless, our algorithm could be easily ported to either programming models. Additionally it may be compiled and run in any CUDA-enabled device thanks to the forward and backward compatibility introduced by NVIDIA's Parallel Thread Execution (PTX) (CUDA PROGRAMMING GUIDE, 2011; FERMI COMPATIBILITY GUIDE, 2011), a pseudo-assembly language that acts as a virtual machine between CUDA code and hardware-specific binary code.

From the point of view of hardware models, the NVIDIA GPU architecture has incrementally changed from different series and families in their basic concept. Currently, Fermi cards consist of two-level of hierarchy for the processors and three-level for the memory, starting by the Streaming Multiprocessors (SM) and Global Memory, respectively. As Figure 6.1 abstractedly illustrates, each SM consists of 32 Scalar Processors (SP - also commonly called CUDA cores), a hybrid on-chip region of configurable shared and L1 cache memories, and also texture and constant memories caches. Furthermore, all SMs share an area of DRAM that is cached by a L2 memory. While data in global memory (DRAM) can be accessed by any SP in any SM, data in shared memory is accessible only from within the SPs in a same SM. Since shared memory is on-chip, it is designed to be much faster than global memory, but also being limited to 48kB out of the total 64kB in the hybrid area. The remaining 16kB is used as L1 cache. Notice, however, that this region may be oppositely configured as 16kB of shared memory and 48kB of cache.

As GPU data must come from the computer's global RAM, the PCI-Express (PCI-E) bus is used for the transference. Its current theoretical maximum throughput is 8 GB/s for the 2.0 versions (used by GPUs). Since this bandwidth is much lower than the peak bandwidth between the GPU's global memory and its processors (192GBps theoretical for NVIDIA most recent non-dual cards), data transfers between CPU and GPU should be minimized in size and frequency (CUDA BEST PRACTICES GUIDE, 2011). This allows for the GPU's fast access rate to be exploited and no transfer overheads to be created. Another feature that must be exploited in a GPU software is the fast thread switching. While a thread switch is very costly on the CPU, the GPU can handle this task with more ease. It is therefore encouraged to create more threads than the number of physical processors available. Such overload promotes high concurrency among all threads, and

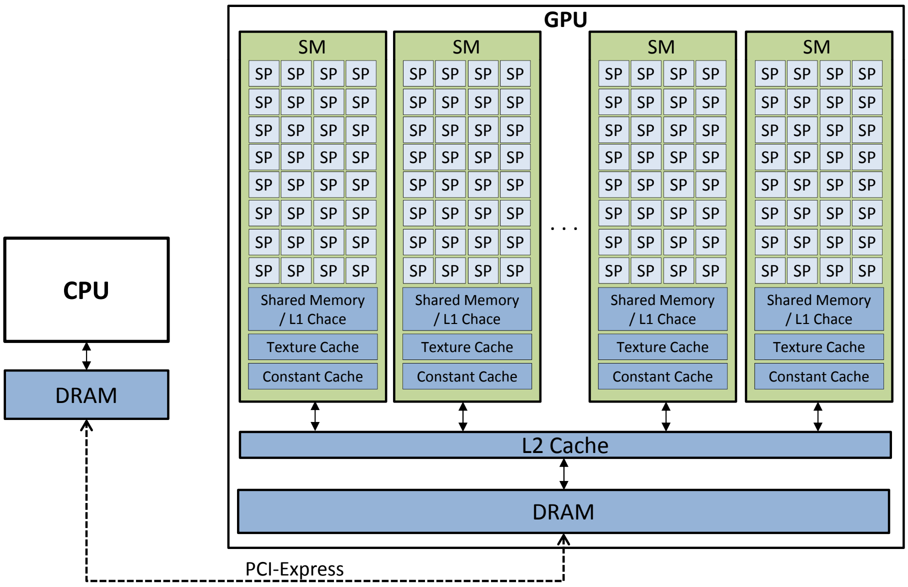


Figure 6.1: Abstracted architecture of a NVIDIA's Fermi GPU.

occupancy of the hardware, which is good, since some threads can be scheduled while others wait for memory transfers (CUDA BEST PRACTICES GUIDE, 2011). Both of these best practices are well exploited by the developed GPU algorithm, and will be better highlighted in later sections.

Now from the point of view of software model, CUDA is a minor extension of the C and C++ languages that allows the writing of heterogeneous software, that is, programs that use both the CPU and GPU for its execution. This is done by designing GPU-turned functions called *kernels*, which executes in parallel across a set of threads. As represented in Figure 6.2, each thread has a private local memory, which resides in the devices global memory. These threads are organized into a hierarchy by the programmer. A group of threads is called a block, and a group of blocks is called a grid. Thread blocks are sets of concurrent threads that may cooperate among themselves through barrier synchronization and access to the shared memory. The management of the threads (creating, scheduling and termination) is done automatically by the hardware at runtime. However, the programmer has to specify, for every kernel invocation, the size of the grid that will be executed by the kernel, and the size of its blocks. Also, only one grid may be designated to a kernel, and multiple kernels may be executed in parallel (new feature in the Fermi cards).

These abstractions done by the CUDA programming model easily allow a two-level hierarchical indexing of all threads (similar to nested parallelism), multidimensional data manipulation and sharing, without making the programmer worry about functional correctness. However, some good practices should be adopted since there is some relationship between the software-level thread organization and the way the hardware handle the threads (CUDA BEST PRACTICES GUIDE, 2011). More precisely, each thread block is scheduled to a SM, and then split into groups of 32 threads called *warps*. Pairs of warps are then scheduled to the SPs by two Warp Scheduler units, and then successively run concurrently, in SIMT (single-instruction multiple-thread) fashion, until the whole block is executed. Given this process, the creation of potential divergent code flow between threads should be avoided, since it is something that is handled automatically by the hardware by serializing their execution. This may happen, e.g., when an *if* condition evaluates

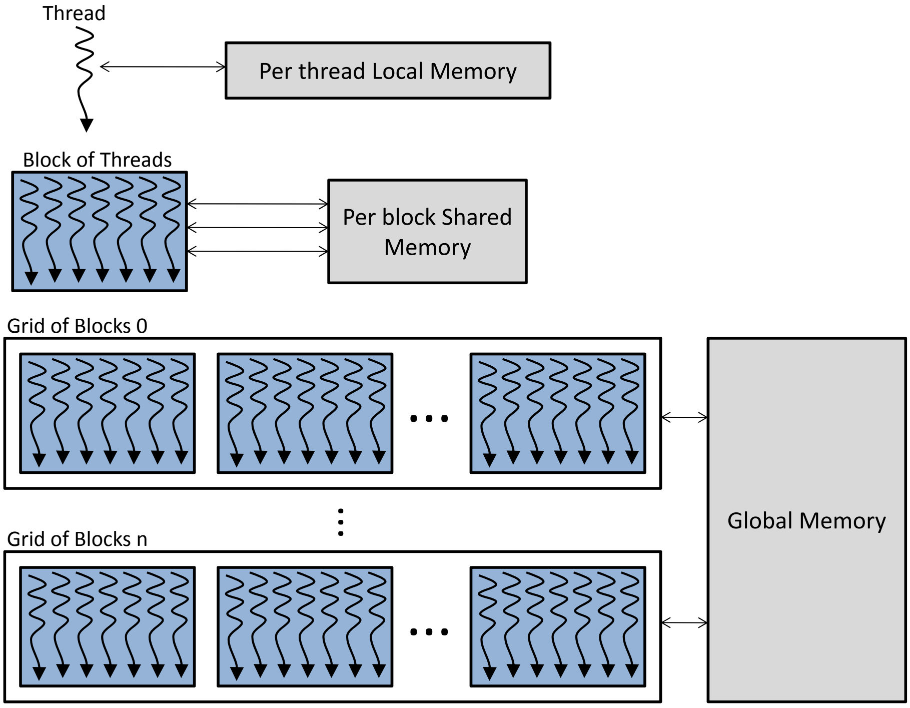


Figure 6.2: CUDA abstraction of threads and memory.

true within a thread and *false* in another one.

Furthermore, some other recommended practices should be prioritized (CUDA BEST PRACTICES GUIDE, 2011), such as creating blocks multiples of 32, so that no warp with less than 32 threads is scheduled, helping increase the occupancy of the GPU. However, the most important practice is to perform coalesced memory access in the GPU's global memory, that is, all of the threads in a half-warp should access global memory at the same time, which is achieved by some coding patterns described in (CUDA BEST PRACTICES GUIDE, 2011), and may have some variations between different devices. In short, the simplest and most efficient way of achieving coalesced access is by making adjacent threads in warp access adjacent words in the global memory, without offsetting the accesses.

6.5 GPU-based SRP-PHAT

The implementation of GPU-based algorithms should focus mainly on exploiting as much parallelism as possible (CUDA BEST PRACTICES GUIDE, 2011). Based on the CUDA's abstraction of the GPU, it is possible to easily exploit two levels of parallelism in GPU-based routines (i.e., parallelize two nested loops). For FDSP and TDSP, this was done in a similar way. While the first level of parallelism is equal in both versions, the second one is different. Therefore, Subsection 6.5.1 presents the first parallelization both versions share, and Subsections 6.5.2 and 6.5.3 present the individual second level of parallelization of each version.

6.5.1 Common Parallelization

As described in Section 6.2.1, the SRP-PHAT must be evaluated once for every point in a search space $\mathcal{Q}$, containing Q points. In CPU-based implementations, this is done in a serial manner, or at most, parallelized among the cores of the CPU. In our application, we parallelize the search space scanning as illustrated in Figure 6.3. At a programming level this is the first parallelization in the software's flow, thus implying in distributing each candidate point to a single block of the kernel's grid. More precisely, this is done before the kernel invocation by specifying its grid size as the same size of the search space.

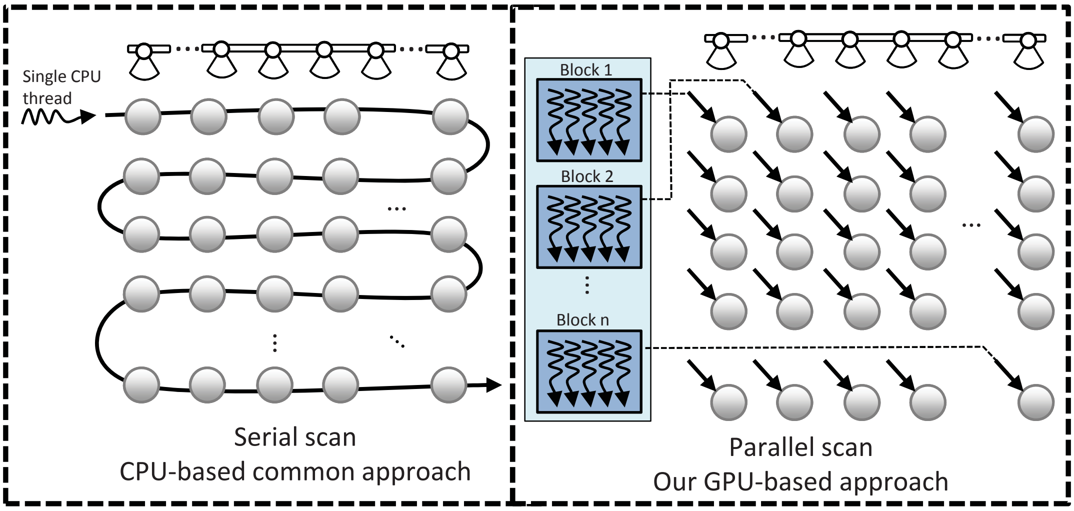


Figure 6.3: Serial scan *vs.* our proposed parallel scan (first level of parallelization).

In this approach, each block will be scheduled to an SM of the GPU, which is an expected behavior, as described in Section 6.4. This makes the algorithm scalable to the GPU being used. The more SMs available, more parallelism will be achievable, while fewer SMs imply in more concurrency (and less parallelism).

It is important to note that this parallel grid search approach can be applied to any search space-based SSL methods in general (including (ZHANG; ZHANG; FLORENCIO, 2007; LEE; KALKER; SCHAFER, 2008; TUNG et al., 1999)) only with the difference in the actual implementation of the computation at each grid. In the next two sections, the case for the SRP-PHAT will be presented.

6.5.2 Parallelization Approach for FDSP

Since the first level of parallelism is done at a grid-to-block level for the parallel space scanning, the second one is done by splitting each block into groups of threads for computing each point's SRP-PHAT. Following the formulation of the SRP-PHAT, given by Eq. (6.4), it can be noticed that its outermost iteration process is the integral (which in practice is a summation), and therefore it is the main candidate to be parallelized at a block-to-thread level. Figure 6.4 illustrates how the parallelization was done.

Each block responsible for a point $\mathbf{q}$ is split into S new threads, each of which computing a portion of the FDSP's integral (summation in practice). This implies in a total of K/S iterations in order to calculate the whole integral (instead of K, as in the serial version). More precisely, in each iteration b, each s-indexed thread computes $Z(\omega) = \left|\sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{j\omega\tau_m^{\mathbf{q}}}\right|^2$, for $\omega = s + (b-1)S$, treating $X_m(\omega)$ as a discrete-time vector (for simplicity) and using one-based indexing as in the picture. Dividing each thread's work like this allows them to access adjacent positions of the $X_m(\omega)$ vector, leveraging from coalesced global memory accesses. For incrementing the computed value $Z(\omega)$ between successive iterations (process represented by the C++ operator $+=$), we

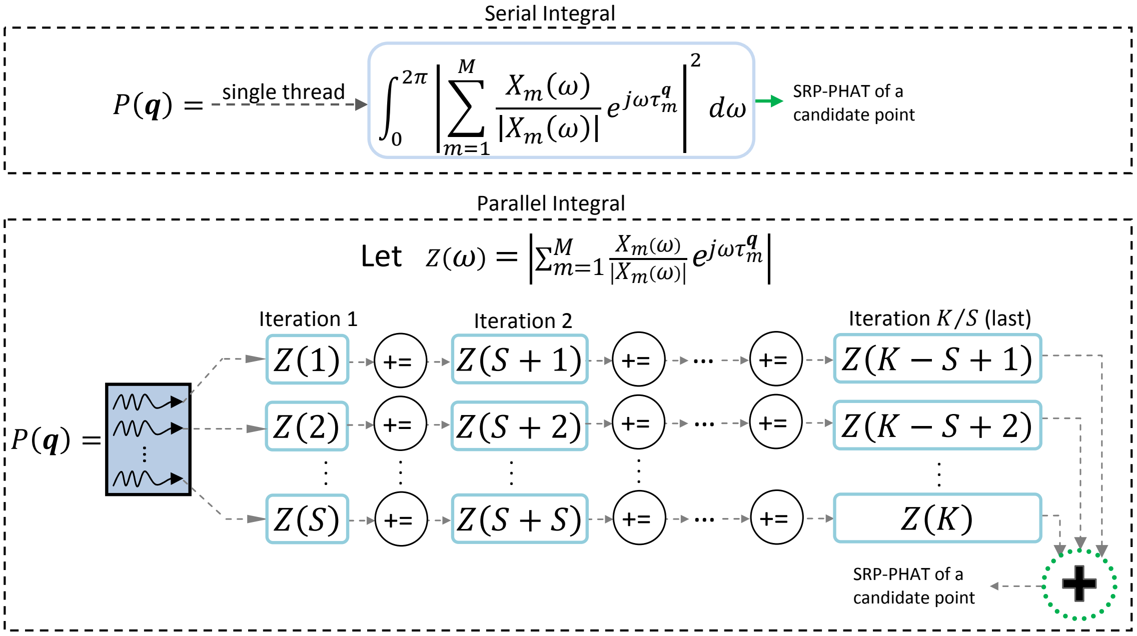

Serial Integral
$P(\boldsymbol{q}) =$
single thread
$\int_0^{2\pi} \left| \sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{j\omega\tau_m^{\boldsymbol{q}}} \right|^2 d\omega$
SRP-PHAT of a candidate point
Parallel Integral
Let $Z(\omega) = \left| \sum_{m=1}^{M} \frac{X_m(\omega)}{|X_m(\omega)|} e^{j\omega\tau_m^{\boldsymbol{q}}} \right|$
Iteration 1
Iteration 2
Iteration K/S (last)
$P(\boldsymbol{q}) =$
$Z(1)$
$Z(2)$
$Z(S)$
+=
$Z(S+1)$
$Z(S+2)$
$Z(S+S)$
$Z(K-S+1)$
$Z(K-S+2)$
$Z(K)$
SRP-PHAT of a candidate point

Figure 6.4: Serial integral *vs.* our proposed parallel integral (second level of parallelization).

use, for each thread, one position of a shared memory array with length S. By the end of the last iteration, each position of the array will hold a partial result of the FDSP, which than have to be summed into one, so that the block finishes computing the SRP-PHAT of the point it represents. Notice, however, that this summation has to be synchronized among the threads so that no wrong data is read/written, and for that, we use the parallel sum reduction algorithm described by Mark Harris in the documentation of the "reduction" example that comes with NVIDIA's GPU Computing SDK (HARRIS, 2011). This is a very efficient reduction algorithm, especially when reducing shared memory arrays, which is our case.

Aside from our parallelization approach, notice that the innermost iterations (the summation) could be the one to be parallelized, but this would obviously imply poorer performance, once it iterates only M times. That is, the number of microphones is generally small (BRANDSTEIN; WARD, 2001) compared to K, and thus less threads would be created. Another issue is that the number of threads hardly would be multiple of 32, what is something necessary to cope with the warp size recommendation mentioned in Section 6.4. For this reason, we chose $S = 64$, which is the multiple of 32 that has experimentally shown to be the best choice among all possible multiples.

6.5.3 Parallelization Approach for TDSP

The TDSP, different from the FDSP, can be divided into two separate stages. The first part is where the GCC-PHATs are precomputed using Eq. (6.6), and the second is the evaluation of the SRP-PHAT for each candidate source location, using Eq. (6.7). For each stage, a separate kernel is developed, since their grid and block sizes must be different in order to achieve higher performance. Figure 6.5 illustrates how this first stage is processed.

For this parallel computation of Eq.(6.6), we also exploit two levels of parallelism. Each GCC-PHAT is assigned to a block, resulting in a grid of $\frac{M(M-1)}{2}$ blocks. Each

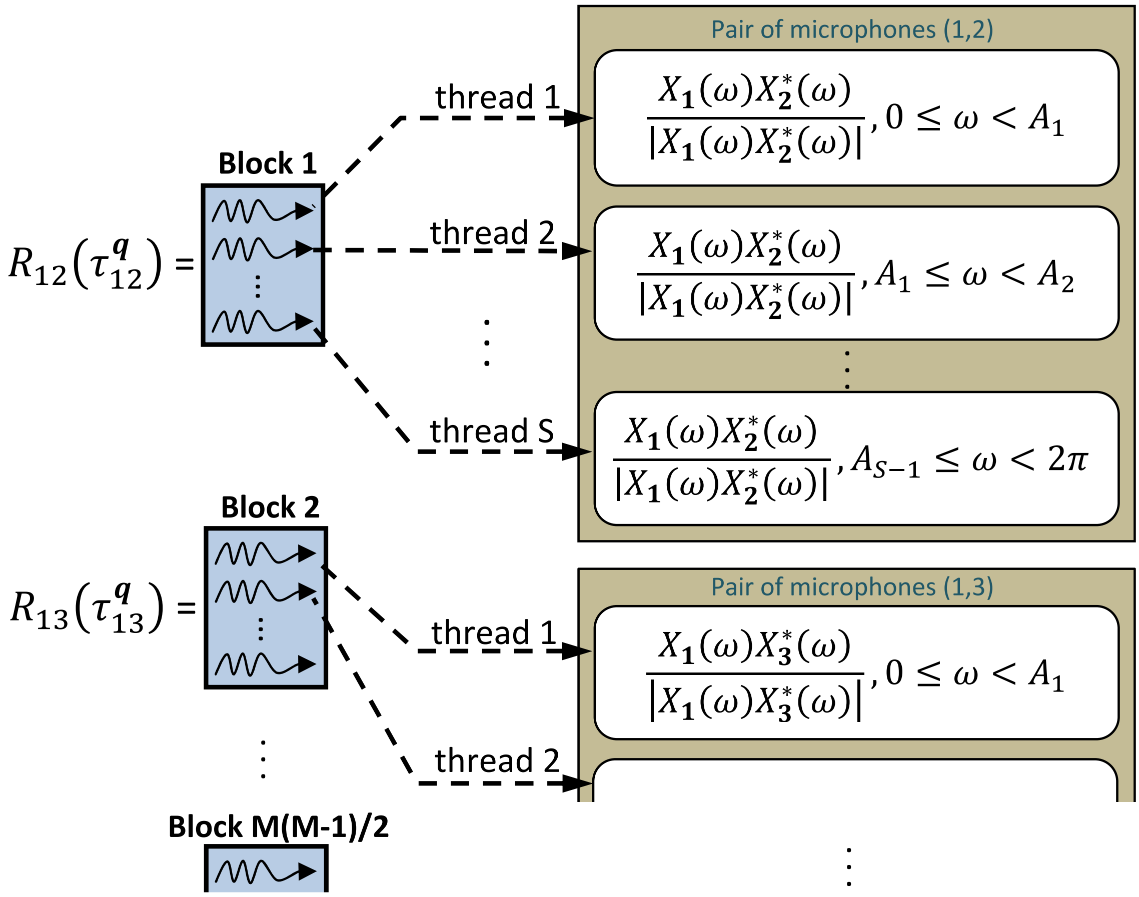

Pair of microphones (1,2)
thread 1
thread 2
thread S
Block 1
$R_{12}(\tau_{12}^{q}) =$
$\frac{X_1(\omega)X_2^*(\omega)}{|X_1(\omega)X_2^*(\omega)|}, 0 \leq \omega < A_1$
$\frac{X_1(\omega)X_2^*(\omega)}{|X_1(\omega)X_2^*(\omega)|}, A_1 \leq \omega < A_2$
$\frac{X_1(\omega)X_2^*(\omega)}{|X_1(\omega)X_2^*(\omega)|}, A_{S-1} \leq \omega < 2\pi$
Block 2
$R_{13}(\tau_{13}^{q}) =$
Pair of microphones (1,3)
thread 1
thread 2
$\frac{X_1(\omega)X_3^*(\omega)}{|X_1(\omega)X_3^*(\omega)|}, 0 \leq \omega < A_1$
Block M(M-1)/2

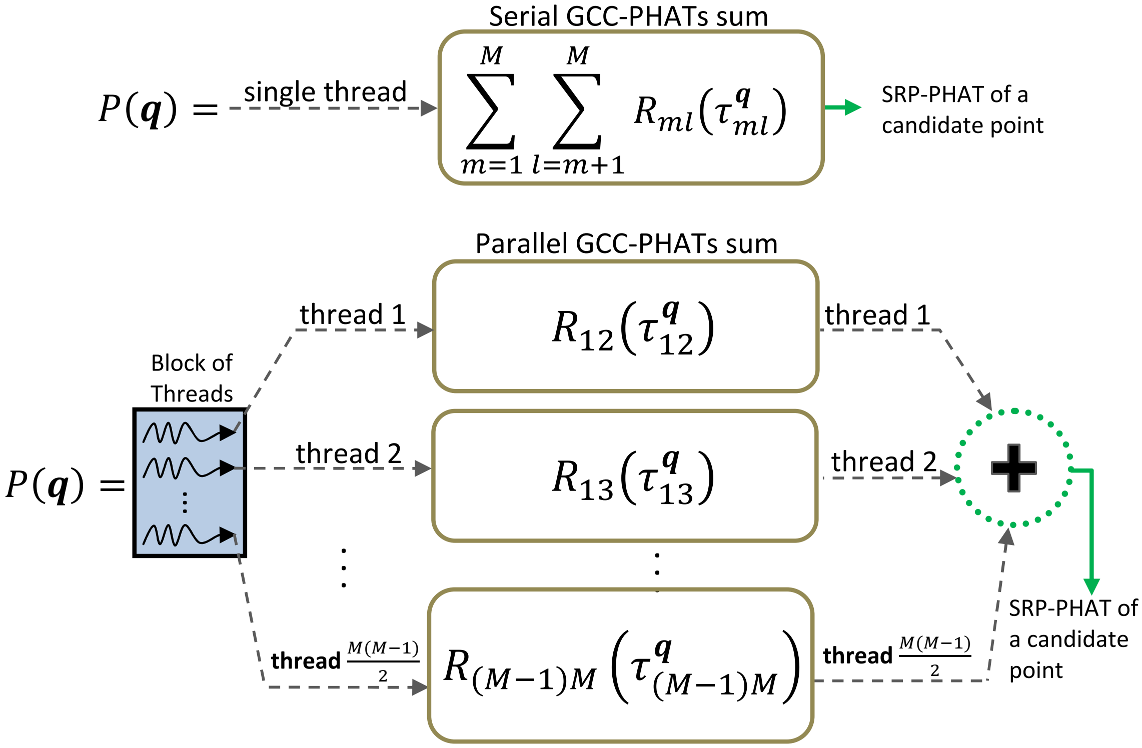
Figure 6.5: Parallel computation of the GCC-PHATS.

block of threads is then divided into groups of S threads, each of which does a portion of the frequency domain computations ($S = 64$, same as for the FDSP). These frequency domain computations are the PHAT-weighted Cross Power Spectrum (CPS) of the pair of microphones the block represents. Here, the division of each block's work is given in the same way as described in Section 6.5.2, except that the starting grid has $\frac{M(M-1)}{2}$ threads instead of Q, and the final results are not reduced in the end. Finally, to finish calculating the GCC-PHAT of each block, we do the Inverse Fast Fourier Transform (IFFT) of the corresponding CPS. This is done using NVIDIA's CUFFT library, which provides faster FFT/IFFT algorithms than other known CPU-based libraries (CUDA CUFFT LIBRARY, 2011).

For the next step of the algorithm, first note that after launching the kernel of Figure 6.5, a synchronization barrier is unavoidable due to the data dependency nature of the algorithm (to perform the TDSP, the GCC-PHATs computation must finish first). Given that, once the GCC-PHATs have been computed, it is then possible to compute the SRP-PHAT of each point starting with the grid search parallelization described in Section 6.5.1 (recalling it is used for both the FDSP and TDSP). For each candidate source, located at $\mathbf{q}$, we need to sum an element of all $\frac{M(M-1)}{2}$ GCC-PHATs, where each element's index corresponds to the TDOA $\tau_{ml}^{\mathbf{q}}$ between the pair of microphones ml and the location $\mathbf{q}$. This step is suited for the second level of parallelism of the TDSP and is illustrated in Figure 6.6. It is done similarly to the parallelization of the FDSP: each thread of the block is responsible for a portion of the SRP-PHAT, which will be summed together at the end. However, each thread's task in this case is to retrieve values from the precomputed

GCC-PHATs, based on the microphone pairs they represent.

One thing to notice is that, for TDSP, the block size is $\frac{M(M-1)}{2}$ instead of $S = 64$. This is a drawback that degrades the algorithm performance since the warp size recommendation previously described is not satisfied ($\frac{M(M-1)}{2}$ will hardly be multiple of 32). Additionally, when accessing the memory in $\mathbf{R}_{ml}$ through the process in Figure 6.6, coalesced access to the global memory is not fulfiled. That is, reading $\mathbf{R}_{12}, \mathbf{R}_{13}, ..., \mathbf{R}_{ml}$ is not done sequentially, and therefore, does not leverage from the fact that when accessing memory positions that are coalesced, the GPU may read one whole sequential region with only one transaction. In our approach, the GPU issues one transaction per read. One way we alleviate this problem is by mapping $\mathbf{R}_{ml}$ into texture memory. While texture memory is also mapped in the DRAM, it has a separate on-chip texture cache (recall Figure 6.1) that may provide fast access to frequently read values in $\mathbf{R}_{ml}$.



Serial GCC-PHATs sum
$P(\boldsymbol{q}) =$
single thread
$\sum_{m=1}^{M}\sum_{l=m+1}^{M} R_{ml}(\tau_{ml}^{\boldsymbol{q}})$
SRP-PHAT of a candidate point
Parallel GCC-PHATs sum
Block of Threads
$P(\boldsymbol{q}) =$
thread 1
thread 2
thread $\frac{M(M-1)}{2}$
$R_{12}(\tau_{12}^{\boldsymbol{q}})$
$R_{13}(\tau_{13}^{\boldsymbol{q}})$
$R_{(M-1)M}(\tau_{(M-1)M}^{\boldsymbol{q}})$
thread 1
thread 2
thread $\frac{M(M-1)}{2}$
SRP-PHAT of a candidate point

Figure 6.6: General GCC-PHATs summation approach vs. our proposed parallel one (second level of parallelization).

6.5.4 Parallelization of the Cubic Splines Interpolation

The critical point in parallelizing the CSI in the GPU is solving the Tridiagonal System (created from the values in $\mathbf{R}'_{ml}$), since there are dependencies between adjacent loop iterations of the algorithm. Other steps are easier to implement, once they summarize to mathematical operations and vector manipulations. In view of that, we use a recently developed GPU-based tridiagonal solver that is available in the CUDPP library, a hybridization of Cyclic Reduction and Parallel Cyclic Reduction (CR+PCR), which is fully described in (ZHANG; COHEN; OWENS, 2010). This algorithm solves many large different tridiagonal systems in parallel, which is exactly the case of the TDISP: we have $\frac{M(M-1)}{2}$ different systems of order $V-2$ to solve.

However, before we can apply the CR+PCR algorithm, the tridiagonal system must be prepared by firstly extracting $\mathbf{R}'_{ml}$ from $\mathbf{R}_{ml}$, as in Eq. (6.8). This is done using CUDA's built in asynchronous memory copies from GPU memory to GPU memory, and thus is fast process (CUDA BEST PRACTICES GUIDE, 2011). After that, we do some algebraic manipulations in $\mathbf{R}'_{ml}$ to create the right-hand vector $\mathbf{Y}$ of the system, as illustrated in Figure 6.7. The creation of $\mathbf{Y}$ is done by a kernel composed by a grid of $\frac{M(M-1)}{2}$ blocks, since each tridiagonal system is related to one pair of microphones. Each block is divided into $2V-2$ threads that will each compute an element of $\mathbf{Y}$. Notice that for each element of $\mathbf{Y}$, we must have three memory accesses into $\mathbf{R}'_{ml}$ which may be slow. For this reason, during the kernel's initiation, the entire $\mathbf{R}'_{ml}$ array is loaded into the shared memory. Finally, for the creation of the coefficient matrix in Eq. (6.11), we allocate the values pre-runtime, since they do not change over different systems.

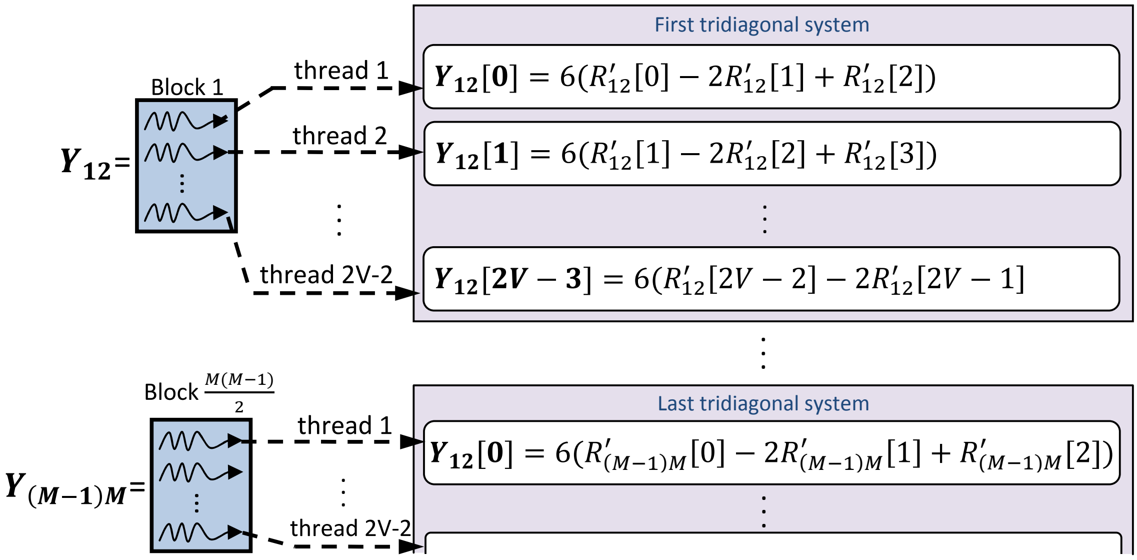

First tridiagonal system
Block 1
$\boldsymbol{Y}_{12}$=
thread 1
thread 2
thread 2V-2
$\boldsymbol{Y}_{12}[\mathbf{0}] = 6(R'_{12}[0] - 2R'_{12}[1] + R'_{12}[2])$
$\boldsymbol{Y}_{12}[\mathbf{1}] = 6(R'_{12}[1] - 2R'_{12}[2] + R'_{12}[3])$
$\boldsymbol{Y}_{12}[\mathbf{2V-3}] = 6(R'_{12}[2V-2] - 2R'_{12}[2V-1]$
Block $\frac{M(M-1)}{2}$
$\boldsymbol{Y}_{(M-1)M}$=
thread 1
thread 2V-2
Last tridiagonal system
$\boldsymbol{Y}_{12}[\mathbf{0}] = 6(R'_{(M-1)M}[0] - 2R'_{(M-1)M}[1] + R'_{(M-1)M}[2])$

Figure 6.7: Parallel preparation of all tridiagonal systems.

After the preparation of the tridiagonal system, the CUDPP's CR+PCR algorithm is executed in order to obtain the unknowns Φ vector from Eq. (6.11). Using Φ we may determine the four coefficients, a_i, b_i, c_i, d_i of the $2V-2$ splines of each GCC-PHAT ($i = 0, 1, \cdots, 2V-2$). This is something needed before $\mathbf{R}'_{ml}$ may be interpolated into $\mathbf{R}^{CSI}_{ml}$. In our approach these coefficients are determined in the same kernel used for the interpolation. Figure 6.8 illustrates this process. Finally, after the execution of this routine, we may use $\mathbf{R}^{CSI}_{ml}$ instead of $\mathbf{R}'_{ml}$ in the approach described in Figure 6.6 to compute the TDISP.

In an overview of our whole GPU-based NCSI approach, it is important to note that even though there are many accesses to the GPU's global memory, they are all performed in a coalesced fashion, which is the optimal way to do it. Furthermore, an advantage of having the NCSI implemented in the GPU is that there is no need to transfer data between the CPU and the GPU after the process in Figure 6.5, what would be highly time consuming. This means that even if we get no speed-up using this approach, it is still better than having to transfer data through the PCI-E and computing it on the CPU.

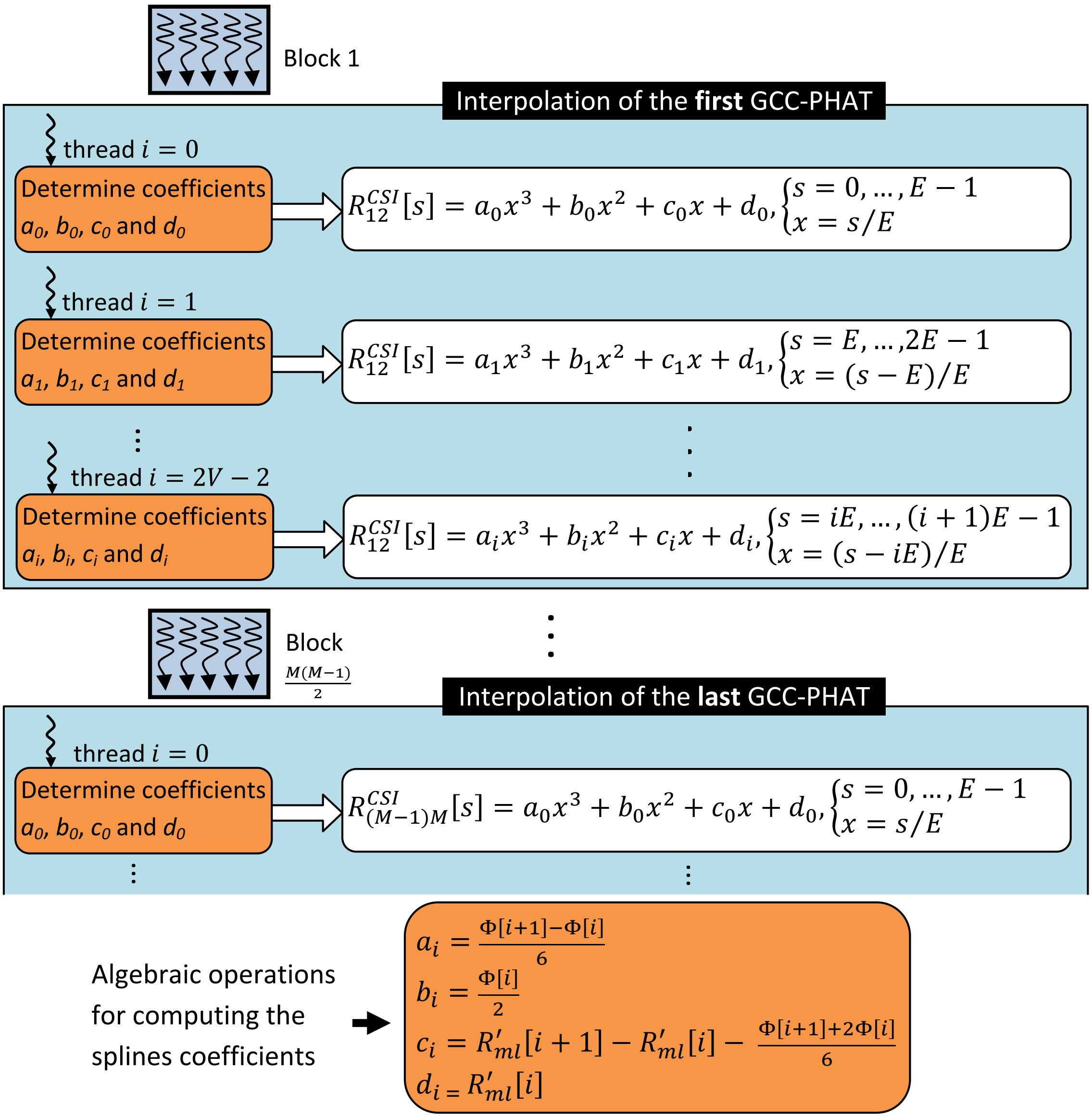

Block 1
Interpolation of the **first** GCC-PHAT
thread $i = 0$
Determine coefficients a_0, b_0, c_0 and d_0
$R_{12}^{CSI}[s] = a_0x^3 + b_0x^2 + c_0x + d_0, \begin{cases} s = 0, \ldots, E-1 \\ x = s/E \end{cases}$
thread $i = 1$
Determine coefficients a_1, b_1, c_1 and d_1
$R_{12}^{CSI}[s] = a_1x^3 + b_1x^2 + c_1x + d_1, \begin{cases} s = E, \ldots, 2E-1 \\ x = (s-E)/E \end{cases}$
thread $i = 2V - 2$
Determine coefficients a_i, b_i, c_i and d_i
$R_{12}^{CSI}[s] = a_ix^3 + b_ix^2 + c_ix + d_i, \begin{cases} s = iE, \ldots, (i+1)E-1 \\ x = (s-iE)/E \end{cases}$
Block $\frac{M(M-1)}{2}$
Interpolation of the **last** GCC-PHAT
thread $i = 0$
Determine coefficients a_0, b_0, c_0 and d_0
$R_{(M-1)M}^{CSI}[s] = a_0x^3 + b_0x^2 + c_0x + d_0, \begin{cases} s = 0, \ldots, E-1 \\ x = s/E \end{cases}$
Algebraic operations for computing the splines coefficients
$a_i = \frac{\Phi[i+1]-\Phi[i]}{6}$
$b_i = \frac{\Phi[i]}{2}$
$c_i = R'_{ml}[i+1] - R'_{ml}[i] - \frac{\Phi[i+1]+2\Phi[i]}{6}$
$d_i = R'_{ml}[i]$

Figure 6.8: Parallel computation of splines coefficients and interpolated GCC-PHATs.

6.6 Experimental Evaluation

In order to evaluate the performance of our GPU SRP-PHATs we have planned a set of experiments. We used CUDA/C++ implementations for both versions of the algorithm in order to make comparisons of their execution time while running on different devices: a Intel Core I7-950 CPU, a GeForce GTS 360M GPU, and a GeForce GTX 570 GPU. The used CPU is equipped with 4 cores that run on a frequency range of 3.06 - 3.33 GHz, support 8 threads via hyperthreading, and is nowadays considered a high-end processor. The GTS 360M is a mobile GPU equipped with 96 cores, each running on a frequency of 575 MHz, and is nowadays classified as mid-range. Finally, the GTX 570 is a more powerful high-end GPU of the Fermi family and it is equipped with 480 cores, each at a 732 MHz frequency. For the CPU implementation, we have parallelized it using the OpenMP API (CHAPMAN; JOST; PAS, 2007) and, although not explicitly vectorized the implementation, we have set the compiler to automatically generate intrinsic SSE functions when possible. Moreover, for sake of simplicity, we named the experiments run on the GTS 360M as GPU1, on the GTX 570 as GPU2 and on the Core I7 as CPU.

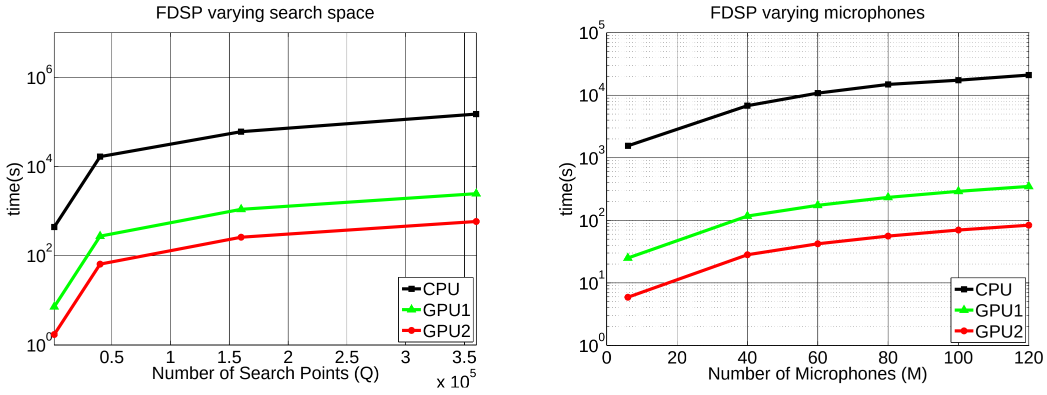


Figure 6.9: Execution time for the FDSP algorithm varying parameters Q (left) and M (right).

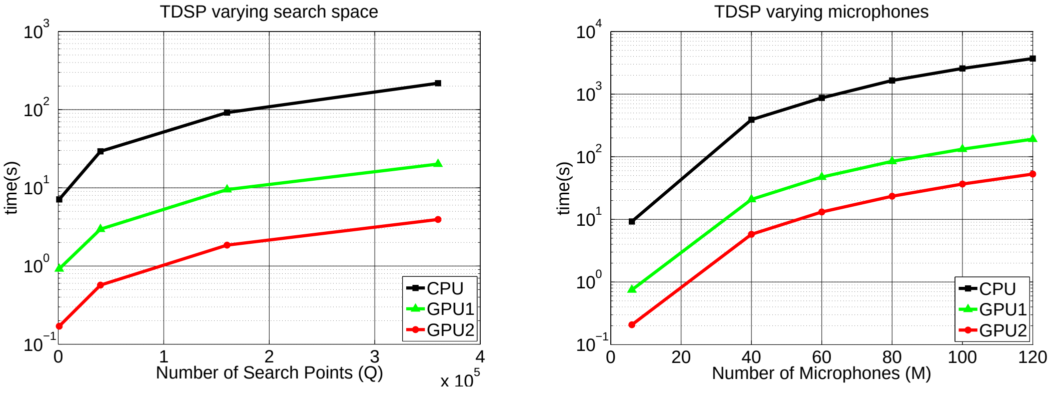


Figure 6.10: Execution time for the TDSP algorithm varying parameters Q (left) and M (right).

Figures 6.9, 6.10 and 6.11 illustrates the execution times for the FDSP, TDSP and TDISP algorithms, all run for CPU, GPU1 and GPU2. The plots on the left illustrates the growing behavior of the runtime as the number of search points increase from $Q = 1000$ to $Q = 360000$ and with a constant $M = 8$. The plots on the right present the same idea, but related to the variation of the number of microphones, $M = 8, ..., 120$ and fixing $Q = 3600$. For the measurements, the average execution time of 100 consecutive runs was taken for each parametrization. For the remaining parameters, we set, $N = 4096$, $K = 2048$, $S = 64$, $E = 10$ (same as (DO; SILVERMAN, 2007)) and $V = V_{max} = 72$ (due to $M = 8$, an equidistant spacing of 0.08 m and a sampling frequency of 44100Hz). Furthermore, the plots' vertical axes use logarithmic scale for better visualization of the time differences.

We may observe that for all the presented experiments, and for the three algorithms, the GPU versions outperform the CPU one. We achieved runtime reductions around to $275\times$ for the FDSP algorithm, and reductions around $70\times$ for the TDSP and TDISP algorithm. For more detailed comparison, Tables 6.1 and 6.2 show the exact measured speedups of the experiments GPU1 and GPU2 compared to CPU, for the same parameterizations as those of the plots.

Analyzing the speed gains, we first notice that they are higher for the FDSP. This may be explained mainly by two reasons. First, the TDSP (TDISP too) inevitably requires a

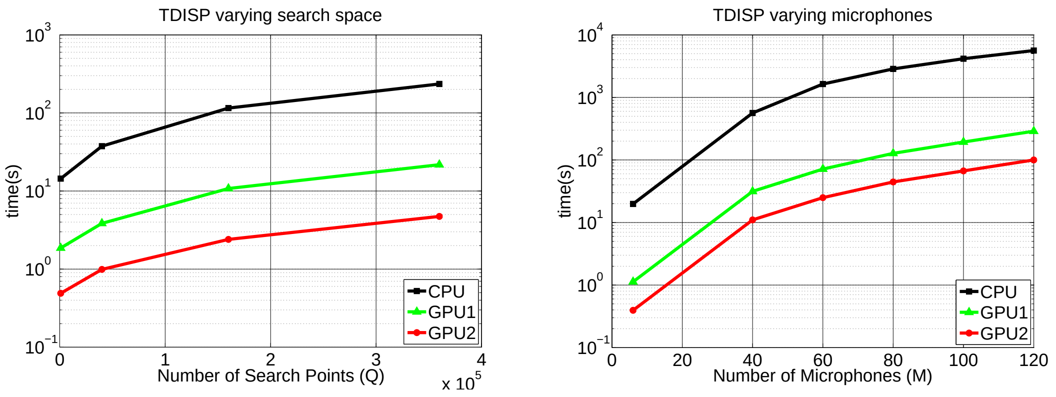


Figure 6.11: Execution time for the TDISP algorithm varying parameters Q (left) and M (right).

		Search Space Size (Q)			
		1000	40000	160000	360000
FDSP	GPU1	$61.24\times$	$60.61\times$	$59.27\times$	$60.66\times$
	GPU2	$276.01\times$	$276.04\times$	$275.7\times$	$276.43\times$
TDSP	GPU1	$7.72\times$	$9.88\times$	$9.67\times$	$10.8\times$
	GPU2	$41.79\times$	$51.33\times$	$49.62\times$	$55.24\times$
TDISP	GPU1	$6.94\times$	$8.54\times$	$8.73\times$	$9.31\times$
	GPU2	$29.41\times$	$37.89\times$	$48.11\times$	$49.68\times$

Table 6.1: Speedups of GPU1 and GPU2 compared to CPU for different search spaces.

high amount of memory accesses to the GPU's global memory in a non-sequential fashion (recall this is related to the TDOAs $\tau_{ml}^{\mathbf{q}}$). This ends up violating the very important performance pattern previously described in Section 6.4: coalesced memory accesses. Secondly, the block size of the TDSP kernel (recall Fig. 6.6) is not multiple of 32, which is also not encouraged due to the warp size restriction also mentioned before. Nevertheless, the GPU TDSP and TDISP still benefit a lot from aspects such as the GPU-based FFT/IFFT algorithms and the parallel computation of each point's SRP-PHAT, still providing a significant speedup over a CPU version.

Additionally, for the TDSP and TDISP, the speedups are higher when varying the parameter M than when varying the parameter Q. This may be explained by the fact that when a high number of microphones is used, the TDSP and the interpolation routines benefit from more GPU power, which mainly happens during the computation of the

		Number of Microphones (M)					
		8	40	60	80	100	120
FDSP	GPU1	$62.81\times$	$59.34\times$	$61.93\times$	$62.43\times$	$59.50\times$	$61.42\times$
	GPU2	$268.88\times$	$269.89\times$	$267.17\times$	$268.27\times$	$269.01\times$	$270.46\times$
TDSP	GPU1	$13.93\times$	$18.72\times$	$18.39\times$	$19.52\times$	$19.40\times$	$19.98\times$
	GPU2	$44.50\times$	$67.59\times$	$66.38\times$	$70.47\times$	$70.03\times$	$69.97\times$
TDISP	GPU1	$12.26\times$	$17.53\times$	$17.64\times$	$17.37\times$	$18.44\times$	$18.88\times$
	GPU2	$42.21\times$	$51.27\times$	$62.59\times$	$61.97\times$	$61.14\times$	$56.03\times$

Table 6.2: Speedups of GPU1 and GPU2 compared to CPU for increasing number of microphones.

GCC-PHATs (Figure 6.5) and during the CSI kernel (Figure 6.8). When $M = 8$, for example, the grid size will be 28 for those routines, which provides a low occupancy of the GPU, specially the GTX 570, that has 480 cores. Oppositely, when $M = 120$, the grid size will be 7140, providing more occupancy of the GPU.

An additional observation is that the speedups, for the TDSP, are higher when interpolation is not being used. This happens because the GPU-based CSI does not provide a high speedup itself, causing the overall TDISP speedups to drop. Figure 6.12 shows a comparison of GPU1, GPU2 and CPU for the CSI alone as well as how much of the TDISP is occupied by the CSI. In the figure, we may notice that, although the CSI alone indeed provides a speedup, it represents higher proportion of the TDISP in the GPU than it does in the CPU. This explained by the fact that the speedup of the TDSP is much higher than the CSI (70x against 11x in the best scenario). In fact, the CSI algorithm is not very favorable for a GPU implementation. Solving a tridiagonal system requires direct dependency between adjacent loop iterations, making it hard to be parallelized. This reflects directly into the CR+PCR algorithm we use (ZHANG; COHEN; OWENS, 2010), to which it is reported speedups around 12× over CPU versions. Moreover all kernels related to the CSI also have block sizes not multiple of 32 and do not perfectly achieve the recommended memory access patterns. Nevertheless, it is important to mention that it is still highly beneficial to perform the CSI in the GPU, for the reason that transferring all the GCC-PHATs back to CPU would be much more time-consuming.

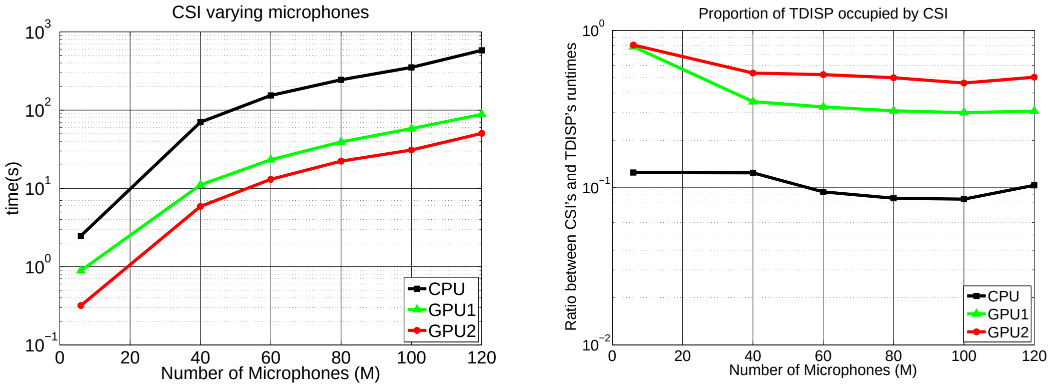


Figure 6.12: Execution time for the CSI algorithm (left) and proportion of the TDISP that is occupied by the CSI (right). Both graphs are for varying number of microphones.

In an overview of our algorithm, its main advantage may be seen as the high speed gains over its CPU version, but we highlight that another benefit of using the GPU for the heavy processing is that the CPU is left free for any other tasks that might be run parallel to the SRP-PHATs. An example would be multimodal speaker localization using audio and video information, in which the CPU could be used for processing video data. Furthermore, we may observe that our algorithms are highly scalable to the GPU's available power, once the runtimes were higher for the GTX 570. This implies that one can always appeal to better devices when faster executions are needed, i.e., when higher values of Q and/or M are used. However, it is interesting to notice that even the GTS 360M runs the SRP-PHATs faster than the Core I7-950, which is a high-end CPU.

6.7 Conclusions

In this chapter we presented efficient GPU approaches for both the frequency-domain and time-domain versions of the SRP-PHAT. These formulations of the algorithm in practice differ in their computational complexity and precision, making them individually preferable in different situations. Although the accuracy of the time-domain version is lower, it is a common practice to enhance it using interpolation techniques. For that reason, we also presented here a GPU approach for computing the one-dimensional cubic splines interpolation algorithm. When comparing our algorithms using a GTX 570 and a Core I7-950, our experimental results indicate that the TD version reaches speedups up to $70\times$, the FD up to $275\times$ and the interpolation up to $11\times$. Furthermore, using our proposed implementations gives the additional advantages of leaving the CPU free to process any other task parallel to the GPU, and also allow for any further modifications of the algorithm that may improve its speed, once their original formulations were not yet altered. Finally, future work will aim at alternatives for improving the TD version, for it is a memory problematic algorithm for the GPU.

References

See the unified bibliography of the dissertation.

7 CONSIDERATIONS AND FUTURE WORK

Voice Activity Detection and Sound Source Localization play an important role in speech-based HCI systems. Taking the applicability of this field of research and its related open problems, this dissertation proposed different ways of performing single and multiple speakers VAD and SSL. Our work was presented as a compilation of already produced articles. We started by describing a proposed HMM-based unimodal single speaker joint VAD and SSL approach (BLAUTH; MINOTTO et al., 2012), which chronologically evolved to a multimodal approach (MINOTTO et al., 2013), and later to a multiple speaker one (Chapter 5). Additionally, we have also presented a GPU implementation of the SRP-PHAT algorithm (MINOTTO et al., 2012), given the requirement for real-time processing of HCI systems.

Our techniques focused on realistic environments, which were exposed to high levels of noise. All the presented experiments were performed in a very active laboratory, with people talking in background, entering/leaving the room, other computers functioning, etc. The cases where simultaneous speech was produced were also naturally generated, opposed some works that approach it using simulated data. For this reason, as a contribution of our work, the multimodal datasets used in all experiments were made available online (Sections 3.4, 4.4 and 5.4).

The next step to further improve our simultaneous speaker multimodal approach is to include other modalities of data. Experiments using information from a RGB-D camera (namely, the Kinect sensor) have already been conducted. We have been applying a feature fusion approach with an SVM classifier and already achieved VAD accuracies above 95%. The depth information allows 3D tracking of facial features through Active Shape Models, thus providing robust visual information for the fusion approach, and also benefits the SSL part of the algorithm, enabling a better 3D localization of the speakers. Nevertheless, this method is still undergoing improvements, and requires a feature selection step as future work.

REFERENCES

ALMAJAI, I.; MILNER, B. Using audio-visual features for robust voice activity detection in clean and noisy speech. In: EUROPEAN SIGNAL PROCESSING CONFERENCE (EUSIPCO 2008), 16. **Proceedings. . .** [S.l.: s.n.], 2008.

ASOH, H. et al. An Application of a Particle Filter to Bayesian Multiple Sound Source Tracking with Audio and Video Information Fusion. In: PROC. INT. CONF. ON INFORMATION FUSION (IF). **Proceedings. . .** [S.l.: s.n.], 2004. p.805–812.

ATREY, P. K. et al. Multimodal Fusion for Multimedia Analysis: a survey. **Multimedia Systems**, [S.l.], v.16, n.6, p.345–379, 2010.

AUBREY, A. et al. Two Novel Visual Voice Activity Detectors based on Appearance Models and Retinal Filtering. In: EUROPEAN SIGNAL PROCESSING CONFENRENCE. **Proceedings. . .** [S.l.: s.n.], 2007. v.1, p.2409–2413.

AUBREY, A.; HICKS, Y.; CHAMBERS, J. Visual voice activity detection with optical flow. **Image Processing, IET**, [S.l.], v.4, n.6, p.463 –472, december 2010.

BENESTY, J.; CHEN, J.; HUANG, Y. **Microphone Array Signal Processing**. [S.l.]: Springer, 2008. (Springer Topics in Signal Processing Series).

BERTRAND, A.; MOONEN, M. Energy-based multi-speaker voice activity detection with an ad hoc microphone array. In: ACOUSTICS SPEECH AND SIGNAL PROCESSING (ICASSP), 2010 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings. . .** [S.l.: s.n.], 2010. p.85 –88.

BINS, J. et al. Feature-Based Face Tracking for Videoconferencing Applications. In: MULTIMEDIA, 2009. ISM '09. 11TH IEEE INTERNATIONAL SYMPOSIUM ON. **Proceedings. . .** [S.l.: s.n.], 2009. p.227 –234.

BLAUTH, D. A. et al. Voice activity detection and speaker localization using audiovisual cues. **Pattern Recognition Letters**, [S.l.], v.33, n.4, p.373 – 380, 2012.

BOUGUET, J.-Y. **Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm**. 2000.

BRANDSTEIN, M.; ADCOCK, J.; SILVERMAN, H. A practical time-delay estimator for localizing speech sources with a microphone array. **Computer Speech and Language**, [S.l.], v.9, p.153–169, 1995.

BRANDSTEIN, M. S.; ADCOCK, J. E.; SILVERMAN, H. F. A Closed-Form Location Estimator for use with Room Environment Microphone Arrays. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.5, p.45–50, 01/1997 1997.

BRANDSTEIN, M.; WARD, D. **Microphone arrays**: signal processing techniques and applications. [S.l.]: Springer, 2001. (Digital signal processing).

BRUTTI, A.; OMOLOGO, M.; SVAIZER, P. Localization of multiple speakers based on a two step acoustic map analysis. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2008. ICASSP 2008. IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2008. p.4349 –4352.

BUTKO, T. et al. Fusion of audio and video modalities for detection of acoustic events. In: INTERSPEECH. **Proceedings...** [S.l.: s.n.], 2008. p.123–126.

BYRD, R. H.; SCHNABEL, R. B.; SHULTZ, G. A. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. **Mathematical Programming**, [S.l.], v.40, p.247–263, 1988.

CAI, W.; ZHAO, X.; WU, Z. Localization of Multiple Speech Sources Based on Subband Steered Response Power. In: ELECTRICAL AND CONTROL ENGINEERING (ICECE), 2010 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. p.1246 –1249.

CHANG, C.-C.; LIN, C.-J. LIBSVM: a library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, [S.l.], v.2, p.27:1–27:27, 2011.

CHAPMAN, B.; JOST, G.; PAS, R. v. d. **Using OpenMP**: portable shared memory parallel programming (scientific and engineering computation). [S.l.]: The MIT Press, 2007.

CHU, P. Desktop Mic Array for Teleconferencing. In: ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1995. ICASSP-95., 1995 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 1995. v.5, p.2999–3002 vol.5.

CONN, A.; GOULD, N.; TOINT, P. **Trust Region Methods**. [S.l.]: Society for Industrial and Applied Mathematics, 2000. (MPS-SIAM Series on Optimization).

CORPORATION, N. **NVIDIA GeForce 8800 GPU Architecture Overview**. [S.l.]: NVIDIA Corporation, 2006.

CORPORATION, N. **NVIDIA's Next Generation CUDA Compute Architecture**: Fermi. [S.l.]: NVIDIA Corporation, 2011.

CUDA Best Practices Guide. [S.l.]: NVIDIA Corporation, 2011. http://developer.nvidia.com/cuda-downloads.

CUDA CUFFT Library. [S.l.]: NVIDIA Corporation, 2011. http://developer.nvidia.com/cuda-downloads.

CUDA Programming Guide. [S.l.]: NVIDIA Corporation, 2011. http://developer.nvidia.com/cuda-downloads.

DEY, P.; SELVARAJ, M.; LEE, B. Robust User Context Analysis for Multimodal Interfaces. In: IN PROC. 13TH INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI), Alicante, Spain. **Proceedings...** [S.l.: s.n.], 2011. p.81–88.

DIBIASE, J. H. **A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays**. 2000. Tese (Doutorado em Ciência da Computação) — BROWN UNIVERSITY.

DO, H.; SILVERMAN, H. SRP-PHAT methods of locating simultaneous multiple talkers using a frame of microphone array data. In: ACOUSTICS SPEECH AND SIGNAL PROCESSING (ICASSP), 2010 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. p.125 –128.

DO, H.; SILVERMAN, H. F. A Fast Microphone Array SRP-PHAT Source Location Implementation using Coarse-to-Fine Region Contraction(CFRC). In: APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS, 2007 IEEE WORKSHOP ON. **Proceedings...** [S.l.: s.n.], 2007. p.295 –298. asdfasdf.

DO, H.; SILVERMAN, H. F. A method for locating multiple sources from a frame of a large-aperture microphone array data without tracking. In: ICASSP. **Proceedings...** IEEE, 2008. p.301–304.

DO, H.; SILVERMAN, H.; YU, Y. A Real-Time SRP-PHAT Source Location Implementation using Stochastic Region Contraction(SRC) on a Large-Aperture Microphone Array. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2007. ICASSP 2007. IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2007. v.1, p.I–121 –I–124.

EPHRAIM, Y. Statistical-model-based speech enhancement systems. **Proceedings of the IEEE**, [S.l.], v.80, n.10, p.1526–1555, 1992.

EPHRAIM, Y.; MALAH, D. Speech Enhancement Using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator. , [S.l.], v.33, n.2, p.443–445, 1985.

FAN, R.-E.; CHEN, P.-H.; LIN, C.-J. Working Set Selection Using Second Order Information for Training Support Vector Machines. **J. Mach. Learn. Res.**, [S.l.], v.6, p.1889–1918, Dec. 2005.

FARKAS, L. G. (Ed.). **Anthropometry of the Head and Face**. [S.l.]: Raven Press, 1994. 344–505p. v.6, n.4.

FERMI Compatibility Guide. [S.l.]: NVIDIA Corporation, 2011. `http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/Fermi_Compatibility_Guide.pdf`. Accessed Apr. 2012.

FIGUEIREDO, M.; JAIN, A. Unsupervised learning of finite mixture models. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.24, n.3, p.381 – 396, Mar. 2002.

FREUND, Y. The alternating decision tree learning algorithm. In: IN MACHINE LEARNING: PROCEEDINGS OF THE SIXTEENTH INTERNATIONAL CONFERENCE. **Proceedings...** Morgan Kaufmann, 1999. p.124–133.

FREUND, Y.; SCHAPIRE, R. E. Large Margin Classification Using the Perceptron Algorithm. In: MACHINE LEARNING. **Proceedings. . .** [S.l.: s.n.], 1998. p.277–296.

GAMA, J. a. Functional Trees. **Mach. Learn.**, Hingham, MA, USA, v.55, n.3, p.219–250, June 2004.

GATICA-PEREZ, D. et al. Audiovisual Probabilistic Tracking of Multiple Speakers in Meetings. **Audio, Speech, and Language Processing, IEEE Transactions on**, [S.l.], v.15, n.2, p.601–616, 2007.

GAZOR, S.; ZHANG, W. A soft voice activity detector based on a Laplacian-Gaussian model. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.11, n.5, p.498 – 505, sept. 2003.

GLASKOWSKY, P. N. **NVIDIA's Fermi**: The First Complete GPU Computing Architecture. [S.l.]: NVIDIA Corporation, 2009.

GOVINDARAJU, N. K. et al. High performance discrete Fourier transforms on graphics processors. In: ACM/IEEE CONFERENCE ON SUPERCOMPUTING, 2008., Piscataway, NJ, USA. **Proceedings. . .** IEEE Press, 2008. p.2:1–2:12. (SC '08).

GURBAN, M.; THIRAN, J. Multimodal Speaker Localization in a Probabilistic Framework. In: EUROPEAN SIGNAL PROCESSING CONFERENCE (EUSIPCO), FLORENCE, ITALY, SEPTEMBER 2006, 14. **Proceedings. . .** IEEE, 2006. (Parallel Computing in Electrical Engineering).

HALL, M. et al. The WEKA data mining software: an update. **SIGKDD Explor. Newsl.**, New York, NY, USA, v.11, n.1, p.10–18, Nov. 2009.

HALL, M.; FRANK, E. Combining Naive Bayes and Decision Tables. In: FLORIDA ARTIFICIAL INTELLIGENCE SOCIETY CONFERENCE (FLAIRS), 21. **Proceedings. . .** AAAI press, 2008. p.318–319.

HARRIS, M. **Optimizing Parallel Reduction in CUDA**. [S.l.]: NVIDIA Corporation, 2011.

HOLTE, R. C. Very simple classification rules perform well on most commonly used datasets. In: MACHINE LEARNING. **Proceedings. . .** [S.l.: s.n.], 1993. p.63–91.

HORDLEY, S. D. et al. Illuminant and device invariant colour using histogram equalisation. **Pattern Recognition**, [S.l.], v.38, p.2005, 2005.

HUBER, P. **Robust Statistics**. [S.l.]: Wiley, 2005. (Wiley Series in Probability and Statistics).

HUBER, P. J. Robust estimation of a location parameter. **Annals of Mathematical Statistics**, [S.l.], v.35, n.1, p.73–101, Mar. 1964.

HUGHES, T. et al. Using a real-time, tracking microphone array as input to an HMM speech recognizer. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998. PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings. . .** [S.l.: s.n.], 1998. v.1, p.249–252 vol.1.

HUGHES, T. et al. Performance of an HMM speech recognizer using a real-time tracking microphone array as input. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.7, n.3, p.346–349, May 1999.

JAIMES, A.; SEBE, N. Multimodal Human-Computer Interaction: a survey. **Computer Vision and Image Understanding**, [S.l.], v.108, n.1-2, p.116–134, October 2007.

JOHNSON, D. H.; DUDGEON, D. E. **Array signal processing** : concepts and techniques / don h. johnson, dan e. dudgeon. [S.l.]: P T R Prentice Hall, Englewood Cliffs, NJ :, 1993. xiii, 533 p. :p.

JOHNSON, D. H.; DUDGEON, D. E. **Array Signal Processing - Concepts and Techniques**. [S.l.]: Prentice, 1993.

KELLERMANN, W. A self-steering digital microphone array. In: ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1991. ICASSP-91., 1991 INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 1991. p.3581–3584 vol.5.

KIRK, D. B.; HWU, W.-m. **Programming Massively Parallel Processors: a hands-on approach**. [S.l.]: Morgan Kaufmann, 2010.

KNAPP, C.; CARTER, G. The generalized correlation method for estimation of time delay. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, [S.l.], v.24, n.4, p.320 – 327, Aug. 1976.

KOHAVI, R. The Power of Decision Tables. In: EUROPEAN CONFERENCE ON MACHINE LEARNING. **Proceedings...** Springer Verlag, 1995. p.174–189.

KWAK, K.-C.; KIM, S.-S. Sound source localization with the aid of excitation source information in home robot environments. **Consumer Electronics, IEEE Transactions on**, [S.l.], v.54, n.2, p.852 –856, may 2008.

LEE, B.; HASEGAWA-JOHNSON, M. Estimation of High-Variance Vehicular Noise. In: TAKEDA, K. et al. (Ed.). **In-Vehicle Corpus and Signal Processing for Driver Behavior**. [S.l.]: Springer US, 2009. p.221–232.

LEE, B.; KALKER, T. A Vectorized Method for Computationally Eficient SRP-PHAT Sound Source Localization. **12th International Workshop on Acoustic Echo and Noise Control**, [S.l.], August - September 2010.

LEE, B.; KALKER, T.; SCHAFER, R. W. Maximum-Likelihood Sound Source Localization With A Multivariate Complex Laplacian Distribution. In: **Proceedings...** [S.l.: s.n.], 2008.

LEE, B.; MUHKERJEE, D. Spectral Entropy-Based Voice Activity Detector for Videoconfencing Systems. In: INTERSPEECH, Makuhari, Japan. **Proceedings...** [S.l.: s.n.], 2010.

LEE, B.; MUHKERJEE, D. Spectral entropy-based voice activity detector for videoconferencing systems. In: INTERSPEECH. **Proceedings...** ISCA, 2010. p.3106–3109.

LINDHOLM, E. et al. NVIDIA Tesla: a unified graphics and computing architecture. **IEEE Micro**, Los Alamitos, CA, USA, v.28, p.39–55, 2008.

LOPES, C. et al. Color-based lips extraction applied to voice activity detection. In: IMAGE PROCESSING (ICIP), 2011 18TH IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2011. p.1057 –1060.

LORENZO-TRUEBA, J.; HAMADA, N. Noise robust Voice Activity Detection for multiple speakers. In: INTELLIGENT SIGNAL PROCESSING AND COMMUNICATION SYSTEMS, 2010 INTERNATIONAL SYMPOSIUM ON. **Proceedings...** [S.l.: s.n.], 2010. p.1 –4.

LUCAS, B. D.; KANADE, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In: **Proceedings...** [S.l.: s.n.], 1981. p.674–679.

M. AOKI K. MASUDA, H. M. T. T.; ARIKI, Y. Voice activity detection by lip shape tracking using EBGM. **Proceedings of the 15th international conference on Multimedia**, Augsburg, Germany, p.561–564, 2007.

MARABOINA, S. et al. Multi-speaker voice activity detection using ICA and beampattern analysis. In: EUROPEAN SIGNAL PROCESSING CONFERENCE EUSIPCO. **Proceedings...** [S.l.: s.n.], 2006. n.Eusipco, p.2–6.

MINOTTO, V. et al. GPU-based Approaches for Real-Time Sound Source Localization using the SRP-PHAT Algorithm. **International Journal of High Performance Computing Applications**, [S.l.], 2012.

MINOTTO, V. et al. Audiovisual Voice Activity Detection Based on Microphone Arrays and Color Information. **Selected Topics in Signal Processing, IEEE Journal of**, [S.l.], v.7, n.1, p.147–156, 2013.

MOHSEN NAQVI, S. et al. Multimodal (audiovisual) source separation exploiting multi-speaker tracking, robust beamforming and time-frequency masking. **Signal Processing, IET**, [S.l.], v.6, n.5, p.466 –477, july 2012.

NEMER, E.; GOUBRAN, R.; MAHMOUD, S. Robust voice activity detection using higher-order statistics in the LPC residual domain. **IEEE Transactions on Speech and Audio Processing**, [S.l.], v.9, n.3, p.217–231, March 2005.

OMOLOGO, M.; SVAIZER, P. Use of the crosspower-spectrum phase in acoustic event location. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.5, n.3, p.288 –292, may 1997.

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **Systems, Man and Cybernetics, IEEE Transactions on**, [S.l.], v.9, n.1, p.62 –66, 1979.

PEREZ, P.; VERMAAK, J.; BLAKE, A. Data fusion for visual tracking with particles. **Proceedings of the IEEE**, [S.l.], v.92, n.3, p.495–513, 2004.

PETSATODIS, T.; PNEVMATIKAKIS, A.; BOUKIS, C. Voice activity detection using audio-visual information. In: DIGITAL SIGNAL PROCESSING, 16., Piscataway, NJ, USA. **Proceedings...** IEEE Press, 2009. p.216–220. (DSP'09).

PLATT, J. C. Advances in kernel methods. In: SCHöLKOPF, B.; BURGES, C. J. C.; SMOLA, A. J. (Ed.). . Cambridge, MA, USA: MIT Press, 1999. p.185–208.

QUINLAN, J. R. **C4.5**: programs for machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

RABINER, L. A tutorial on hidden Markov models and selected applications in speech recognition. **Proceedings of the IEEE**, [S.l.], v.77, n.2, p.257 –286, feb 1989.

RABINER, L.; SCHAFER, R. **Digital processing of speech signals**. [S.l.]: Prentice-Hall, 1978. (Prentice-Hall signal processing series).

RAMIREZ, J. et al. Statistical voice activity detection using a multiple observation likelihood ratio test. **Signal Processing Letters, IEEE**, [S.l.], v.12, n.10, p.689–692, 2005.

RAMIREZ, J. et al. Improved Voice Activity Detection Using Contextual Multiple Hypothesis Testing for Robust Speech Recognition. **Audio, Speech, and Language Processing, IEEE Transactions on**, [S.l.], v.15, n.8, p.2177–2189, 2007.

ROBIN, T. et al. Specification, estimation and validation of a pedestrian walking behavior model. **Transportation Research Part B: Methodological**, [S.l.], v.43, n.1, p.36 – 56, 2009.

ROHANI, R. et al. Lip segmentation in color images. In: INNOVATIONS IN INFORMATION TECHNOLOGY, 2008. IIT 2008. INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2008. p.747 –750.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: a modern approach**. [S.l.]: Pearson Education, 2003.

SAVIOJA LAURI; VÃd’LIMÃd’KI, V. S. J. O. Audio Signal Processing Using Graphics Processing Units. **J. Audio Eng. Soc**, [S.l.], v.59, n.1/2, p.3–19, 2011.

SCHAPIRE, R. E.; FREUND, Y. Boosting the margin: a new explanation for the effectiveness of voting methods. **The Annals of Statistics**, [S.l.], v.26, p.322–330, 1998.

SCOTT, D. et al. Video Based VAD Using Adaptive Color Information. In: IEEE INT. SYMP. MULTIMEDIA ISM ’09, 11. **Proceedings...** [S.l.: s.n.], 2009. p.80–87.

SHALEV-SHWARTZ, S.; SINGER, Y.; SREBRO, N. Pegasos: primal estimated subgradient solver for svm. In: INTERNATIONAL CONFERENCE ON MACHINE-LEARNING, 24. **Proceedings...** [S.l.: s.n.], 2007. p.807–814.

SHI, J.; TOMASI, C. Good features to track. In: COMPUTER VISION AND PATTERN RECOGNITION, 1994. PROCEEDINGS CVPR &#039;94., 1994 IEEE COMPUTER SOCIETY CONFERENCE ON. **Proceedings...** IEEE, 1994. p.593–600.

SILVEIRA, L. G. da et al. A GPU Implementation of the SRP-PHAT Sound Source Localization Algorithm. **The 12th International Workshop on Acoustic Echo and Noise Control**, [S.l.], September 2010.

SILVERMAN, H. et al. Performance of real-time source-location estimators for a large-aperture microphone array. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.13, n.4, p.593 – 606, july 2005.

SILVERMAN, H.; PATTERSON W.R., I.; FLANAGAN, J. The Huge Microphone Array. 2. **Concurrency, IEEE**, [S.l.], v.7, n.1, p.32–47, Jan-Mar 1999.

SMALL, C. G.; WANG, J. **Numerical Methods for Nonlinear Estimating Equations**. [S.l.]: Oxford University Press, 2003.

SODOYER, D. et al. An Analysis of Visual Speech Information Applied to Voice Activity Detection. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING. **Proceedings...** [S.l.: s.n.], 2006. v.1, p.I–I.

SODOYER, D. et al. A study of lip movements during spontaneous dialog and its application to voice activity detection. **Journal of the Acoustical Society of America**, [S.l.], v.125, n.2, p.1184–1196, February 2009.

SOHN, J. et al. A Statistical Model-Based Voice Activity Detection. **IEEE Signal Process. Lett**, [S.l.], v.6, p.1–3, 1999.

SOHN, J.; SUNG, W. A voice activity detector employing soft decision based noise spectrum adaptation. **Proc. Int. Conf. Acoust., Speech, and Sig. Process.**, [S.l.], p.365–368, 1998.

SOON, I. Y.; KOH, S. N.; YEO, C. K. Improved noise suppression filter using self-adaptive estimator of probability of speech absence. **Sig. Process.**, [S.l.], v.75, n.2, p.151–159, 1999.

TAGHIZADEH, M. et al. An integrated framework for multi-channel multi-source localization and voice activity detection. In: HANDS-FREE SPEECH COMMUNICATION AND MICROPHONE ARRAYS, 2011 JOINT WORKSHOP ON. **Proceedings...** [S.l.: s.n.], 2011. p.92 –97.

TAKEUCHI S., H. T. T. S.; HAYAMIZU, S. **Voice activity detection based on fusion of audio and visual information**. 2009.

TALANTZIS, F.; PNEVMATIKAKIS, A.; CONSTANTINIDES, A. Audio-Visual Active Speaker Tracking in Cluttered Indoors Environments. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, [S.l.], v.38, n.3, p.799–807, 2008.

TALANTZIS, F.; PNEVMATIKAKIS, A.; CONSTANTINIDES, A. Audio-Visual Active Speaker Tracking in Cluttered Indoors Environments. **IEEE Transactions on Systems, Men and Cybernetics - Part B**, [S.l.], v.39, n.1, p.7–15, February 2009.

TANYER, S.; OZER, H. Voice activity detection in nonstationary noise. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v.8, n.4, p.478 –482, jul 2000.

TERVO, S.; LOKKI, T. Interpolation Methods for the SRP-PHAT Algorithm. **The 11th International Workshop on Acoustic Echo and Noise Control**, [S.l.], September 2008.

THIRAN, J.-P.; MARQUÉS, F.; BOURLARD, H. (Ed.). **Multimodal Signal Processing, Theory and Applications for Human-Computer Interaction**. [S.l.]: Academic Press, 2010.

TIAWONGSOMBAT, P. et al. Robust visual speakingness detection using bi-level HMM. **Pattern Recogn.**, New York, NY, USA, v.45, n.2, p.783–793, Feb. 2012.

TSINGOS, N. Using Programmable Graphics Hardware for Acoustics and Audio Rendering. **Audio Engineering Society Convention 127**, [S.l.], v.59, n.9, p.1–16, 2009.

TUNG, T. et al. Source localization and spatial filtering using wideband MUSIC and maximum power beamforming for multimedia applications. In: SIGNAL PROCESSING SYSTEMS, 1999. SIPS 99. 1999 IEEE WORKSHOP ON. **Proceedings...** [S.l.: s.n.], 1999. p.625 –634.

VERMAAK, J. et al. Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking. In: PROCEEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.: s.n.], 2001. p.741–747.

WANG, C.; BRANDSTEIN, M. S. A Hybrid Real-Time Face Tracking System. In: ICASSP98. **Proceedings...** [S.l.: s.n.], 1997. p.3737–3740.

WANG, C.; GRIEBEL, S.; BRANDSTEIN, M. Robust automatic video-conferencing with multiple cameras and microphones. In: MULTIMEDIA AND EXPO, 2000. ICME 2000. 2000 IEEE INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2000. v.3, p.1585–1588 vol.3.

WANG, H.; CHU, P. Voice Source Localization for Automatic Camera Pointing System in Videoconferencing. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING - VOLUME 1, 1997., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 1997. p.187.

WANG, L.; WANG, X.; XU, J. Lip Detection and Tracking Using Variance Based Haar-Like Features and Kalman filter. In: FRONTIER OF COMPUTER SCIENCE AND TECHNOLOGY (FCST), 2010 FIFTH INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. p.608 –612.

WEBB, A. R. **Statistical Pattern Recognition, 2nd Edition**. [S.l.]: John Wiley & Sons, 2002.

WEINSTEIN, E. et al. **LOUD**: a 1020-node modular microphone array and acoustic beamformer. [S.l.]: MIT Computer Science and Artificial Intelligence Laboratory, 2004.

WÖLFEL, M.; MCDONOUGH, J. **Distant Speech Recognition**. 1..ed. Chichester, UK: Wiley, 2009.

WU, T.-F.; LIN, C.-J.; WENG, R. C. Probability Estimates for Multi-class Classification by Pairwise Coupling. **J. Mach. Learn. Res.**, [S.l.], v.5, p.975–1005, Dec. 2004.

YAMAMOTO, S. et al. Real-Time Robot Audition System That Recognizes Simultaneous Speech in The Real World. In: INTELLIGENT ROBOTS AND SYSTEMS, 2006 IEEE/RSJ INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2006. p.5333 –5338.

YANG, M.; AHUJA, N. Gaussian Mixture Model for Human Skin Color and Its Applications in Image and Video Databases. In: ITS APPLICATION IN IMAGE AND VIDEO DATABASES.Ť PROCEEDINGS OF SPIE Š99 (SAN JOSE CA. **Proceedings...** [S.l.: s.n.], 1999. p.458–466.

YAO, H.; GAO, W. Face detection and location based on skin chrominance and lip chrominance transformation from color images. **Pattern Recognition**, [S.l.], v.34, n.8, p.1555 – 1564, 2001.

ZHANG, C. et al. Boosting-Based Multimodal Speaker Detection for Distributed Meeting Videos. **Multimedia, IEEE Transactions on**, [S.l.], v.10, n.8, p.1541–1552, 2008.

ZHANG, C.; ZHANG, Z.; FLORENCIO, D. Maximum Likelihood Sound Source Localization for Multiple Directional Microphones. In: **Proceedings...** [S.l.: s.n.], 2007. v.1, p.I–125 –I–128.

ZHANG, W.; RAO, B. A Two Microphone-Based Approach for Source Localization of Multiple Speech Sources. **Audio, Speech, and Language Processing, IEEE Transactions on**, [S.l.], v.18, n.8, p.1913 –1928, nov. 2010.

ZHANG, Y.; COHEN, J.; OWENS, J. D. Fast Tridiagonal Solvers on the GPU. In: ACM SIGPLAN SYMPOSIUM ON PRINCIPLES AND PRACTICE OF PARALLEL PROGRAMMING (PPOPP 2010), 15. **Proceedings...** [S.l.: s.n.], 2010. p.127–136.