ROYAL INSTITUTE OF TECHNOLOGY (KTH)

SCHOOL OF ELECTRICAL ENGINEERING

Speaker Localization Techniques in Reverberant Acoustic Environments

MASTER OF SCIENCE THESIS AT THE DEPARTMENT OF ELECTRICAL ENGINEERING

AUTHOR: Carlos Castro Gonzalez

Stockholm, September 2007

Master Thesis

DESIGN, IMPLEMENTATION, EVALUATION AND IMPROVEMENT OF SPEAKER LOCALIZATION AND TRACKING TECHNIQUES IN REVERBERANT ACOUSTIC ENVIRONMENTS

Author:

Carlos Castro Gonzalez

Carried out at:

SPEECH TECHNOLOGY GROUP (GTH)
DEPARTMENT OF ELECTRONICS ENGINEERING (DIE)
TECHNICAL SCHOOL OF TELECOMMUNICATION ENGINEERING (ETSIT)
TECHNICAL UNIVERSITY OF MADRID (UPM)

External Advisor:

Javier Macias Guarasa.
Professor at the Department of Electrical Engineering.
Technical University of Madrid (UPM), Spain.

Examiner:

Bastiaan Kleijn.
Professor at the Department of Electrical Engineering.
Royal Institute of Technology (KTH), Stockholm, Sweden.

Stockholm, September 2007

To my parents

ABSTRACT:

Nowadays, the use of speech technology to operate with electronic and informatic systems is getting more and more frequent and its usefulness indisputable. Given that speech is the most natural way in which humans interact with their environment, there is a clear tendency to make use of this technology as the mean to communicate with devices.

Automatic speech recognition systems are already able to properly understand reasonably complex human commands. Nevertheless, there are certain accoustic conditions under which the error rates commited by them are still too high. Particularly, the capture of distant speech in reverberant accoustic environments is specially conflictive for this type of systems which usually show lower performances than expected. At the same time, this distant speech acquisition is of high importance since it allows a more natural way to interact, without neccesarily having to carry intrusive, close-talk microphones, and it is present in multiple and common situations, such as those given in a digital home or a conference room.

This present Master Thesis intends to design, implement and evaluate from scratch an accurate and complete tool to estimate the speaker localization in these reverberant accoustic environments. This localization application is crucial to improve the results of the mentioned recognition systems since it can be used to exploit the spatial filtering ability of an array, which allows the speech signal from one talker to be enhanced as the signals from other talkers as well as undesired sources and noise are supressed.

After a detailed theoretical study, a robust localization system was implemented based on the Steered Response Power (SRP) of an array when focused at different locations. In turn, this scheme lies itself on a more basic localization algorithm able to estimate the Direction of Arrival (DOA) of the given speech by computing the Generalized Cross Correlation (GCC) between microphone pairs. Finally, an exhaustive evaluation of the results obtained by the system was carried out in order to check the validity of its outcomes, hint the possible improvement techniques, get some general conclusions about its performance and suggest future lines of investigation.

KEY WORDS:

Automatic speech recognition, speaker localization, Generalized Cross Correlation (GCC), Steered Response Power (SRP), beamforming, spatial filtering, microphone array.

RESUMEN:

Actualmente, cada vez es más frecuente el uso de la tecnología del habla en el manejo de sistemas electrónicos e informáticos. Existe una clara tendencia a usar esta tecnología como el principal medio de comunicación con dispositivos, dado que el habla es la forma más natural que poseen los humanos para interactuar con su entorno.

Los sistemas de reconocimiento automático de voz son ya capaces de comprender correctamente mandatos humanos razonablemente complejos. Sin embargo, existen ciertas condiciones acústicas bajo las cuales las tasas de error cometidas por estos sistemas son aún demasiado elevadas. En concreto, la captura de habla lejana en entornos acústicos reverberantes es especialmente conflictiva para estas técnicas que a menudo presentan un rendimiento por debajo de lo deseado. Al mismo tiempo, esta adquisición de habla lejana es especialmente importante ya que permite una comunicación más natural, libre de molestos e intrusivos micrófonos de habla cercana, y está presente en múltiples y frecuentes situaciones como las que se dan en el interior de una casa domótica o en un aula de conferencias.

El presente Proyecto va a abordar el diseño, implementación y evaluación desde cero de una completa y precisa herramienta de estimación de la posición en estos entornos acústicos reverberantes. Esta aplicación será muy importante a la hora de mejorar los resultados de los mencionados sistemas de reconocimiento ya que puede ser usada para explotar la habilidad que los arrays de micrófonos tienen para filtrar espacialmente, centrándose en un determinado punto de manera que se realce el habla de un hablante en concreto mientras que el resto de señales de ruido y de otros hablantes son evitadas.

Tras un detallado estudio teórico, se ha implementado un sistema de localización robusto basado en el Método de la respuesta en potencia (SRP) de un array cuando éste se apunta a distintas localizaciones. A su vez, este esquema se basa en una técnica de localización más básica, capaz de estimar la dirección de llegada (DOA) del habla mediante el cálculo de la correlación cruzada generalizada (GCC) entre pares de micrófonos. Por último, se realizó una exhaustiva evaluación de los resultados obtenidos con este sistema de manera que se pudiera comprobar la validez de los mismos, dar alguna pista sobre las posibles técnicas de mejora, obtener ciertas conclusiones generales sobre su rendimiento y apuntar a futuras líneas de investigación que pudieran mejorar el trabajo realizado en este Proyecto.

Report

Contents

Figure Index **viii**

Table Index **x**

1 Introduction **2**

1.1 Presentation 2

1.2 Motivation 4

1.3 Structure 5

2 Theoretical review **7**

2.1 Introduction 7

2.2 Sound wave propagation 7

2.2.1 Effect of attenuation, noise and reverberation 8

2.2.2 Direct path propagation 9

2.2.3 The room impulse response and the multi-path model 9

2.2.4 Microphone signal model 11

2.2.5 Far-field vs. near-field 11

2.3 Microphone arrays 15

2.3.1 Spatial aliasing 18

2.3.2 Beamforming 19

2.3.2.1 Delay-and-sum beamformer 21

2.3.2.2 Filter-and-sum beamformer 21

2.3.2.3 Constant Directivity Beamforming (CDB) 22

2.4 Localization algorithms 23

2.4.1 Based on Time Delay Estimation (TDE) 25

2.4.1.1 The normalized Cross Correlation (CC) 26

2.4.1.2 The Generalized Cross Correlation (GCC) . . . 27
2.4.1.3 GCC-PHAT implementation . . . 29
2.4.1.4 A source location method based on TDOA . . . 32
2.4.2 Based on Spectral-Estimation-Based Sub-Spaces . . . 33
2.4.3 Based on Steered Response Power (SRP) . . . 33
2.4.3.1 The SRP-PHAT algorithm . . . 35
2.4.3.2 SRP in terms of GCC . . . 36
2.4.3.3 SRP-PHAT implementation . . . 37
2.4.4 Additional estrategies . . . 39
2.4.4.1 Coarse to fine search . . . 39
2.4.4.2 Signal to noise ratio considerations . . . 40
2.4.4.3 Estimation of localization confidence . . . 41
2.4.4.4 Interpolation techniques . . . 42
2.4.4.5 Filtering techniques . . . 44
2.4.4.6 Use of geometrical information . . . 44
2.5 Tracking algorithms . . . 45
2.5.1 The Kalman filter . . . 46
2.5.2 Particle Filtering (PF) . . . 47
2.6 Voice activity detection . . . 47
2.7 Summary . . . 48

3 Experimental results **52**
3.1 Introduction . . . 52
3.2 Evaluation strategy . . . 52
3.2.1 Main Evaluation Metrics: The CHIL Evaluation Plan . . . 52
3.2.2 Other Evaluation Metrics . . . 54
3.2.3 Sample Results Table . . . 54
3.2.4 Tunable parameters . . . 56
3.3 Databases . . . 61
3.3.1 IDIAP AV16.3 . . . 61
3.3.1.1 Geometry . . . 61
3.3.1.2 Contents . . . 61
3.3.1.3 Annotation . . . 63

3.3.2 HIFI-MM1 . . . 64
3.3.2.1 Geometry . . . 64
3.3.2.2 Contents . . . 64
3.3.2.3 Annotation . . . 66
3.3.3 Simulated SONY . . . 67
3.3.3.1 Geometry . . . 67
3.3.3.2 Contents . . . 68
3.3.3.3 Annotation . . . 68
3.3.4 Simulated HIFI . . . 69
3.4 Baseline results (GCC-PHAT) . . . 71
3.4.1 AV16.3 . . . 71
3.5 Basic algorithm characterization (SRP) . . . 73
3.5.1 Sampling frequency and interpolation techniques . . . 73
3.5.1.1 HIFI . . . 73
3.5.1.2 AV16.3 . . . 77
3.5.1.3 HIFI vs. AV16.3 . . . 79
3.5.2 Frame size . . . 80
3.5.2.1 AV16.3 . . . 80
3.5.2.2 HIFI . . . 86
3.5.3 FFT size . . . 88
3.5.3.1 AV16.3 . . . 89
3.5.4 Windowing . . . 90
3.5.4.1 AV16.3 . . . 90
3.5.4.2 HIFI . . . 93
3.5.5 Search space grid . . . 93
3.5.5.1 AV16.3 . . . 94
3.5.5.2 HIFI . . . 96
3.5.5.3 SONY . . . 103
3.5.6 Comparison between real and simulated data . . . 106
3.5.6.1 HIFI . . . 106
3.6 Array geometry evaluation . . . 108
3.6.1 Number of microphones . . . 108
3.6.1.1 Simulated HIFI . . . 109

3.6.1.2 Real HIFI . . . 115
3.6.2 Intermicrophone distance . . . 118
3.6.2.1 Simulated HIFI . . . 118
3.7 Speaker position influence . . . 120
3.7.1 Real HIFI . . . 120
3.8 Computational demands . . . 123
3.9 Evaluation of aditional strategies . . . 129
3.9.1 Coarse to fine strategy . . . 129
3.9.1.1 HIFI . . . 131
3.9.2 Noise masking . . . 133
3.9.2.1 Real HIFI . . . 134
3.9.3 Estimation of localization confidence . . . 137
3.9.3.1 Simulated HIFI . . . 138
3.9.4 Filtering techniques . . . 143
3.9.4.1 AV16.3 . . . 143
3.9.4.2 Real HIFI . . . 146
3.9.4.3 SONY . . . 150
3.9.5 Use of geometrical information . . . 152
3.9.5.1 AV16.3 . . . 152
3.10 Selected final experiments . . . 153
3.10.1 AV16.3 . . . 154
3.10.2 Real HIFI . . . 155
3.10.3 Simulated HIFI . . . 156

4 Conclusions **158**

5 Future work **163**
5.1 Introduction . . . 163
5.2 Implementation of the Time Delay Selection (TIDES) algorithm . . . 163
5.3 Improvement of the coarse to fine search method . . . 164
5.4 Further estimation of the localization confidence . . . 165
5.5 Use of better array configurations . . . 166
5.6 Algorithm optimization . . . 167

5.7 Tracking algorithms implementation 167
5.8 Multiple speaker localization 168

Bibliography **168**

A Speed of sound **174**

B Windowing **175**

C User manual **177**
C.1 Introduction 177
C.2 Steered Response Power (SRP) 177
C.2.1 Introduction 177
C.2.2 Command line options 177
C.2.3 Way of operation 178
C.2.4 Example 181

Figure Index

2.1 Instance of a room impulse response. Three parts can be distinguished: the direct wave, the early reflections and the late reflections 9
2.2 Sound propagation inside a room 10
2.3 Microphone array in far-field situation 13
2.4 Microphone array in near-field situation 14
2.5 Directivity pattern for varying number of sensors (f=1KHz, L=0.5m) . . . 17
2.6 Directivity pattern for varying array effective length (f=1KHz, N=5) 17
2.7 Directivity pattern for varying frequency (400Hz<f<3000Hz, N=5, d=0.1m) 18
2.8 Array directivity pattern withouth (a) and with (b) spatial aliasing 19
2.9 Unsteered and steered directivity patterns (ϕ'=45 degrees,f=1KHz,N=10,d=0.15m) 20
2.10 Frequency domain block diagram of a delay-and-sum beamformer 22
2.11 Frequency domain block diagram of a filter-and-sum beamformer 22
2.12 Geometry of a CDB array of 25 elements (adapted from [AM01]) 23
2.13 Directivity pattern of a CDB array as a function of frequency (adapted from [AM01]) 24
2.14 Zotkin and Duraiswami beamformer peak width as a function of frequency 40
2.15 Discrete sinusoid (left) and the magnitude of its DFT (right). (Taken from Varma **??**) 43
2.16 Interpolated discrete sinusoid (left) and the magnitude of its zero-padded DFT.(Taken from Varma **??**) 43
2.17 QIO block diagram 49

3.1 Sound speed as a function of temperature 59
3.2 Idiap Smart Meeting Room 62
3.3 3 m-long by 2 m-wide L-shaped area for speakers distribution in Idiap Room 62
3.4 The Edecan Project Room sited at the Speech Technology Group (GTH) in the Technical University of Madrid (UPM) 65
3.5 Microphone array configuration for Simulated HIFI and Simulated Sony databases. 33 equispaced linear array (circles) and 11 harmonic array (squares). d = 20 mm 68
3.6 Absolute error (in samples, top) and relative error (in %1, bottom). Performance comparison between GCC (crosses) and SRP (circles) 73
3.7 Absolute error (in samples, top) and relative error (%, bottom) histograms (boxes) and accumulated histograms (lines). Comparison between GCC (solid) and SRP (dashed) 74

3.8 Static AV16.3 average error and localization rate as a function of the frame size considered. Top: Average error fine+gross (solid), average error fine (dashed). Bottom: Pcor (solid), A-MOTA (dashed). 81
3.9 Static AV16.3 run time and FFT size as a function of the frame size 82
3.10 Moving AV16.3 average error and localization rate as a function of the frame size considered. Top: Average error fine+gross (solid) and average error fine (dashed). Bottom: Pcor (solid) and A-MOTA (dashed). Aggregated results for seq 11 and 15 are shown in thick lines, while seq15 alone is shown with thin lines. 85
3.11 Real HIFI average error and localization rate as a function of the frame size considered 87
3.12 Real HIFI. Search grid effect. Top: Average error fine+gross (solid) and average error fine (dashed) as function of the grid spacing. Bottom: Pcor (solid) and A-MOTA (dashed) as function of the grid spacing. 99
3.13 Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 500 Hz. 100
3.14 Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 1000 Hz. 100
3.15 Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 5000 Hz. 101
3.16 Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 10000 Hz. 101
3.17 Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 20000 Hz. 102
3.18 Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 1000 Hz. 103
3.19 Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 5000 Hz. 104
3.20 Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 10000 Hz. 104
3.21 Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 20000 Hz. 105
3.22 Simulated HIFI. Effective length and number of mics effect. 114
3.23 Experimental results (solid) and curve-fitting (dashed) of our array configuration peak width as a function of frequency 130

5.1 Instance of a Constant Directivity Beamformer array (left) and its correponding directivity pattern along all the frequency bands (right). An almost frequency-constant directivity pattern is obtained. 167

B.1 Commonly used windows (taken from J. Ordonez, [OV03]) 176

C.1 SRP program block diagram 178

Table Index

2.1 Far-field limits for a linear array of 33 equispaced 20 mm and a linear array of 4 elements equispaced 200 mm. 12

3.1 Instance of the standard results table used in this Master Thesis 55
3.2 List of the annotated sequences. Tags mean: [ST]Static speaker, [MV]Moving speaker 63
3.3 Main features of the different databases used in the experimentation . . . 70
3.4 Real HIFI. Interpolation techniques. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 150 mm 75
3.5 Real HFI. Rounding techniques. fs = 16 KHz. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 150 mm 76
3.6 Real HIFI. Rounding techniques II. fs = 48 KHz. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 250 mm 77
3.7 ST-AV16.3. Interpolation techniques II. Frame size: 640 ms. Grid step in which the room was divided: 50 mm 78
3.8 ST-AV16.3. Interpolation techniques. Frame size: 320 ms. Grid step in which the room was divided: 100 mm 78
3.9 Static AV16.3. Instance of computational loads for interpolation schemes . 79
3.10 ST-AV16.3. Frame size effect. Grid step in which the room was divided: 150 mm 80
3.11 MV-AV16.3. Frame size effect. Grid step in which the room was divided: 150 mm 83
3.12 MV-AV16.3 seq11. Frame size effect. Grid step in which the room was divided: 150 mm 83
3.13 MV-AV16.3 seq15. Frame size effect. Grid step in which the room was divided: 150 mm 84
3.14 Real HIFI. Frame size effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Grid step in which the room was divided: 150 mm 86
3.15 ST-AV16.3. FFT size effect. fs = 16 KHz. Frame Size = 512 ms equivalent to 8192 samples at fs = 16KHz 89
3.16 ST-AV16.3. Windows effect. fs = 16 KHz. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm 91

3.17 MV-AV16.3. Windows effect. fs = 16 KHz. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm 92
3.18 Real HIFI. Windows effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm 93
3.19 ST-AV16.3. Grid spacing effect. fs = 16 KHz. Frame Size= 640 ms 95
3.20 Real HIFI. Grid spacing effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms 97
3.21 HIFI. Maximum difference in degrees between adjacent points as a function of grid inter-spacing. 102
3.22 Simulated SONY. Grid spacing effect. fs = 48 KHz. Sequences considered= 8740 recordings from 20 different speakers placed at one single, static position (P2). Frame Size= 640 ms 105
3.23 Real HIFI vs. Simulated HIFI. Microphone array: 4 sennheiser microphones equispaced 200 mm. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms. Grid step in which the room was divided: 150 mm 107
3.24 Real HIFI vs. Simulated HIFI II. Microphone array: 4 sennheiser microphones equispaced 200 mm plus 3 L-shaped crown microphones. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms. Grid step in which the room was divided: 150 mm 108
3.25 Simulated HIFI. Array geometry effect I. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm . . 110
3.26 Simulated HIFI. Array geometry effect II. Microphone array: Linear array of 4 elements equispaced 200 mm : effective longitude, L= 800 mm. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 111
3.27 Simulated HIFI. Number of microphones effect with growing effective length. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 113
3.28 Real HIFI. senn vs. crown mics effect II. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm . 116
3.29 Real HIFI. senn vs. crown mics effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm . 117
3.30 Real HIFI. Array geometry effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm 118
3.31 Simulated HIFI. Intermic distance effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm . . 119

3.32 Real HIFI. Position effect. fs = 48 KHz. Sequences considered: Recordings from 12 different speakers placed at static positions P1 (44 recordings), P2 (43 recordings), P3 (45 recordings), P4 (45 recordings) and P5 (46 recordings). Frame Size= 320 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 150 mm 122
3.33 Real HIFI. SRP vs. FSRP computational load. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms : FFT Size= 16384 at fs = 48 KHz. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 250 mm : 429 locations 125
3.34 ST-AV16.3. Frame size computational load effect. Microphone array: Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm : 56867 locations 126
3.35 Simulated HIFI. Number of mics computational load effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: The physical longitude of all the arrays compared is 660 mm, their effective lenghts are different though. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm : 37180 locations 127
3.36 MV-AV16.3. Grid spacing computational load effect. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each 128
3.37 Real HIFI. Coarse to fine strategy. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 500 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 100 mm 131
3.38 Simulated HIFI. Fixed-threshold noise masking effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm 135
3.39 Simulated HIFI. Adaptive threshold noise masking effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm 136
3.40 Simulated HIFI. Microphone distance weighting effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 33 sennheiser microphones equispaced 20 mm (L = 660 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 139
3.41 Simulated HIFI. Microphone distance weighting effect II. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear harmonic array of 11 sennheiser microphones. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 140

3.42 Simulated HIFI. Microphone distance weighting effect IV. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 17 sennheiser microphones equispaced 20 mm (L = 340 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 141
3.43 Simulated HIFI. Microphone distance weighting effect III. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 5 sennheiser microphones equispaced 160 mm (L = 800 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 142
3.44 ST-AV16.3. Low-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm 144
3.45 ST-AV16.3. High-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm 144
3.46 ST-AV16.3. Band-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm 145
3.47 Real HIFI. Low-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm 147
3.48 Real HIFI. High-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm 148
3.49 Real HIFI. Band-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm 149
3.50 Simulated SONY. Low-pass filtering effect. fs = 48 KHz. Sequences considered: 8740 recordings from 20 different speakers placed at one single, static position (P2). Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm 151
3.51 ST-AV16.3. Use of geometrical information effect. fs = 16 KHz. Frame Size= 40 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm 152
3.52 ST-AV16.3. Dead areas effect. fs = 16 KHz. Frame Size= 40 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm 153

3.53 ST-AV16.3. Selected final experiment. fs = 16 KHz. Frame size= 500 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 100 mm. Windowing: Hamming. Band-pass filter: [4-8KHz]. Rounding applied. 155
3.54 Real HIFI. Selected final experiment. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame size= 320 ms. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm. Windowing: Hamming. Band-pass filter: [2-16KHz]. Rounding applied. 156
3.55 Simulated HIFI. Selected final experiment. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame size= 640 ms. Microphone array: Linear harmonic array of 11 sennheiser microphones. Grid step in which the room was divided: 50 mm. Windowing: Hamming. Microphone distance weigthing applied. Rounding applied. 157

Acronyms list

2D	2 Dimensions
3D	3 Dimensions
AE	Average Error
A-MOTA	Audiovisual Multiple Object Tracking Accuracy
CC	Cross Correlation
CDB	Constant Directivity Beamforming
CSP	Crosspower Spectrum Phase
DFT	Discrete Fourier Transform
DKF	Decentralized Kalman Filter
DOA	Direction of Arrival
EKF	Extended Kalman Filter
FSRP	Frequency-domain Steered Response Power
GCC	Generalised Cross-Correlation
EVD	Eigen Value Decomposition
FFT	Fast Fourier Transform
LMS	Least Mean Square
ML	Maximum Likelihood
MOTP	Multiple Object Tracking Precision
MUSIC	Multiple Signal Classification
NN	Neural Network
PF	Particle Filtering
PHAT	Phase Transform
QIO	Qualcomm-ICSI-OGI front-end
RMSE	Root Mean Square Error
SCOT	Smoothed Coherence Transform
SMC	Sequencial Monte-Carlo
SNR	Signal-to-Noise Ratio
SRP	Steered Response Power
TDE	Time Delay Estimation

TDOA	Time Difference of Arrival
TSRP	Time-domain Steered Response Power
UKF	Unscented Kalman Filter
VAD	Voice Activity Detector

Chapter 1

Introduction

This introductory chapter will offer both a general view of the objectives aimed to be achieved in this Master Thesis and a justification of the Thesis itself within the research environment and the international state of the art in which it was performed.

1.1 Presentation

Nowadays, speech technology use is becoming increasingly popular in all kind of environments. Speech is the natural human way of comunication and, hence, using it as the direct mechanism to interact with machines has become an attractive field of research in order to get more supportive and less burdensome computing and communication services [OM06]. More and more applications accesible to the final user are being developed: speech dialing in mobile phones, dictating programs, domotic applications, automatic recording of meetings, etc. Not only this is a more efficient way to solve some problems, but also, sometimes, the only possible alternative to do so: for instance, applications designed for deaf people or to be run in reduced devices that do not allow easy dialing.

Within this context, the automatic speech recognizing systems have improved largely in the last few years and are already able to perform reasonably complex tasks. However, it is important to point out that the performance of these speech recognizing systems is severely affected by the accoustic conditions of the environment in which the speech was captured. In particular, those applications using distant microphones to capture speech, for instance in meeting, car, home or lecture scenarios, suffer from high error rates since both the tough reverberant accoustic conditions and the low signal to noise ratio negatively affect to the performance ([MG02], [WK02]). Lots of efforts have been devoted in order to improve automatic speech recognition on these environments. Using close-talk microphones to capture the speech would offer a solution, however, a non-

intrusive mechanism would be desirable: the speaker could freely move without carrying a microphone and, still, the speech recognition system would equally work. The use of microphone arrays in this task ([Sel03], [Her05]) has been proposed as a way to improve speech quality captured by distant microphones.

This Master Thesis aims at getting an automatic speaker localization system with reasonable accuracy within reverberant accoustic environments. Properly localizing the place where the speaker talks allows to "point" the microphone array to that exact place by using beamforming, therefore getting a much cleaner speech signal, free of most of the noise and reverberations, that can be used to get better automatic speech recognition rates [RF07].

The objectives followed during this Master Thesis are:

- Design and implementation of speaker localization techniques in accoustic reverberant environments

  This Master Thesis is one of the first research works in localization techniques within the Speech Technology Group at Technical University of Madrid and it therefore aims not only at being fully compatible with the speech recognition systems developed within the Group but also at improving them so that they can fully operate within noisy, reverberant environments.

  The system developed must be flexible, that is to say, all its key parameters should be easily modified so that they can be quickly adapted to the different experiment conditions and situations. Moreover, the system should be designed in such a way that allows future extensions, improvements and additional localization techniques to be implemented, it must aim at being as robust and efficient as possible and it must tend to perform in real-time rates.

  Also, this Master Thesis must provide a complete description and documentation of the sytems involved and software developed so that it can be later on easily used.

- Evaluation of the developed algorithms

  After the conclusion of this Master Thesis, it must be possible to make some statements and conclusions about the validity of the localization algorithms implemented as well as the improvement techniques used. In order to do so an extensive experimentation corpus must be carried out using different databases and environments. Specifically, this evaluation must concentrate on measuring the localization improvement or worsening of the different techniques considered and the different accoustic conditions as well as how big their impact is in each case. The statistical reliability of the results obtained will also be taken into account in order to fairly consider the experiments performed.

Both in the design and implementation Section as well as in the evaluation one, a detailed validation of the results obtained will be done in order to compare it to the objectives exposed above and check wether they have been fulfilled in a satisfactory way or not and why.

Finally, it is important to point out that all the tools used to edit, develop, compile and debug the code written, along with all the graphic generators and text processors used during this Master Thesis writing belong to the free software movement [1]. Also some propietary programs such as Matlab were used in specific cases in order to check the validity of the results obtained by the open source code developed.

1.2 Motivation

As comented above, this Master Thesis constitutes one of the first research projects carried out within the Speech Technology Department (GTH) at the Technical University of Madrid (UPM) in the field of speech localization within reverberant environments. It is based in a previous, basic research work developed at the group, [MH06], in which some simple techniques about Direction of Arrival (DOA) determination had already been developed.

The Speech Technology Department (GTH) has at the moment reliable, high quality speech recognizing systems which are expected to work in future projects not only just with close-talk speech but also within highly reverberant environments such as a conference room or a digital home. Within this context, this current Master Thesis has been developed in close collaboration with the Thesis "Design, implementation and evaluation of techniques for speech signal improvement in reverberant environments: application in automatic speech recognition systems", [RF07], which has partly used the information about the speaker localization given by this Thesis in order to improve the performance of the automatic speech recognition systems working in reverberant environments.

The importance of this current Master Thesis within the research context in which it was carried out was therefore to be the first in implementing some advanced localization techniques able to make accurate estimates about the speaker position. Also, the laboratory at the Speech Technology Department was equiped with speech acquisition hardware during the development of this Master Thesis. This fact allowed this Master Thesis to be the first in the Group to succesfully provide localization estimates related to real life conditions recorded at its laboratory. What is more, an extensive experimentation

[1] More information about free software philosophy can be found here: www.fsf.org/philosophy/free-sw.html

was carried out, both under real and simulated databases, in order to check the validity of its results and suggest techniques and future lines to improve its performance. This Thesis is also intended to serve as basis to future research projects as it is thus provided with a complete documentation depicting its way of operation.

Regarding the international context, there is plenty of research activity currently being developed about the topic. The most relevant works about the issue has been taken as reference and the most common localization techniques depicted in the state-of-the-art have been implemented. The testing process has also been done according to international evaluation standards so that our results could be directly comparable to those of different research projects. At the same time, the algorithms implemented here have been tested not only with the database recorded at this research Group but also with some other public available databases used in different, international, renowned projects. Future research works about the topic could extend this Master Thesis and add as many new techniques and algorithms as it may appear in the close future.

1.3 Structure

This Master Thesis is organized in 4 chapters:

In the firt chapter (the present one), an introduction to the Master Thesis is done. This introduction offers an overview of the context in which it was developed, both in the research group where it was carried out, the Speech Technology Group (GTH) at the Technical University of Madrid (UPM), and in the international research scene about the topic. It also explains the aim of the Thesis as well as its justification and details a brief explanation about its contents distribution in order to achieve a more accesible, clearer reading proccess.

In the second chapter there is a general description about the state of the art in speaker localization techniques within reverberant environments. This chapter also provides the basic theoretical background about the algorithms taken into account.

In the third chapter, an extensive evaluation of the implemented algorithms is carried out. There is an exhautive description of the experiments performed: the databases used, the multiple strategies and improvements followed, the different conditions under which they were performed and the results and conclusions they led to.

The fourth chapter summarizes the conclusions we got to based on the experiments realized.

The fifth chapter introduces the future lines of research work within the speaker localization field.

Then, the bibliography used in this Master Thesis is detailed.

Finally, a serie of appendix show the implementation details of the different algorithms and tools developed as well as a documentation about them. Also, some issues regarding the speech frames windowing process as well as the speed of sound issues used in this Master Thesis are explained in detail.

Chapter 2

Theoretical review

2.1 Introduction

In this chapter we will introduce and explain the key aspects developed in this Master Thesis in order to achieve a proper speech recognizing technology in acoustic reverberant environments using microphone arrays. We will particularly concentrate on the conditions in these reverberant environments as well as on those techniques aimed at localizing and enhancing the speech signals captured on them.

Speech recorded in real environments by distant microphones is dramatically degraded by factors like noise and reverberation. In the case of applications relaying in close-talk microphones, the influence of these two factors is relatively low which permits the development of succesful human to machine communication systems. But whenever signals are collected in real environments where the microphones are located further away from the sources of interest, the influence of these degradations dramatically degrade the performance of the developed systems. The alternatives for the design of a robust localization system in reverberant rooms are depicted in the following Sections.

2.2 Sound wave propagation

Throughout this Thesis, sound will be assumed to propagate in spherical waves according to the solution derived from the linear wave equation in [AM01]. More realistic radiation patterns of the human head have been described in [MS94]. However, the application of these complicated models is beyond the scope of this Thesis.

With this framework, we will now deploy a model describing how the speech propagation affects the signal received at a microphone array assuming the realistic acoustic

conditions given in a small room environment. We have to take into account three main factors affecting the speech signal propagation: attenuation, noise and reverberation.

2.2.1 Effect of attenuation, noise and reverberation

When propagating in spherical wavefronts, the signal amplitude decays at a rate proportional to the distance travelled from the source as described in [AM01]. This produces a fall in the signal SNR ratio and may even lead the speech power to drop behind the level of the noise if the capturing microphone is placed too far-away.

Apart from attenuation, in any propagation environment there is always the addition of some acoustic noise. As explained in [AG07] pp. 9-10, acoustic noise refers to the overall undesired sound events, that is, any accumulation of external disturbances to the target information received at a microphone. For instance, in the case of automatic speech recognition in a conference room, everything but the target speaker will be considered noise althought their features can significally diverge. In general, we can grossly classify noises in two kinds: First, the *non-directional* ones, mainly refering to the background noise that is often considered to be spatially white. As referred in [PSO97], this kind of noise reduces the SNR but, instead, does not significally bias the direct-path component of the dominant accoustic source. On the contrary, the second type of noises, the *directional* ones (for instance the speech produced by some other people eventually talking in the room or the stationary noises coming from concrete positions in space such as computer fans or air-conditioning systems), they do act as competitive sources to the uttered signal and introduce ambiguity in the localization estimates.

Finally, the propagation of acoustic signals in closed spaces is generally multi-path, that is, the sound wave reaches its target following different, multiple ways, see Figure 2.2. Therefore, apart form the direct path contribution, the recorded signal will contain several delayed, attenuated and distorted copies of the original speech due to reflections and diffractions with the objects and boundaries present in the environment. This multi-path phenomena is commonly known as reverberation. The room reverberation is characterized by the room impulse response, basically characterized in Figure 2.1, and its importance depends on two main factors: the size of the room, controlling the time amount that the reverberation persists, and the surfaces in that room, controlling how much energy is lost in each reflection and, thus, how many reflections and multi-paths may persist.

Thus, reverberation causes reflections of the signal to come in the recording microphone from paths and directions different to that of the original signal. This effect, depending on the strength of the reflections (being the early reflections the most limitting

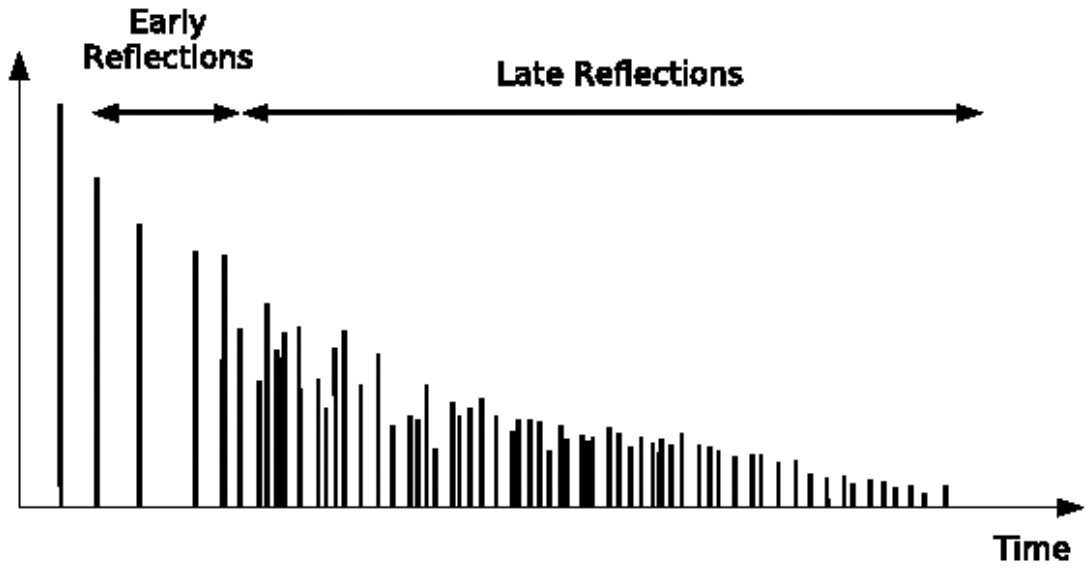

Figure 2.1: Instance of a room impulse response. Three parts can be distinguished: the direct wave, the early reflections and the late reflections

ones), can therefore mislead the localization results.

2.2.2 Direct path propagation

Under simple acoustic conditions, see [Dib00] p. 9, and in free space propagation, where sound waves are not interfered by objects such as walls, furniture or people, the signal received at a fixed microphone is linearly related to the originally uttered speech as expressed in the next equation. Although this assumption is not realistic in our small-room environments, it accurately describes the direct-path propagation of sound from source to receiver even in the presence of reverberation.

$$x_{direct}(r,t) = \frac{a}{r} \cdot s(t - \frac{r}{c}) = \frac{a}{r} \cdot s(t - \tau) \tag{2.1}$$

where $x_{direct}(r,t)$ is the signal captured at the microphone, $s(t - \frac{r}{c})$ is the original speech, a is the sound wave amplitude, r is the distance from the source, c is the sound speed and τ is the time delay between transmitter and receiver.

2.2.3 The room impulse response and the multi-path model

As described in [Dib00] pp. 9-13, when propagating within a room limitted by sound-reflecting surfaces, the uttered signal is modified according to the room accoustics. This effect has extensively been modeled by the linear systems theory: a relationship between

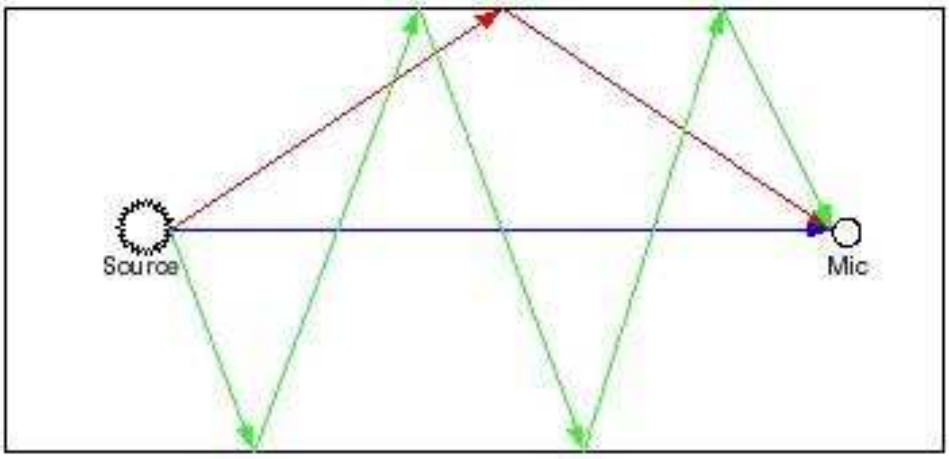

Figure 2.2: Sound propagation inside a room

the original and received signals can be stated in terms of convolution of $s(t)$ with the *room impulse response*, $h(t)$:

$$x(\vec{r}_m, \vec{r}_s, t) = s(t) * h(\vec{r}_m, \vec{r}_s, t) \tag{2.2}$$

where $s(t)$ is the source signal, $\vec{r}_s$ is the source location, $x(t)$ is the received signal and $\vec{r}_m$ is the location of the receiver microphone indexed by m.

The *room impulse response* characterizes all acoustic paths from the source to the receiver, including the direct-path one, and, as we can note, is highly dependent on the source and receiver locations. What is more, $h(t)$ varies with any environment change and temperature and it is very difficult to estimate in practical situations.

A more useful model can then be used that reflects the propagation of a direct-path sound plus the sumation of different reflected sounds as follows:

$$x(\vec{r}_m, \vec{r}_s, t) \simeq \frac{a}{r} \cdot s(t - \tau_m) + s(t) * u(\vec{r}_m, \vec{r}_s, t) \tag{2.3}$$

In this equation, the reflected sounds are modeled as a filtered version of the original signal and $u(t)$ represents the impulse response characterizing all the acoustic paths but the direct one.

$$h(\vec{r}_m, \vec{r}_s, t) \simeq \frac{a}{r} \cdot \delta(t - \tau_m) + u(\vec{r}_m, \vec{r}_s, t) \tag{2.4}$$

This new model contains thus as much information as the *room impulse response* but it expresses it in terms of an interesting, easily measurable parameter: the time delay τ_m.

In addition, a complete knowledge of $u(t)$ is not necessary for the model to be useful, although partial knowledge of it, indicating us for instance the reverberation maximum duration or the stronger multi-paths, may be of great help in methods trying to estimate τ_m.

2.2.4 Microphone signal model

We can now finally formulate the expression of the signal present at the receiving microphone indexed m when capturing a source signal uttered in an acoustic reverberant environment. According to the last model and taking into account the presence of an additive noise, $\nu(t)$:

$$x_m(t) \simeq \frac{a}{r_m} s(t - \tau_m) + s(t) * u_m(\vec{r}_s, t) + \nu_m(t) \tag{2.5}$$

For simplicity, let $\tilde{\nu}_m(t)$ be a new noise term including the reverberant noise plus the acoustic, original one:

$$\tilde{\nu}_m(t) = s(t) * u_m(\vec{r}_s, t) + \nu_m(t) \tag{2.6}$$

And, therefore, the microphone signal can be expressed as follows:

$$x_m(t) \simeq \frac{a}{r_m} s(t - \tau_m) + \tilde{\nu}_m(t) \tag{2.7}$$

Since most of the localization techniques rely on the direct-path component to make their estimations, it is convenient to use this formulation as it clearly reflects the received signal as a delayed and scaled version of the original one plus a noise component containing all the acoustic noise and reverberation components.

2.2.5 Far-field vs. near-field

As referred at the beginning of Section 2.2, speech signals propagate through spherical wavefronts. However, when the distance from the source to the microphone array, r, is much larger than the physical length of the array, R, the waves arriving to the array "seem" to be planar as the curvature of the propagating spherical wave is too small with respect to the array's size. This is called the *far-field* condition and, according to [AM01] it must satisfy the following inequation:

$$|r| > \frac{2R^2}{\lambda} \tag{2.8}$$

Whenever this condition is not satisfied we say to work in *near-field* conditions. Table 2.1 shows examples of the distances that are considered to be the limits for the far-field assumption at different frequencies for two different microphone arrays: a linear array of 33 elements equispaced 20 mm and a linear array of 4 elements equispaced 200 mm which will be later used in this Master Thesis experimentation.

	R = 660 mm	R = 800 mm
500 Hz	1.26 m	1.85 m
1 KHz	2.52 m	3.71 m
2 KHz	5.05 m	7.42 m
4 KHz	10.11 m	14.84 m
8 KHz	20.20 m	29.68 m
16 KHz	40.40 m	59.36 m
20 KHz	50.50 m	74.20 m

Table 2.1: Far-field limits for a linear array of 33 equispaced 20 mm and a linear array of 4 elements equispaced 200 mm.

This distinction turns out to be of importance when trying to compute time delay differences between microphones signals.

In the far-field situation depicted in Figure 2.3, the speech signal takes τ_0 seconds to get to microphone 0, x_0, and τ_1 seconds to get to microphone 1, x_1. It is clear to see that the wavefront needs to travel an extra distance, d', in order to get to get to x_0 compared to x_1. Then, the time delay difference between the two microphones signals will be:

$$\Delta\tau = \tau_0 - \tau_1 = \frac{d'}{c} = \frac{d \cdot sin\theta}{c} \tag{2.9}$$

where c is the speed of sound and θ is the signal Direction Of Arrival (DOA) to the array.

The previous equation gives us a simple way to determine the DOA of a speech source given its signal in two different elements of a microphone pair under a far-field assumption and set some basics about source localization: For instance, if we are able to measure the time delay difference between two microphones, $\Delta\tau$, it will be easy to determine the direction in space, θ, where the source is located. However, this far-field assumption is not always feasible, as demonstrated in Table 2.1, and there can be cases in which it will be necessary to work under near-field conditions such as in Figure 2.4.

Under this near-field situation, the time delay difference between the microphones

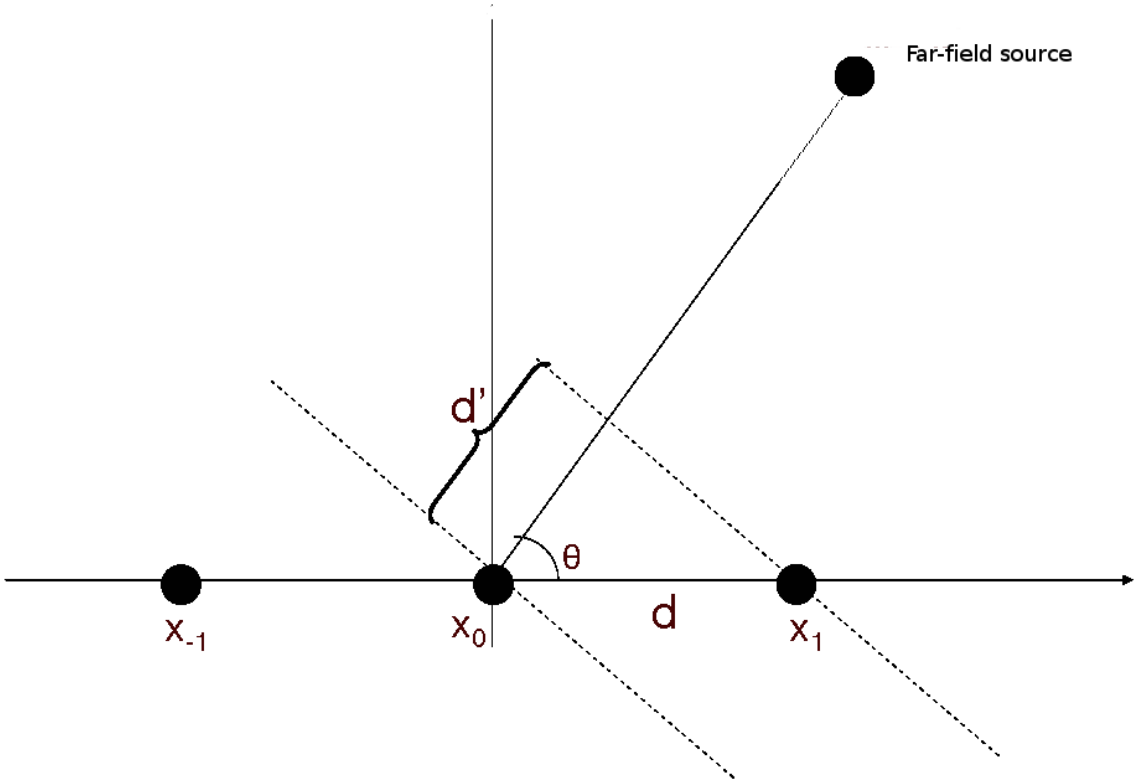

Figure 2.3: Microphone array in far-field situation

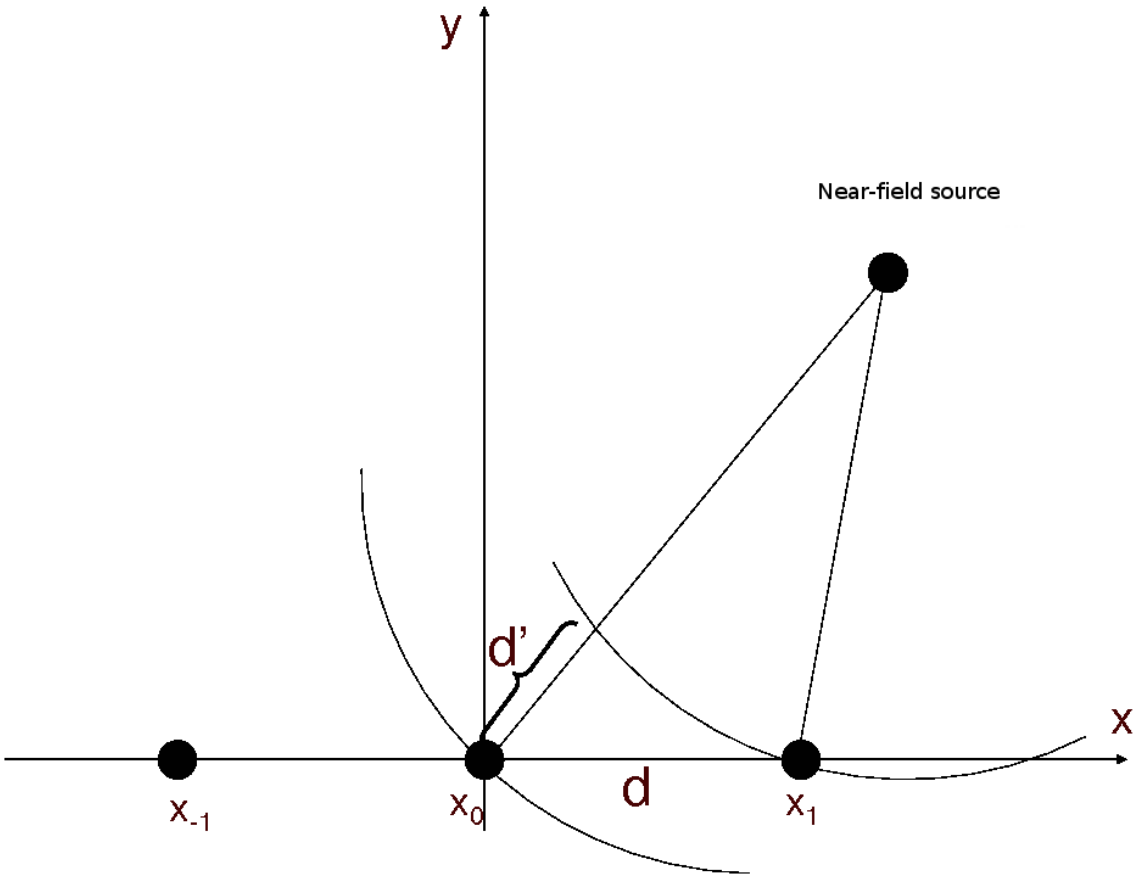

Figure 2.4: Microphone array in near-field situation

will be as follows:

$$\Delta\tau = \tau_0 - \tau_1 = \frac{d'}{c} = \frac{d_0 - d_1}{c} \tag{2.10}$$

where d_i represents the euclidean distance from the source to microphone i.

In this near-field situation is not possible to straight-ahead infer the DOA of the speech source with respect to the array (in fact it does not exist such thing since the source is so close that its DOA varies with respect to every element of the array).

2.3 Microphone arrays

A microphone array consists on a set of sensors located at specific spatial locations. They have been widely used within the speech signal processing field as they allow effective speaker localization tasks as well as enhancing and quality improvement of the captured audio signals in comparison to the results that one single microphone would obtain, [Pro06]. Their ability to perform spatial filtering, ([BW01], [VB88]), it is specially interesting to develop applications that can separate the audio source of interest from other undesired, interfering signals.

There is an extensive introduction about microphone arrays in [AM01]. We will here concentrate just on the aspects related to the speech signal processing. In general, a microphone array can be considered as the sampled version of a continous sensor being the same size as the array. The *effective length*, L, of an uniform sensor array is the length of the continous aperture which it samples, that is:

$L = Nd$ where N is the number of elements in the array and d is the inter-microphone distance.

The actual *physical length* of the array, as given by the distance between the first and the last microphone, is however $d(N-1)$.

Generally, we will consider a linear array of equispaced elements. The joint response of all its elements can be modeled as the sumation of each individual element response. Then, its directivity pattern can be expressed as follows according to [AM01]:

$$D(f,\theta,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f) e^{j\frac{2\pi}{\lambda}\sin\theta\cos\phi\cdot nd} \tag{2.11}$$

where N is the number of elements, $\omega_n(f)$ is the complex weight for element n and d is the inter microphone distance.

If we just concentrate on the horizontal directivity pattern ($\theta = \pi/2$):

$$D(f,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f) e^{j\frac{2\pi}{\lambda}nd\cos\phi} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f) e^{j\frac{2\pi f}{c}nd\cos\phi} \tag{2.12}$$

Now, as we can see in equation 2.12, the directivity pattern for a linear, equally spaced array of identical sensors depends upon three main factors:

- the number of elements in the array, N.
- the inter-element spacing, d.
- the frequency, f.

If we plot some instances of linear arrays directivity patterns in different scenarios we can appreciate several interesting features:

- If we keep L and f fixed and just vary the number of elements, N, we can appreciate in Figure 2.5 that the level of the side lobes descends and, thus, the overall directivity increases.
- If we keep N and f fixed and just vary the effective length, $L = Nd$, we can appreciate in Figure 2.6 that the main lobe width decreases, and the directivity increases likewise, as L (and thus the inter-microphone distance) is made longer.
- If we keep N and L fixed and just vary the frequency, f, we can appreciate in Figure 2.7 that as the frequency increases, the beam width will decrease, and the directivity increase likewise.

It is often interesting to have a constant beam width. Let's recall that the beam width is given by, [AM01]:

$\frac{2\lambda}{L} = \frac{2c}{Lf} = \frac{2c}{Nd \cdot f}$

Therefore, the main lobe width is inversilly proportional to the product $(Nd \cdot f)$. Given that N is fixed in most of the applications, we must ensure that the product fd remains relatively constant in order to get a constant beam width.

When designing speech recognition systems in reverberant environments, it will be interesting that the microphone array capturing the audio is as directive as possible so that we can spatially filter the undesired signals as much as possible. However, the speaker position is not known a priori and must be estimated based on localization techniques

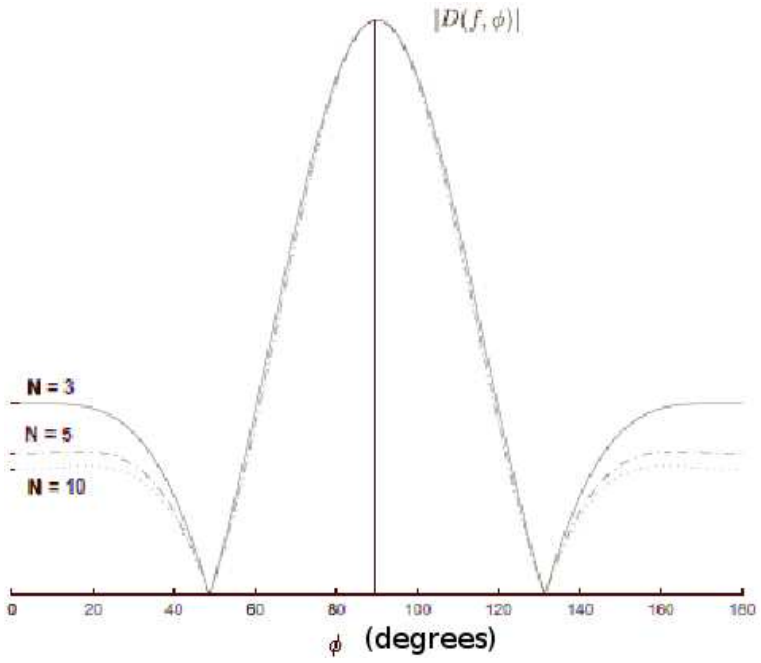

Figure 2.5: Directivity pattern for varying number of sensors (f=1KHz, L=0.5m)

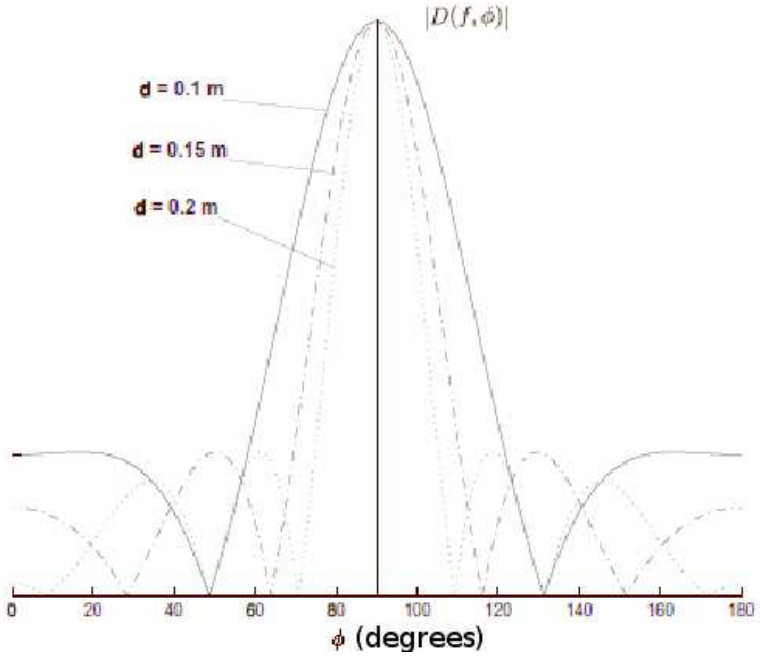

Figure 2.6: Directivity pattern for varying array effective length (f=1KHz, N=5)

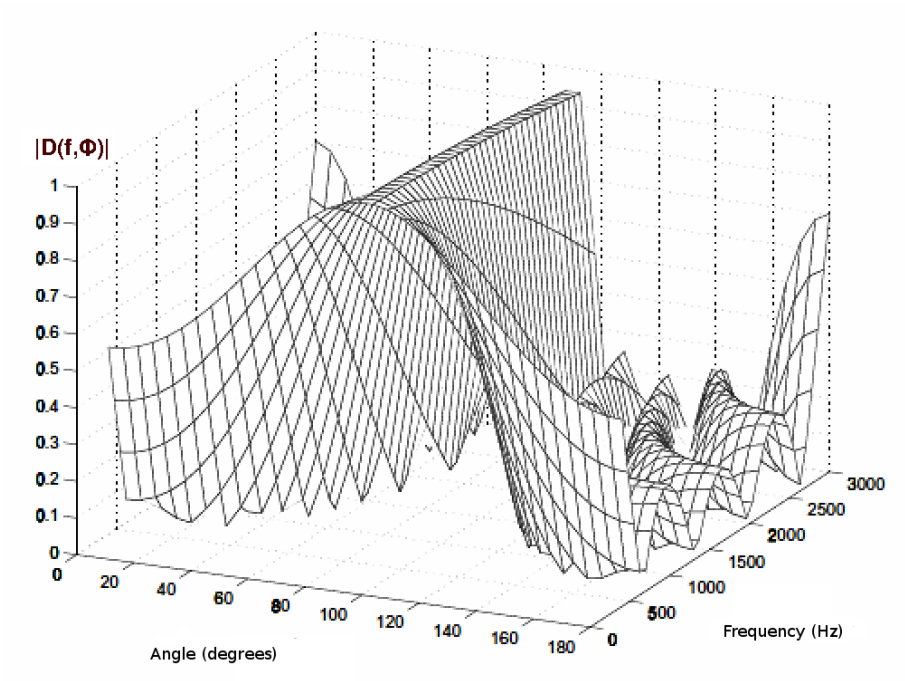

Figure 2.7: Directivity pattern for varying frequency (400Hz<f<3000Hz, N=5, d=0.1m)

that may output slighty biased locations. When pointing too accurately to this biased estimates we will also be filtering the audio source and get in consequence poor results. Therefore, there is an important trade-off between high directivity and speech recognition results as pointed out in [RF07].

2.3.1 Spatial aliasing

According to Nyquist principle, in order to avoid frequency aliasing when sampling an analog signal, the sampling frequency must be higher than two times the maximum frequency component present in that signal. This principle can also be applied to equispaced microphone arrays (which are spatially sampled version of a continous aperture) in terms of spatial aliasing, [AM01]:

$$f_s = \frac{1}{T_s} \geq 2 \cdot f_{max} : f_{x_s} = \frac{1}{d} \geq 2 \cdot f_{x_{max}} \tag{2.13}$$

where $f_{x_{max}}$ is the highest spatial frequency component and f_{x_s} is the spatial sampling frequency expressed in samples per meter and given by:

$$f_{x_s} = \frac{\sin\theta\cos\phi}{\lambda} : f_{x_{max}} = \frac{1}{\lambda_{min}} \tag{2.14}$$

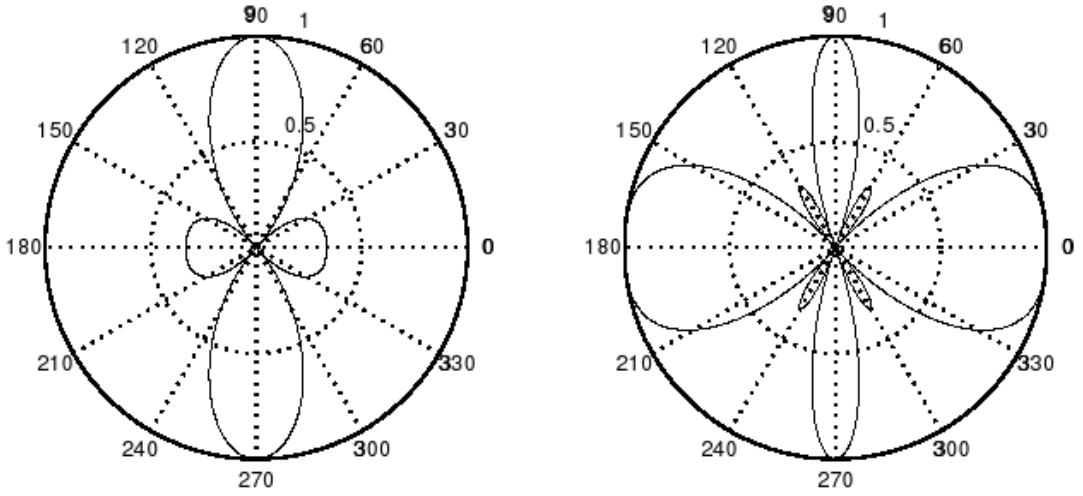

Figure 2.8: Array directivity pattern withouth (a) and with (b) spatial aliasing

And consequently:

$$d < \frac{\lambda_{min}}{2} \tag{2.15}$$

where λ_{min} is the minimum wavelength in the signal of interest and d is the inter-microphone distance that must be respected in order to avoid spatial aliasing, that is, the appearance of grating lobes at undesired directions of space as depicted in Figure 2.8.

2.3.2 Beamforming

Let's assume far-field conditions and consider the horizontal directivity pattern of a linear array depicted in Figure 2.5 for instance. As we can see, the maximum gain is offered to signals coming from direction $\phi = 90^o$. *Beamforming* is the technique that allow us to steer our array directivity pattern to a different spatial direction, see Figure 2.9.

Let's recall equation 2.11 in page 15 and focus on the complex weight parameter applied to each microphone, $\omega_n(f)$. As described in [Zio95], this set of values can be shaped according to different types of functions, called amplitude windows, in order to control the main lobe width and the secondary lobes power of the directivity pattern.

In all the Figures displayed up to now, we had assumed equally weighted sensors when calculating the directivity pattern, assuming N sensors:

$\omega_n(f) = \frac{1}{N}$

In general, complex weighting can be expressed as follows:

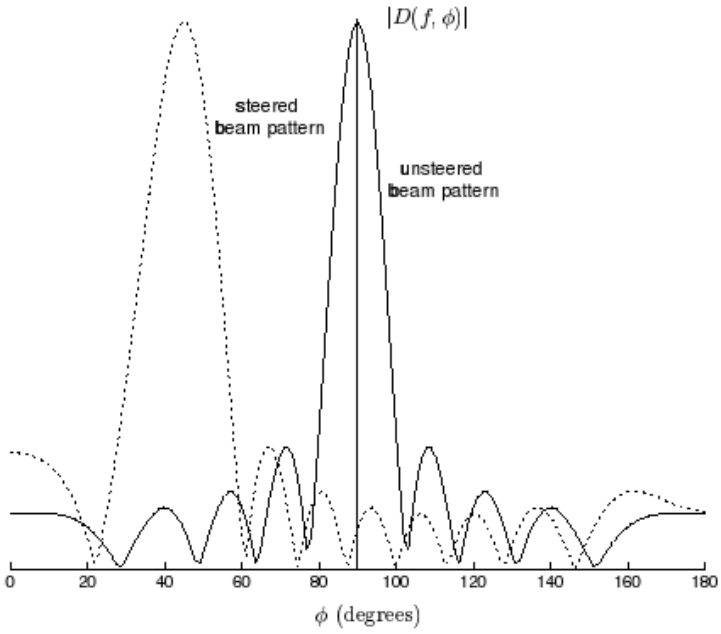

Figure 2.9: Unsteered and steered directivity patterns (ϕ'=45 degrees,f=1KHz,N=10,d=0.15m)

$\omega_n(f) = a_n(f)e^{j\varphi_n(f)}$

where $a_n(f)$ and $\varphi_n(f)$ are the real amplitude and phase weights respectively. By modifying the amplitude weights, $a_n(f)$, we can modify the shape of the directivity pattern, while modifying the phase weights, $\varphi_n(f)$, can control the angular location of the main lobe. Beamforming techniques determine these phase weights in order to get the desired steering of the array directivity as follows. Modifying equation 2.11:

$$D(f,\theta,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} a_n(f)e^{j\frac{2\pi}{\lambda}\sin\theta\cos\phi nd+\varphi_n(f)} \tag{2.16}$$

If we use the phase weights, $\varphi_n(f)$:

$$\varphi_n(f) = -2\pi\frac{\sin\theta'\cos\phi'}{\lambda}nd \tag{2.17}$$

the directivity patterns steers to the θ' and ϕ' directions. It is important to note that, since we are not modifying the amplitude weights but just the phase ones, the only difference between the unsteered and steered patterns is the direction the point to and not their shape and levels which remain equal as seen in Figure 2.9.

2.3.2.1 Delay-and-sum beamformer

Delay-and-sum beamforming is the simplest of all beamforming techniques. It makes use of the time delay property describe above to improve the quality of the signal acquired. Equation 2.18 describes the scheme applied to a N elements equispaced array:

$$y[n] = \sum_{m=0}^{N-1} a_m x_m[n - \tau_m] \tag{2.18}$$

where $y[n]$ is the beamformer overall signal, $x_m[n]$ is the signal captured by microphone m and a_n is the weight given to microphone m, generally set to be $1/N$ for every microphone. τ_m is the delay associated to microphone m and it is given in sample units, that is, the delay in seconds multiplied by the sampling frequency.

Improvements in the overall signal captured by a delay and sum beamformer are achieved because the desired signal, coming from the direct path, sum in phase increasing their power. Meanwhile, undesired signals and noise coming from different directions sum out of phase and decrease their power.

Since delaying a signal in the time domain is equivalent to multiplying by an exponential in the frequency domain, signal $y[n]$ can also be obtained by first multiplying the FFT transforms of the microphone signals by the proper exponentials and later perform the IFFT transform as shown in Figure 2.10. This operation can be interpreted as applying phase weights, $e^{j\varphi_n(f)}$, whose value corresponds to [AM01]:

$$\varphi_n(f) = 2\pi f \tau_n = 2\pi f \frac{(n-1)d\sin\theta' f}{c} \tag{2.19}$$

Let's recall that, as seen in Section 2.2.5 in page 11, under far-field conditions, the time delay in the signal between two adjacent microphones in an equispaced linear array is given by:

$$\tau = \frac{d\sin\theta}{c}$$

2.3.2.2 Filter-and-sum beamformer

The filter-and-sum beamforming algorithm is considered to be a generalization of the delay-and-sum beamformer as it simply consists on adding a pre-filter to each microphone channel.

Hence, the overall signal obtained by a filter-and-sum algorithm can be expressed as, [Sel03]:

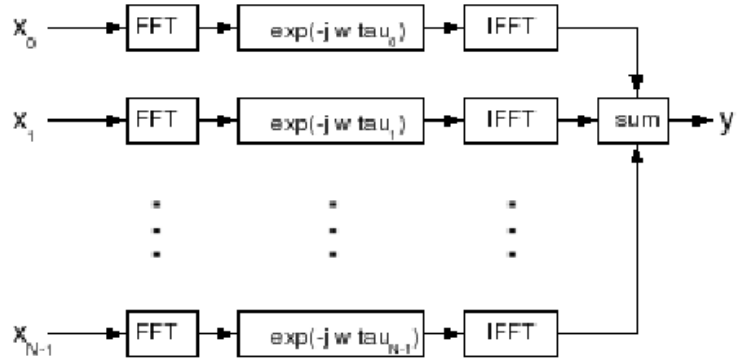

Figure 2.10: Frequency domain block diagram of a delay-and-sum beamformer

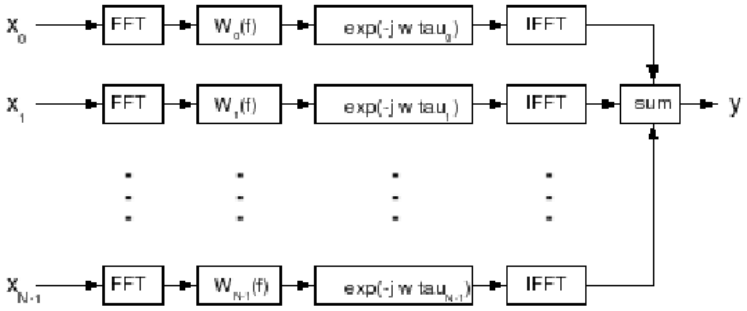

Figure 2.11: Frequency domain block diagram of a filter-and-sum beamformer

$$y[n] = \sum_{m=0}^{N-1} \sum_{p=0}^{P-1} h_m[p] \cdot x_m[n - p - \tau_m] \tag{2.20}$$

where $h_m[p]$ is the filter associated to microphone m.

Once again, we can translate this operation into the frequency domain according to the diagram shown in Figure 2.11.

It is important to note that, depending on the filter chosen, not only the phase but also the amplitude weights of a filter-and-sum beamformer can vary.

2.3.2.3 Constant Directivity Beamforming (CDB)

Human speech frequency range covers a band ranging from 20 Hz to 20 KHz. This spectra is significally wide since it covers different frequency octaves. As we saw in Figure 2.7 in page 18, an array directivity pattern, and particularly its beamwidth, are highly dependent on the frequency of the input signal. Specifically, the main lobe width is inversally proportional to the product fd.

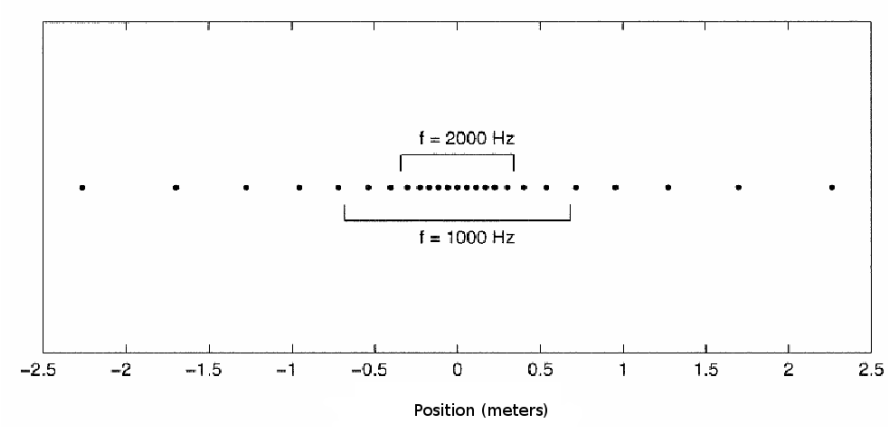

Figure 2.12: Geometry of a CDB array of 25 elements (adapted from [AM01])

Hence, using this type of arrays for spatial filtering applications working with human speech does not output optimal results: The undesired signals coming from undesired directions will in fact be attenuated at high frequency bands, where the main lobe is more precise, but this attenuation will be lower and lower as the frequency decreases and the main lobe gets wider, therefore leading to strong interferences at the low frequency ranges.

In order to solve this problem, it was suggested to design a Constant Directivity Beamforming (CDB) array, see Figure 2.12. This type of arrays offers a constant directivity pattern through wide frequency bands, see Figure 2.13. The technique to get such behaviour, depicted in [AM01], consists on using arrays whose elements are able to form different equispaced sub-arrays, each of them having a different inter-microphone distance, d, and therefore being suited to deal with different frequency bands. The responses of these sub-arrays are later on combined through an appropiate pass-band filtering to achieve the desired result.

2.4 Localization algorithms

As exposed above in Section 2.2 in page 7, those situations with the speaker standing in a close room far-away from the receiver microphone are subject to poor signal to noise ratio and reverberation effects. Making use of a microphone array is an efficient way to attenuate these phenomena as seen in Section 2.3 in page 15. In general, the speaker will not stay static and it will be necessary to track its localization around the room in order to take advantage of the microphone array characteristics: Centering the array reception pattern around the source localization in order to avoid undesired noise, audio sources and reverberations.

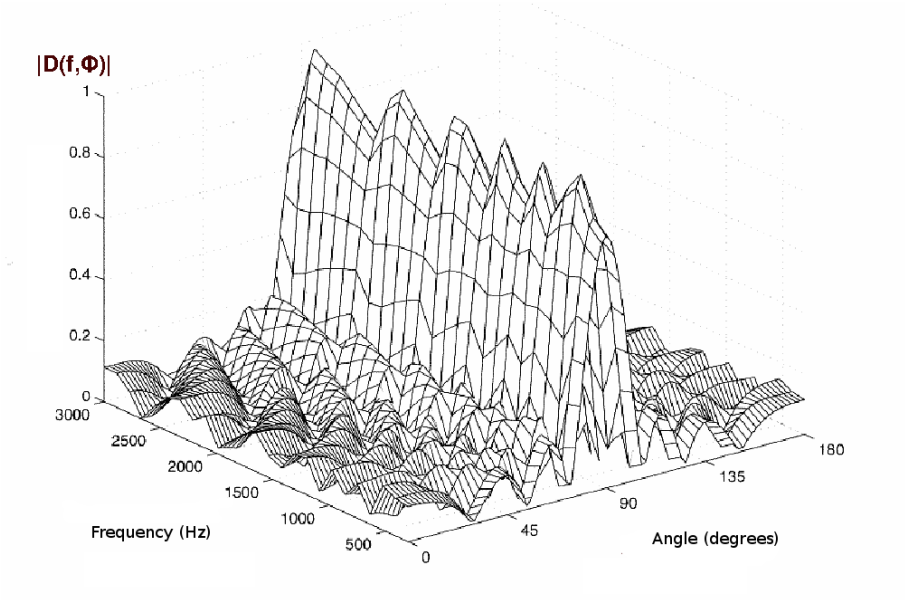

Figure 2.13: Directivity pattern of a CDB array as a function of frequency (adapted from [AM01])

The primary goal of a speech localization system is accuracy. In general, and according to [BW01] p. 157, the system estimates precision is dependent mainly on:

- The quantity and quality of the microphones involved.
- The microphone placement relative to each other and the speech sources to be analyzed.
- The ambient noise and reverberation levels.
- The number of active sources and their spectral content.

The systems results generally improve then with the number of microphones in the array, particularly with adverse acoustic conditions. Arrays with a large number of microphones (up to 512, [HFSF96]) have been built. However, when acoustic conditions are reasonable and microphone placement is proper, source localization can be performed adequately using a lower number (as low as 4 elements for instance). Performance is then fully dependent on the array geometry and its optimal design is often strongly related to the environment acoustic conditions and geometry as well as to the specific application conditions.

Apart from accuracy, this systems are asked to work with certain speed, adapted to real-time conditions, so that the localization can be effective and properly adapt to the source movements in time. Hence, the localization estimates must be updated at a rate frequent enough.

Several algorithmic approaches are available for speech source localization, this Section summarizes the main ones and makes some comments on the general merits and drawbacks of each one. There are mainly three possible methods for audio source localization:

- those employing *Time Difference of Arrival (TDOA)* information.
- schemes using *high-resolution spectral estimation* concepts.
- approaches based on maximizing the *Steered Response Power (SRP)*.

2.4.1 Based on Time Delay Estimation (TDE)

This first localization strategy is based on a two-steps procedure.

- First, the Time Difference Of Arrival (TDOA) of the speech signals relative to pairs of spatially separated microphones is performed. According to [OS94], there are three different TDOA estimation techniques:

  The normalized Cross Correlation (CC).

  The Crosspower-Spectrum Phase (CSP) analysis, so-called the *Generalized Cross-Correlation (GCC).*

  The *Least Mean Squared (LMS)* adaptive filters.

  Comparison among these three techniques carried out in [OS94] demonstrated that GCC methods, based on the basic CC, show the best properties for the estimation of the wavefront arrival direction. This Master Thesis will mainly concentrate on these two techniques depicted in Sections 2.4.1.1 and 2.4.1.2 respectively.

- Secondly, once the TDOA is known, we can make use of it along with the information about the spatial microphone positions in order to generate the hyperbolic curves that represent the geometrical places where the speaker is likely to be according to the given TDOA obtained. An extensive study about this hyperbolic curves can be found at [MH06] pp. 31-43. These hyperbolic curves are then intersected in some optimal sense in order to arrive to a source location estimate. A number of variations of this principle have been developed, [Var02] pp. 24-27, [PSO97] are examples. They differ considerably in the method of derivation of the source coordinates as well as in the extent of their applicability (2D vs. 3D, near-field vs. far-field, etc.). An instance of one of these derivations is depicted in Section 2.4.1.4.

These TDOA-based localization procedures main advantage is their simplicity and low-computational cost. Nevertheless, their utility in realistic, acoustic environments is

limited since their performance has shown to clearly decrease in high noise or reverberation scenarios. What is more they are not suited for multi-source localization and the position estimate they output generally consists on a Direction Of Arrival (DOA) estimation rather than an exact spatial localization. Steered-Beamformer strategies are computationally more intensive, but tend to offer more robust localizations as seen in [BW01] pp. 164-178.

2.4.1.1 The normalized Cross Correlation (CC)

As explained above, a way of localizing an acoustic source consists on finding an estimation of the time delay, τ, so-called Time Difference of Arrival (TDOA), between the speech signals captured by a pair of spatially separated microphones in an array. The equations of these speech signals, arriving to microphones i and j from the acoustic source, can be expressed as follows:

$$\begin{cases} x_i(t) = \alpha_i \cdot s(t) + n_i(t) \\ x_j(t) = \alpha_j \cdot s(t + \tau_{ij}) + n_j(t) \end{cases} \tag{2.21}$$

where $s(t)$ is the uttered signal, n_i and n_j are the noises captures by each microphone, α_i and α_j are the attenuations and τ_{ij} the time delay between signals due to the distance difference between the source and the two microphones.

The most common way to determine the time delay difference, τ_{ij}, given the signals $x_i(t)$ and $x_j(t)$, requires to compute the Cross Correlation (CC) function, $c_{x_i x_j}(\tau)$,that analyzes the similitude between two different signals, for every time delay, τ:

$$c_{x_i x_j}(\tau) = E[x_i(t) \cdot x_j(t - \tau)] \tag{2.22}$$

Given the equation 2.21, expression 2.22 ends up being:

$$c_{x_i x_j}(\tau) = \alpha_i \alpha_j \cdot c_{s_i s_j}(\tau - \tau_{ij}) + c_{n_i n_j}(\tau) \tag{2.23}$$

where $c_{s_i s_j}(\tau)$ represents the autocorrelation of the source signal $s(n)$ at lag τ.

τ_{ij} could be theoretically derived just by maximizing $c_{x_i x_j}(\tau)$ function with respect to τ. However, due to the finite observation time, this function can only be estimated for a given temporal window of length T. We denote this estimate as $\hat{c}_{x_i x_j}(\tau)$:

$$\hat{c}_{x_i x_j}(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_i(t) \cdot x_j(t - \tau) dt \tag{2.24}$$

And the TDOA estimation, $\hat{\tau}_{ij}$, can be derived from it as follows:

$$\hat{\tau}_{ij} = arg\ max_{\tau}\ \hat{c}_{x_i x_j}(\tau) \tag{2.25}$$

An alternative way to compute the CC function can be found by applying the *Discrete Correlation Theorem*, [OS89], to equation 2.22, resulting:

$$c_{x_i x_j}(\tau) \equiv DFT(x_i(t)) \cdot DFT'(x_j(t)) \tag{2.26}$$

where $DFT(x)$ is the Discrete Fourier Transform of $x(t)$ and "$'$" denotes the signal complex conjugate.

DFT can be efficiently computed using the Fast Fourier Transform (FFT) algorithm. Then, according to equation 2.26, the CC expression when using FFTs looks as follows:

$$c_{x_i x_j}(\tau) = Re[IFFT(FFT(x_i) \cdot FFT'(x_j))](\tau) \tag{2.27}$$

where $Re[]$ denotes the real part of a complex function.

The Cross Correlation turns out to be a good technique to estimate the time delay when signals are just affected by uncorrelated noise sources. Nevertheless, it can easily fail in presence of strong reverberation since the signal will be strongly correlated to its replicas, therefore leading to other peaks placed at wrong lags that may mislead the estimates.

2.4.1.2 The Generalized Cross Correlation (GCC)

There is a more general version of expresion 2.22 called Generalized Cross Correlation (GCC) (see [KC76]) that consists on prefiltering the signals before computing its correlation in order to improve, [OS94], the results offered by the common cross correlation. GCC function, $c^{(g)}_{x_i x_j}(\tau)$, is given by the following expression:

$$c^{(g)}_{x_i x_j}(\tau) = E[(h_i(t) * x_i(t)) \cdot (h_j(t-\tau) * x_j(t-\tau))] \tag{2.28}$$

The Fourier transform of the cross correlation function is known as the *Cross-Power Spectrum (CSP)* and denoted as $C^{(g)}_{x_i x_j}(\omega)$:

$$C^{(g)}_{x_i x_j}(\omega) = \int_{-\infty}^{\infty} c_{x_i x_j}(\tau) e^{-j\omega\tau} d\tau \tag{2.29}$$

By substituting equation 2.29 into equation 2.28 and applying the convolution property of Fourier transforms, the Cross-Power Spectrum can be expressed in terms of the Fourier transforms as follows:

$$C_{x_i x_j}^{(g)}(\omega) = (H_i(\omega)X_i(\omega)) \cdot (H_j'(\omega)X_j'(\omega)) \tag{2.30}$$

If we define the frequency dependent weighting function, $\Phi_{x_i x_j}(\omega) = H_i(\omega) \cdot H_j'(\omega)$, and apply the inverse Fourier transform to equation 2.30 we get the GCC function as:

$$c_{x_i x_j}^{(g)}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_{x_i x_j}(\omega) X_i(\omega) X_j'(\omega) e^{-j\omega\tau} d\omega \tag{2.31}$$

Ideally, with an appropiate weighting function, $c_{x_i x_j}^{(g)}(\tau)$ should exhibit a peak which corresponds to the exact TDOA between microphones i and j. The TDOA estimate, once again, is the time lag that maximizes $c_{x_i x_j}^{(g)}(\tau)$:

$$\hat{\tau}_{ij} = arg\ max_\tau\ c_{x_i x_j}^{(g)}(\tau) \tag{2.32}$$

Note that finding $\hat{\tau}_{ij}$ requires a simple, low-cost, one-dimensional search. In general, function $c_{x_i x_j}^{(g)}(\tau)$ will hold several maxima. The amplitudes and time lags of these maxima will depend on a serie of factors including the levels of noise and reverberation, the separation distance between microphones and the choice of the weighting function $\Phi_{x_i x_j}(\omega)$.

Several different prefilter functions, $\Phi_{x_i x_j}(\omega)$, have been proposed. An extensive definition and discussion of them can be found in [KC76]:

- *The Roth Processor*, has the desirable effect of suppressing those frequency regions where the noise power spectrum is large and the estimation is more likely to be in error.

$$\Phi_{x_i x_j}^{Roth}(\omega) = \frac{1}{C_{x_i x_i}(\omega)} \tag{2.33}$$

- *The Smoothed Coherence Transform (SCOT)*, improves the results got by Roth by weighting the signals according to their SNR characteristics.

$$\Phi_{x_i x_j}^{SCOT}(\omega) = \frac{1}{\sqrt{C_{x_i x_i}(\omega) C_{x_j x_j}(\omega)}} \tag{2.34}$$

- *The Maximum Likelihood (ML) Estimator*, imposes to know the spectral properties of

the source signal, $S(\omega)$ and the noise, $N(\omega)$ that, in practice, must be estimated.

$$\Phi_{x_i x_j}^{ML}(\omega) = \frac{|S(\omega)|}{|N_i(\omega)||N_j(\omega)|} \cdot (1 + \frac{|S(\omega)|}{|N_i(\omega)|} + \frac{|S(\omega)|}{|N_j(\omega)|})^{-1} \tag{2.35}$$

- *The Phase Transform (PHAT)*,
  Expression in equation 2.23, page 26, can be rewritten in the following way:

$$c_{x_i x_j}(\tau) = \alpha_i \alpha_j \cdot c_{s_i s_j}(\tau) * \delta(t - \tau_{ij}) + c_{n_i n_j}(\tau) \tag{2.36}$$

  where $*$ denotes convolution.

  For uncorrelated noises $c_{n_i n_j}(\tau) = 0$ and expression 2.36 can be interpreted as a delta function placed at the proper time lag but spread or "smeared" by the Fourier transform of the source signal spectrum. However, if $s(t)$ were a white noise source, then its Fourier transform would be a delta function and no spreading would take place.
  Therefore choosing a pre-whitening filter seems to be an optimal strategy. According to this scheme, the PHAT weighting function has been defined to be:

$$\Phi_{x_i x_j}^{PHAT}(\omega) = \frac{1}{|C_{x_i x_j}(\omega)|} = \frac{1}{|X_i(\omega) X_j'(\omega)|} \tag{2.37}$$

  By placing equal emphasis on each frequency, the PHAT weighting is sub-optimal under reverberation-free conditions, see [BS97], yet performs considerably better than other prefilters in realistic environments. One apparent defect of the PHAT is to weight the signal as the inverse of its modulus. Thus, errors are accentuated where signal power is smallest. A bandpass weight has been proposed, see [Dib00] p. 46, in conjunction with PHAT in order to emphasize only the frequency bands where most of the speech energy lies.

In conclusion, the GCC-PHAT is the method showing more interesting results. It has been shown to be effective in real situations, see [PSO97], and it will be the scheme implemented in this Master Thesis.

2.4.1.3 GCC-PHAT implementation

There are some practical details we must take into account when implementing the algorithm depicted above. First of all, as hinted in Section 2.4.1.1 in page 26, we cannot take infinite observation time but just a certain temporal window in order to analyze the signals. Secondly, our analysis will not use the original, analog signals but digital, discrete versions of them got after a sampling proccess.

Given this, the discrete-time microphone signals will be denoted $x_1[n]...x_M[n]$. Any source localization technique will start by segmenting these signals into blocks and, generally, applying the Discrete Fourier Transform (DFT) to each of them afterwards. Each block of data is usually windowed with a tapered window prior to the DFT application in order to improve the signal spectral representation as this mainly eliminates the effects caused by the discontinuities at the ends of the blocks. Consecutive data blocks usually overlap in the time-domain to let the data placed at the end of one block, which is supressed by the tapered window, to be centered in the next one, giving all data an equal weight in the analysis. Any source localization algorithm will operate on the DFT of each data block to produce a location estimate (under the assumptiion that the source location is stationary for the duration of each block). Since each block advances in time, the algorithms are able to track moving speakers. The rate at which location estimates are produced depends on the advance of the data blocks, so-called the *frame shift*, and the latency of each estimate depends on the *frame size*.

Now, the expressions for the discrete-time microphone signals $x_1[n]...x_M[n]$ and their DFTs when segmented into blocks of length N are:

$$x_{m,b}[n] = w[n] \cdot x[bA + n] \; for \; n = 0...N - 1 \tag{2.38}$$

where $x_{m,b}[n]$ is the windowed data of the m-th microphone and the b-th block. A is the frame shift, a constant, positive integer that defines the block advance. The blocks overlap when $A < N$, a typicall set is $A = \frac{N}{2}$. $w[n]$ is the window function, a typical chose is the Hanning window, see appendix B in page 175.

The K-point Discrete Fourier Transform (DFT) of the previous block, $x_{m,b}[n]$, can be expressed as follows:

$$X_{m,b}[k] = \sum_{n=0}^{N-1} x_{m,b}[n] e^{-jk\frac{2\pi}{K}n} \; for \; k = 0...K - 1 \tag{2.39}$$

Note that the DFT length is K and $K \geq N$. Hence, the signal data block needs to be zero-padded to increase its length before the K-points DFT is performed. Generally, K will be chosen to be a multiple of 2 in order to allow quick computations of the DFTs via the Fast Fourier Transform (FFT) algorithm.

Now, the expression of a DFT-based GCC-PHAT function between microphones i and j in the data block b, $\hat{c}_{ij,b}(\hat{\tau})$, can be defined by substituting in equation 2.31, page 28, the Fourier transforms for the DFT blocks previously defined:

$$\hat{c}_{ij,b}(\hat{\tau}) = \frac{1}{K}\sum_{k=0}^{K-1}\Phi_{ij}[k]X_{i,b}[k]X'_{j,b}[k]e^{jk\frac{2\pi}{K}\hat{\tau}} = \frac{1}{K}\sum_{k=0}^{K-1}\Phi_{ij}[k]C_{ij,b}[k]e^{jk\frac{2\pi}{K}\hat{\tau}} \tag{2.40}$$

where $\Phi_{ij}[k]$ is the discrete version of the frequency weighting function $\Phi_{ij}(\omega)$ and $\omega_k = \frac{2\pi k}{K}$ is the DFT frequency index.

Taking into account the theorem in equation 2.26 in page 27:

$$\hat{c}_{ij,b}(\hat{\tau}) = Re[IFFT(\Phi_{ij}[k]X_{i,b}[k]X'_{j,b}[k])](\hat{\tau}) = Re[IFFT(\frac{X_{i,b}[k]X'_{j,b}[k]}{|X_{i,b}[k]||X_{j,b}[k]|})](\hat{\tau}) \tag{2.41}$$

This expression shows us how to implement the GCC-PHAT function based on the FFT of the signals captured by a microphone pair. The TDOA estimate between those two microphones can be found by searching the lag at which this GCC-PHAT function, $\hat{c}_{ij,b}(\hat{\tau})$, is maximum.

The separation distance between the microphones, d, physically limits the range of valid time delays. The largest TDOA (Time Difference Of Arrival) possible is then that of $\frac{d}{c}$, where c is the sound speed. Therefore $\tau\epsilon[-\frac{d}{c}, \frac{d}{c}]$. However, there is a important remark to do about the possible values of τ in the case of a DFT-based, discrete version of the GCC-PHAT function. While τ is a continous variable in equation 2.31 in page 28, equation 2.40 is discrete and all its values are sampled in practice. Therefore, there will be an inherent loose of precision when translating the real-unit time delays in seconds, τ, to discrete-unit time delays in samples, $\hat{\tau}$, as shown in the next expression:

$$\hat{\tau} = round(\tau[sec] \cdot f_s[sec^{-1}]) \tag{2.42}$$

Assuming far-field conditions and according to equation 2.9 in page 12, equation 2.42 can be rewritten as:

$$\hat{\tau} = round(\frac{d \cdot sin\theta}{c} \cdot f_s) \tag{2.43}$$

As we can see, the imprecision committed depends upon four different factors:

- The intermicrophone distance, d. The longer it is, the smaller the imprecision should be. Experimental results about this statement can be found in the experimental corpus of this Master Thesis in Section 3.4 in page 71.

- The sampling frequency, f_s. The higher it is, the smaller the imprecision should be. A technique that artificially increases the sampling frequency by an interpolation process that adds zeros to the FFTs was proposed by [Var02] pp. 60-68 with the objective of achieving sub-sample resolution. Experimental results about this statement can be found in the experimental corpus of this Master Thesis in Section 3.5.1 in page 73.

- The Direction Of Arrival (DOA), θ. The more tilted it is, the smaller the imprecision. The particular case when $\theta = 0$ makes $\tau = 0$ and no rounding imprecision is committed. Experimental results about this statement can be found in the experimental corpus of this Master Thesis in Section 3.7 in page 120.

- The *rounding* function. We can think of different rounding functions. The objective will be to evaluate the discrete GCC functions in such a way that the values obtained resemble the continous GCC functions as much as possible. In other words, the objective is to emulate as precisely as possible those continous values of the GCC functions that are not present in their discrete versions (since they are located at lags placed in between the gaps left by the discrete, finite set of time delays computed by the IFFT). More details about these schemes can be found at Section 2.4.3.3 in page 37.

2.4.1.4 A source location method based on TDOA

Once the Time Difference Of Arrival (TDOA) has been estimated thanks to the implementation of some of the previous methods, preferably GCC-PHAT, we can make use of this data, all toguether with the array geometry information, in order to ouput a proper speaker localization estimate. Various methods to do so have been proposed.Here, we will just show one of them: a simple 2D version proposed by Varma in [Var02] pp. 24-27. Designed to work under far-field conditions, it makes a position estimate based on a *Least Mean Square* (LMS) method.

As seen in Figure 2.3 in page 13, the TDOA, τ_{ij} between the signals captured by two microphones i and j can be expressed as follows:

$$d_{ij} sin\theta = -c\tau_{ij} \tag{2.44}$$

First we make a TDOA estimation via the GCC-PHAT method for every possible microphone pair combination of the array. Then, this estimates are stored in a vector τ. Likewise, we can also store all the microphone pair distances in another vector $\mathbf{d}$, which transforms equation 2.44 into:

$$\mathbf{d} \cdot sin\theta = -c \cdot \tau \tag{2.45}$$

Expression 2.45 represents an equation system, just with one unknown parameter, the DOA (θ), that can be solved to get a θ estimation via the *Least Mean Square* (LMS) method as follows:

$$\hat{\theta} = arc\ sin[(\mathbf{d}^T \cdot \mathbf{d})^{-1}\mathbf{d}^T(-c\tau)] \tag{2.46}$$

It is important to note that this method does not provide an estimate of the exact spatial source position but rather a Direction Of Arrival (DOA) estimation. Nevertheless, this θ estimate is equally valid when trying to steer our array to the proper direction where the speaker lies.

2.4.2 Based on Spectral-Estimation-Based Sub-Spaces

These methods aim at getting an estimate of the signal power distribution so that they can detect the energy peaks present on it. In order to do so, they make use of an Eigen Value Descomposition (EVD) of the cross correlation matrix to divide it into two sub-spaces: One containing the speech signal and the other containing the noise signal. Both the source and the noise are supposed to be stationary and their positions fixed. An instance of this type of methods is the so-called MUSIC (MUltiple SIgnal Classification), more details can be found at [Sch86].

These methods are specially suited for multi-source scenarios and are able to distinguish close sources more accurately than SRP schemes since the algorithm outputs sharp peaks at the correct directions. However, these techniques have been designed for narrow band sources and their extent to broad band signals such as speech is complex and heavily accentuates the computational load. In addition, they tend to be less robust to source and sensor modeling errors than conventional beamforming methods such as SRP. Primarly for these reasons, localization methods based on these high-resolution strategies will not be considered further in this Master Thesis.

2.4.3 Based on Steered Response Power (SRP)

Many digital signal processing techniques rely on the ability of microphone arrays to *focus* to particular locations or directions in space. These techniques make use of some type of *beamforming* which can be applied either to source signal capture or to source localization. If the source position is known, the beamformer can be focused to it in order to

output an enhanced version of the signal, see [RF07]. If the source location is not known, the beamformer can be used to scan, or steer the array, over a set of spatial locations in a predefined search space. When used this way, the output of the beamformer is known as the *steered response*. After this, a Maximum Likelihood (ML) estimator searches for a maximum peak in the output power that should coincide with the speaker location.

The simplest type of steered response is obtained using the output of a delay-and-sum beamformer, see Section 2.3.2.1 in page 21. Different time shifts, designed to match to the source signal propagation delays, are applied to the array signals. These signals are then time-aligned and summed toguether to form a single output signal. More general and sophisticated beamformers apply filters to the array signals as well as this time aligment, see Section 2.3.2.2 in page 21. When beamforming techniques are applied to voice capture applications, these filters must aim not only at suppresing undesired background noise and unwanted sources, but also at not significally distorting the desired signal. However, when beamforming techniques are used just for source localization, these filters need only to boost the power of the desired source signal in the beamformer output. With this need in mind, we will make use of Phase Transform (PHAT) filters, see Section 2.4.1.2 in page 27, that have demonstrated to be useful in terms of TDOA estimation although they obviously distort the input signals. This way, we will get a steered response useful for localization purposes but not for voice capture. In Section 2.4.1 in page 25, it was stated that the GCC technique for TDOA estimation did not output estimates robust enough under high noise and reverberation. It has been hypothesized, see [Dib00] pp. 83-84, that the incorporation of multiple microphone signals may improve the performance of this pairwise technique. Given this background, a robust technique was proposed, see [BW01] pp. 164-178, that makes the Steered Response Power (SRP) equivalent to the sum of all possible combinations of pairwise phase transforms. This technique has been named SRP-PHAT and its robustness lies on the fact that exploits the spatial microphone redundancy by averaging all possible pairwise GCC-PHAT crossings.

SRP methods offer better and more robust localization results than TDOA-based ones, [Var02] pp. 122-123. They have also been successfully extended to the case of multiple signal sources, [WK83]. It is mainly because of these two reasons that we decided to rely on this algorithm as the principle source localization method in this Master Thesis. Its main shortcoming is their high computational load that particularly increases with the growing number of microphones in the array as well as with the increase in the set of spatial locations where to steer to.

2.4.3.1 The SRP-PHAT algorithm

As seen in Section 2.3.2.2 and Figure 2.11 in page 21, the output of a M-element filter-and-sum beamformer can be defined in the frequency domain as:

$$Y(\omega, \mathbf{q}) = \sum_{n=1}^{M} W_n(\omega) X_n(\omega) e^{j\omega\Delta_n} \tag{2.47}$$

where Δ_n is the appropiate *steering delay* in microphone n for focusing the array to the spatial location $\mathbf{q}$ and $X_n(\omega)$ and $W_n(\omega)$ are the Fourier transforms of the n-th microphone signal and its associated filter.

This is equivalent to the time-domain beamformer that can be used as a mean for source localization by steering the array to a set of specific spatial points of interest and analizing the power of the output signal in each one of them. When the focus corresponds to the location of the source, the SRP should reach a global maximum. The expression for the Steered Response Power for a spatial location $\mathbf{q}$ can be expressed as the output power of the filter-and-sum beamformer:

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} |Y(\omega)|^2 d\omega = \int_{-\infty}^{\infty} Y(\omega) Y'(\omega) d\omega \tag{2.48}$$

And the correct localization estimate, $\hat{\mathbf{q}}_\mathbf{s}$, is found as:

$$\hat{\mathbf{q}}_\mathbf{s} = arg\ max_q\ P(\mathbf{q}) \tag{2.49}$$

However, this power function may in practice peak at a number of incorrect locations as well due to either strong reflective conditions or the effect of the array geometry and signal conditions, therefore misleading the localization results. Choosing the appropiate filters can help to minimize these effects. As seen, the strategy followed by the Phase Transform (PHAT) of weigthing each frequency component equally has proved to be advantageous for practical situations. Joining the advantages of the steered beamformer for source location with the robustness offered by the PHAT weighting, labelled as SRP-PHAT, was first proposed by [Dib00] pp. 80-82 and can be expressed as follows:

$$P(\mathbf{q}) = \sum_{i=1}^{M} \sum_{j=1}^{M} \int_{-\infty}^{\infty} \Phi_{ij}(\omega) X_i(\omega) X_j'(\omega) e^{j\omega(\Delta_j - \Delta_i)} \tag{2.50}$$

where:

- $\Phi_{ij}(\omega) = W_i(\omega) W_j'(\omega) = \dfrac{1}{|X_i(\omega) X_j'(\omega)|} \Leftrightarrow W_n(\omega) = \dfrac{1}{|X_n(\omega)|}$

are the desired SRP-PHAT filters.

- $\tau_{ij} = \Delta_j - \Delta_i$
  is the Time Difference of Arrival (TDOA) between the i-th and the j-th microphones for the sound waves coming from location $\mathbf{q}$.

2.4.3.2 SRP in terms of GCC

This section shows that the SRP of an M elements array is equivalent to the sum of the Generalized Cross Correlations (GCCs) of all the possible combinations of microphone pairs:

$$\binom{M}{2} = \frac{M!}{2! \cdot (M-2)!} = \frac{M(M-1)}{2}$$

This way, the SRP is able to make spatial averaging by integrating the data coming from multiple microphones. Hence, as the number of microphones increase SRP naturally extends the robustness of the GCC method. It is important to note also that the SRP of a 2 elements array will be equivalent to the GCC of these two elements.

If we combine the SRP expression in equation 2.50 with the GCC of two microphone signals in equation 2.31 in page 28, a time-domain version of the Steered Response Power can now be expressed as a function of the Generalized Cross Correlations summation:

$$P(\mathbf{q}) = P(\Delta_1 ... \Delta_M) = 2\pi \sum_{i=1}^{M} \sum_{j=1}^{M} c_{ij}(\Delta_j - \Delta_i) = 2\pi \sum_{i=1}^{M} \sum_{j=1}^{M} c_{ij}(\tau_{ij}) \tag{2.51}$$

where $\Delta_1 ... \Delta_M$ are the appropiate steering delays able to focus the array on location $\mathbf{q}$ and $c_{ij}(\tau_{ij})$ is the GCC-PHAT of the signals from microphones i and j.

This is then the sum of all possible pairwise GCC permutations which are time-shifted by the differences in the steering delays. Included in this sumation is the sum of the M autocorrelations, which is the GCC evaluated at a lag of zero. These terms contribute only a DC offset to the Steered Response Power since they are independent of the steering delays. Also, equation 2.51 includes both permutations of each crossing. However, summing a GCC combination plus its "time-flipped" permutation is equivalent to scaling one permutation by two since the associated differences in the steering delays are opposite for each permutation:

$$c_{ij}(\Delta_j - \Delta_i) = c_{ji}(\Delta_i - \Delta_j) =: c_{ij}(\Delta_j - \Delta_i) + c_{ji}(\Delta_i - \Delta_j) = 2c_{ij}(\Delta_j - \Delta_i)$$

This way, it has been shown that the SRP, within a scale factor and a constant offset, is equivalent to the summation of all possible mic pair GCC combinations (instead of permutations therefore saving computational load by a factor of 2).

2.4.3.3 SRP-PHAT implementation

In this Section we will present the different methods and strategies that were implemented in this Master Thesis to get a working SRP-PHAT algorithm.

As presented in section 2.4.1.3 in page 29 for GCC-PHAT, SRP-PHAT can also be implemented using the same block-processing scheme that employs time-limited, overlapped, windowed DFTs as estimates of the microphone signals spectra: The array signals are segmented in time into short blocks and the steered response algorithm is computed for each one of them. The block DFTs are denoted by $X_{m,b}[k]$ where m is the microphone index and b the block index.

Equation 2.51 defines the steered response as the summation of GCC functions. If we substitute these GCCs by their implementation from equation 2.40 in page 31 we get an estimate of the steered response power at the block b, $\hat{P}_b$:

$$\hat{P}_b(\hat{\Delta}_1 ... \hat{\Delta}_M) = 2\pi \sum_{i=1}^{M} \sum_{j=1}^{M} \hat{c}_{ij,b}(\hat{\tau}_{ij}) = 2\pi \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{ij}[k] X_{i,b}[k] X'_{j,b}[k] e^{jk\frac{2\pi}{K}\hat{\tau}_{ij}} \quad (2.52)$$

And according to equation 2.41 in page 31, we can express 2.52 in FFT terms:

$$\hat{P}_b(\hat{\Delta}_1 ... \hat{\Delta}_M) = \sum_{i=1}^{M} \sum_{j=1}^{M} Re[IFFT(\frac{X_{i,b}[k] X'_{j,b}[k]}{|X_{i,b}[k]||X_{j,b}[k]|})](\hat{\tau}_{ij}) \quad (2.53)$$

Equation 2.52 shows us a time-domain implementation of the Steered Response Power (TSRP) got with a speech block captured by an M elements array when focusing at the spatial location which defines the time delays set $\Delta_1 ... \Delta_M$. However, although these steering delays are continous in equation 2.51 they must be sampled in practice in equation 2.52. This fact introduces an important unaccuracy effect: Due to the discretization process, the summated GCC functions will be evaluated at discrete time lags, in samples, which are slightly shifted with respect to the real TDOAs between the microphone pairs. In Section 2.4.1.3 in page 29 it was explained how the inter-microphone distance, the

sampling frequency and the DOA angle had an influence on this effect. Now, we will here present some of *rounding* techniques applied in this Master Thesis implementation when evaluating the discrete GCC functions. They have been thought so that the information in between the samples of the discrete GCC functions can be predicted based on the sampled values of them that we keep.

- *No rounding*
  $\hat{\tau}_{ij} = ceil(\tau_{ij})$
  where $ceil$ is a function returning the closer integer towards zero. This is the simplest, and the less smart way to choose which time delay in the IFFT-discretized GCC function we must evaluate.

- *Interpolation*
  $\hat{\tau}_{ij} = (1 - \alpha) \cdot ceil(\tau_{ij}) + \alpha \cdot floor(\tau_{ij})$
  where $alpha = |\tau_{ij} - ceil(\tau_{ij})|$ and $floor$ is a function returning the closer integer towards $\pm\infty$. This is a more smart way to act since it tries to predict, by interpolation, the missing continous value based on the weighted information given by the two neighbouring, discretized lags.

- *Rounding*
  $\hat{\tau}_{ij} = round(\tau_{ij})$
  where $round$ is a function simply returning the closest integer. It can be seen as a simpler, more abrupt interpolation technique based on a step-function.

Some experimental results about these rounding techniques can be found in the experimental corpus of this Master Thesis in Section 3.5.1 in page 73.

Hence, the time-domain implementation of the SRP algorithm (TSRP) implies a loss of precision when evaluating the sampled GCC functions. We then searched for an alternative implementation with no loss of accuracy, in other words, a SRP version where we could make use of the real time delays without having to discretize them. Equation 2.50 in page 35 presents a frequency-domain version of the SRP algorithm (FSRP). With this method, the TDOA shifts between two microphones in TSRP are substituted in the frequency domain by multiplications of the GCC functions by complex exponentials evaluated at the proper time lags, with the difference that, this time, we can make use of the real time delays in real units without having to discretize them as it was neccessary in the TSRP implementation.

Experimental results evaluating the performance and computational load of these two different implementations, TSRP and FSRP, can be found in Sections 3.5.1 and 3.8 in pages 73 and 123 respectively. During the experimentation, it was also demonstrated

the theoretical equivalence between the two implementation methods: TSRP and FSRP. FSRP was performed with a forced rounding conversion of the TDOA units as it must be done in TSRP. The results yielded then by the two strategies were totally equivalent.

2.4.4 Additional estrategies

This Master Thesis aimed at improving the results obtained just by applying the baseline SRP-PHAT technique depicted above. It is because of this reason that some new techniques were incorporated to the main algorithm in order to rise its performance. We will depict them in the following sections.

2.4.4.1 Coarse to fine search

As explained in the introduction to SRP algorithms in Section 2.4.3 in page 33, their main drawback consists on their typically high computational cost. In [Var02] pp. 74-75, Varma demonstrates that the SRP-PHAT method requires a number of evaluations several orders of magnitude higher than the GCC-PHAT method when computing a localization estimate.

Recently, new efficient search algorithms have been proposed in order to take advantage of the robustness and accuracy of the SRP-PHAT methods without suffering their computational expense. In this sense, Zotkin and Duraiswami proposed in [RDD01] and [ZD04] a hierarchical search of the Steered Power Response in various levels, from the coarser to the finer one. In order to do so, they make use of the space-frequency relationship of sound. Higher frequencies correspond to small wavelenghts that can explore the space in a finer way, while low frequencies correspond to big wavelenghts that integrate big areas of space. Zotkin and Duraiswami performed some experiments with their array configuration in order to get to a relationship between the spatial energy peak width and the source frequency. The results, depicted in Figure 2.14, threw the following relationship:

$$b \simeq \frac{2\lambda}{5} \; where \; b \; is \; the \; beamformer \; peak \; width. \tag{2.54}$$

From the previous equation and Figure 2.14, we can rapidly infere that using just low-frequency bands of our signal let us explore and integrate the energy coming from wide areas of space (since the peak width consequently defined is wider). Therefore, we will be able to cover the whole search area by just evaluating a few and highly spaced spatial points. On the contrary, as we increase the frequency band that we use in localization tasks, we are able to evaluate regions every time smaller and smaller. This

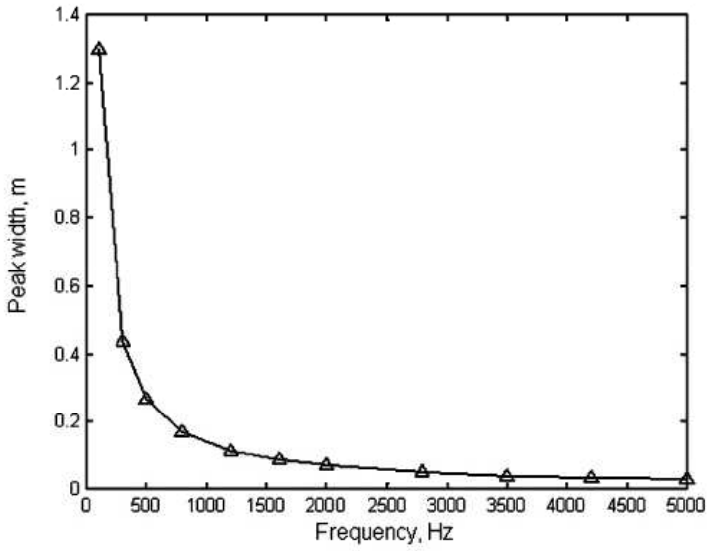

Figure 2.14: Zotkin and Duraiswami beamformer peak width as a function of frequency

way, the evaluation of a high number of spatial locations is not neccessary to get to high accuracy/finesse levels.

Eventually, we decided to add this new technique to our algorithm and evaluate how it worked. In the first step, we can select an appropiate, relatively low cut-off frequency that will define a small number of gross search areas according to the frequency-spatial relatioship showed above. Once we have selected the gross area containing the strongest speech energy, we further divide it into 8 equispaced cubes, called octrees, and explore them themselves with a cut-off frequency twice as big as the previous one (since the areas to explore have become twice as small). We can continue like this until we get to search areas as fine and precise as desired. However, the computational time implied in order to get to these fine levels should not be here a drawback since we are discarding from the beginning large areas of space and just concentrate on accurately exploring those (smaller) ones which are more likely to contain the speaker according to the previous algorithm steps.

The evaluation and experimental results about this technique can be found in the experimental corpus of this Master Thesis in Section 3.9.1 in page 129.

2.4.4.2 Signal to noise ratio considerations

It is also important to take into account some signal to noise ratio considerations as hinted by DiBiase in [Dib00] pp. 28-34. It is advantageous to discard those speech blocks where there are pauses or very low-power speech as they result in poor localization es-

timates. To avoid this effect it is first important to make use of a good Voice Activity Detector (VAD), see Section 2.6 in page 47, that will discard those speechless frames from being processed. Nevertheless, even within speech frames, the noise can sometimes be powerful enough as to corrupt the localization estimates. In this sense, we incorpored DiBiase's idea to our system by implementing SNR masks.

In order to do so in our implementation, we first neet to make an estimate of the noise power present in the recording thanks to a measurement performed during the first, speechless moments of the recording. Based on this information, we will define a power threshold. Whenever any speech sample in the FFT transforms holds a power lying behind this threshold, we will directly not take it into account in our localization calculations. There can be two different kind of thresholds we can make us of:

- *Fixed threshold*, we define it by measuring the noise power present in an initial speechless frame and then averaging it over all frequencies. This fixes the minimum bound to which compare our speech samples, if their value is not a certain dB level above this bound they will not be taken into account.
- *Adaptive threshold*, this threshold is again determined by the initial measurement of a speechless frame, with the difference that, this time, we will store the different noise power values present at each of the frequency components considered. This way, we can form a frequency-dependent noise power mask to which compare the power spectrum of our speech frames. Now, we will only discard a sample in the FFT transform of the frame when its power value lies behind the corresponding noise value located at the same corresponding frequency.

2.4.4.3 Estimation of localization confidence

During the development of this work it was suggested that, in order to improve the algorithm performance, we could somehow try to match known set-up conditions of our system to the confidence on the estimates that it may yield given these conditions. In other words, any kind of a priori information about how the reliability of our estimations is related to the prior experimental conditions can be used to improve our localization results by designing techniques that put more emphasis on those conditions which yield the best results.

There are several initial parameters that can affect the final position estimates: the microphones geometry, the signal frequency bands, the speech spectral content (log energy in bands), the Signal to Noise Ratio (SNR), etc. Measuring the effect each one of them has on our estimations accuracy and designing a scheme that puts the stress on those favorable conditions and "hides" those negative ones is the base of this strategy.

We can think of two main schemes that can be used for this emphasizing strategy:

- *A weighting function*, that can be used to filter the contributions of a certain parameter based on how much reliability we can put on it. Reliable frequency or spectral bands, array configurations, etc. will "weight" more at the moment of deciding an estimate than those which are not favorable a priori.
- *A Neural Network (NN)*, is a much more powerful tool that can be designed according to MuMe, an environment for neural computing described in [Jab94]. The resulting NN can be trained with a set of input parameters, selected because of their influence on the estimates accuracy. After the training phase, the NN sorts out itself the appropiate weighting that it must apply to the different parameters depending on their value so that the system overall estimates are as precise as possible.

For instance, in Section 2.4.1.3 in page 29, it was demonstrated that longer inter-microphone distances imply smaller imprecissions when evaluating the discrete GCC functions. We could then think of a weigthing function to emphasize the contributions coming from those microphone pairs which are more distant. Section 3.9.3 in page 137 presents an evaluation of the experimental results about this insight.

2.4.4.4 Interpolation techniques

As introduced in Section 2.4.1.3 in page 29, when working with digital signals, we must assume that the delay between signals is an integer number of samples. However, this is not often the case and most of the time delays among signals will lie in between integer sample delays. The imprecission commited depends upon a number of factors. Equation 2.42 in page 31 stated that the higher the recording sampling frequency is, the smaller the imprecission committed when evaluating the GCC functions at the incorrect delay.

Nevertheless, the sampling frequency of a recording is usually fixed and there is no possibility to vary it. A technique was suggested by Varma in [Var02] pp. 60-68 in order to achieve sub-sample resolution within the integer delay values in samples of the discrete GCC functions. It consisted on the interpolation of the discrete GCC-PHAT functions by performing frequency domain zero-padding. Examples of this can be found in Figures 2.15 and 2.16. The zero-padding is done in the middle of the DFT and the number of zeros padded is such that the lenght of the DFT is doubled. Notice that zero-padding in the frequency domain is equivalent to interpolation in the time-domain. In effect, by zero-padding the DFT we have decreased the discrete frequency step by a factor of 2, thus artificially increasing the sampling frequency by the same factor.

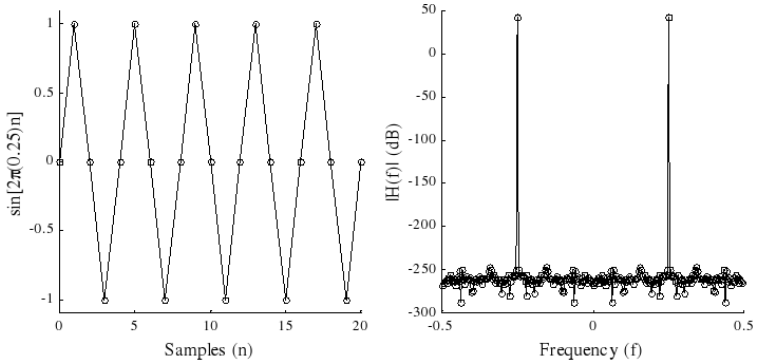

Figure 2.15: Discrete sinusoid (left) and the magnitude of its DFT (right). (Taken from Varma **??**)

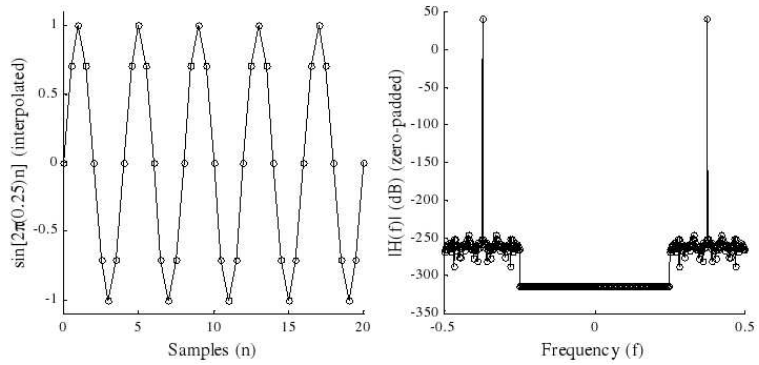

Figure 2.16: Interpolated discrete sinusoid (left) and the magnitude of its zero-padded DFT.(Taken from Varma **??**)

Section 3.5.1 in page 73 presents an evaluation of the experimental results about this technique.

2.4.4.5 Filtering techniques

As suggested in Section 2.4.4.3, it can be interesting to perform the localization algorithm just with certain frequency bands in order to see which effect the different frequency components have on the final localization estimates. With this aim, we will apply different low-pass, high-pass and pass-band filters to check their influence on our algorithm.

This kind of technique was already slighty pointed out by DiBiase in [Dib00] p. 46. He suggested that the use of a bandpass weighting could be used in conjunction with the PHAT prefiltering in order to emphasize those frequency bands where most of the speech energy lies. He also proposed to apply low-pass filtering bellow 300 Hz since most of the noise power lies on these bands. In addition, a low-pass filtering can be advantageous since the large wavelengths of low-frequency waves are not of much use for localization purposes. Nevertheless, it is also important to point out that most of the energy of the human speech concentrates on relatively low frequencies and quickly decreases as the frequency increases. This way, the highest frequency bands, those being more suited for source localization since their small wavelengths can explore the space in a more accurate mood, will contain less speech power and consequently suffer from poor Signal to Noise Ratios. Thus, there is a clear trade-off between the accuracy of the wavelenghts when exploring the search space and their SNR depending on their frequency range.

Section 3.9.4 in page 143 presents an extensive evaluation of the experimental results got with different filtering strategies and assesses about the best configuration in order to get good localization estimates.

2.4.4.6 Use of geometrical information

When trying to improve the localization performance of the system, it is fundamental to take into account any possible a priori information that may help us reduce the uncertainty about the speaker position. In this sense, using the knowledge we have about the environment where the speaker will talk, its geometry, bounds and dead areas, can lead to better localization rates.

We concentrated on two main aspects on this area:

- *Environment bounds.* We have a priori knowledge about the geometry of the room where the speaker will talk and, therefore, we can set the search area to adjust accurately to the room limits. We can still make things in a smarter way if we

tighten these bounds and reduce them to fit just those areas where the speaker effectively speaks, see Section 3.3.1.1 in page 61 for instance. With this strategy we can define a more accurate area where to search and get better localization results. This is, then, a proactive technique.

- *Dead areas*. Within the room we can identify certain places where the speaker is unlikely to be. For instance, it is almost sure that the speaker won't be talking on top of a table or a wardrobe, even though they are included inside the search area. In our localization algorithm, we took into account this fact by implementing a new flag. When activated, the program compares all the localization estimates it outputs to a previously provided list of dead areas. In case any estimate belongs to any of these dead areas we can assume the system has commited an error in its localization and, thus, discard this estimate which will be replaced by the previous, correct position estimate. This is, then, a reactive technique.

Section 3.9.5 in page 152 presents the evaluation of the experimental results got with these strategies.

2.5 Tracking algorithms

The implementation of tracking algorithms can be taken into account as a way to improve the robustness of our localization algorithms. In effect, it is desirable to integrate information coming from different measurements. We already did so during our system design when taking SRP into account: its localization estimates demonstrate to be more robust since this technique integrates the information collected by every possible mic pair combination. It is not the only example, in Section 2.4.4.3 in page 41, it was suggested the idea of integrating the information coming from those mic pairs or frequency bands selected to be more reliable.

Nevertheless, all the systems proposed above based their estimations on the analysis of one single speech frame as seen in Section 2.4.3.3 in page 37. In this context, the use of tracking algorithms can act as an useful tool to integrate the information coming from temporally different speech frames: In effect, tracking can be viewed as the task of filtering the instantaneous localization estimates output by the systems depicted above in order to get a spatially smoothed trajectory. This way, if our system suddenly outputs a wrong estimate which happens to represent an abrupt change in the speaker trajectory, the estimate will automatically be corrected when compared to the previous speaker localizations and filtered so that it resembles more a place where the source is likely to be according to its past behaviour.

There are two main alternatives depicted in literature, see [Pro06], about how to perform this filtering:

- *The Kalman filter*
- *Particle Filtering (PF)*

2.5.1 The Kalman filter

The *Linear Kalman Filter* is based on the comparison of the incoming measurements with the ongoing estimates in order to recursively produce estimates of the system estate. The gains, or weights, applied to the input data are dependent on important factors such as the measurement accuracy and the object motion. An exhaustive introduction to the Kalman filter can be found in [Kal60]. If the models are accurate, the resulting state estimates are optimal in the mean square sense. A simple implementation of the Kalman filter for tracking can be found at [CSH06] where the final localization estimates are obtained based on a SRP-PHAT method that weights, by an adaptive smoothing factor given by the Kalman predictor, the Cross-Power Spectrum (CSP) information of the actual and the previous speech frames.

Nevertheless, the linear Kalman filter models the dynamics to be linear and Gaussian. Consequently, it cannot be used for measurements which are nonlinearly related to the system state such as those resulting from spontaneous speech which can be both highly changing in space (sharp turns, speaker changes) and sporadic over time (short utterances).

The *Extended Kalman Filter (EKF)* was introduced in order to solve this problem by a first linearization step that accomodates the non-linear measurement equations to be linear functions of the current state, see [BW01] pp. 210-212. Nevertheless, the tuning of the EKF parameters demonstrated to be very difficult to tune in practice, see [JU97].

More recently, the *Unscented Kalman Filter (UKF)* was proposed to avoid this linearization step and accomodate non-Gaussian measurements such as noise sources, see [DG05].

Apart from integrating the information coming from several speech frames, it was also proposed, see [BW01] pp. 203-225, to make use of different estimates coming from different localization systems based on different sensors. In particular, it was suggested the use of a joint audio-video tracking system. A microphone array would output a localization estimate based on the methods depicted in this Master Thesis. At the same time, a video camara would obtain its own source position estimate. Since each method has its specific strengths and weaknesses, it is wise to integrate their measurements via a

Decentralized Kalman Filter (DKF) in order to get to a more robust localization system.

2.5.2 Particle Filtering (PF)

As an alternative to the Kalman filter, Sequential Monte-Carlo (SMC) methods, also known as Particle Filtering (PF), were proposed. They approximate the optimal Bayesian filter by representing probability distributions through a set of particles. A PF recursively approximates the filtering distribution of states given observations by predicting candidate configurations and measuring their likelihood, see [Leh04]. Applications to single acoustic sources have been implemented, [DBWW03]. However, its use in multi-speaker environments is still problematic due to the changing turns and number of active speakers.

2.6 Voice activity detection

As pointed out in the CHIL consortium evaluation lines, see [OM06] p. 46, errors in the final estimates of a localization system can be attributed either to inaccuracy of the localization system or to a failure of the prior Voice Activity Detector (VAD). In effect, a VAD is a required component of an acoustic source localizer since no localization can ever be performed if the speaker is silent.

The importance of a good speech/silence detection was also hinted by DiBiase in [Dib00] pp. 31-33 since it allows us to discard those speech blocks with pauses or very low-power speech. All in all, the use of an accurate VAD as a prior step to any localization system turns out to be crucial so that the position estimates are limited just to those periods of time when there is a speaker effectively speaking. There are two ways a VAD can fail, both of them decreasing our system performance:

- *Silence period marked as speech*. It makes our localization system to try to estimate a source position while no source is in fact active. Consequently, it will result in an error that will make the overall localization rate drop. This type of error is also known as *False Alarm* in the CHIL nomeclature.
- *Speech period marked as silence*. It makes our localization system discard periods of time where there is in fact an active speaker. Consequently, no estimate will be output for these periods. This type of error is also known as *Deletion* in the CHIL nomenclature.

The traditional techniques for a VAD design have been usually based on the signal

energy. In [Com90], adaptive energy thresholds were applied to the output of a microphone array delay-and-sum beamformer in order to identify the speech boundaries. However, in [LAS03], a new method for VAD design was proposed based on the Signal Cross-Power Spectrum (CSP). This scheme is advantageous since it allows to use the same technique (CSP), in the first step, for the speech activity detector, and later, for the talker location itself (it is important to remind that both GCC and SRP localization methods are based upon the CSP analysis, see Section 2.4.1 in page 25).

These techniques were taken into account from the proposal of this Master Thesis. Nevertheless, finally, an already availabe free software tool, the Qualcomm-ICSI-OGI front-end (QIO), was used for VAD purposes in this Master Thesis. This choice was based on the evaluation and paralell work developed by Francisco Jose Royo in the Speech Technology Group at Technical University of Madrid (UPM). QIO is a feature extraction algorithm developed by ICSI (International Computer Science Institute) in Berkeley (California), OGI (Oregon Graduate Institute of Science and Technology) and Qualcomm Inc. As it can be seen in Figure 2.17, the tool goes through a high number of different steps. However, the QIO front-end allows to make use of just the Voice Activity Detection (VAD) block if desired. This block is composed by a feed-forward Neural Network trained to discriminate between speech and non-speech frames using a back-propagation algorithm based on the speech low-passed filtered log energies. A complete description about this scheme implementation details can be found in [qio02]. Based on its decisions the frames in an audio file are labelled with a binary flag: "0" in case they are judged to be non-speech and "1" when the opposite.

2.7 Summary

This theoretical introduction has dealed with the characteristic issues of the acoustic environments in which microphones happen to be distant from the sources of interest, such as a conference room or a digital home. In these environments, when propagating, the sound waves suffer not only from the natural amplitude attenuation, but also from high noise and reverberation. A description of the sound propagation issues under these environments can be found in the first Section of this chapter. Under these conditions, the captured signal results to be difficult to understand by the later speech recognition systems.

The use of microphone arrays demonstrate to be an useful way to avoid these negative effects thanks to their ability to perform spatial filtering. That is, they enhance the signal coming from a desired direction while rejecting all the others. This effect is achieved by beamforming, a technique that allows to steer the array to the chosen direction.

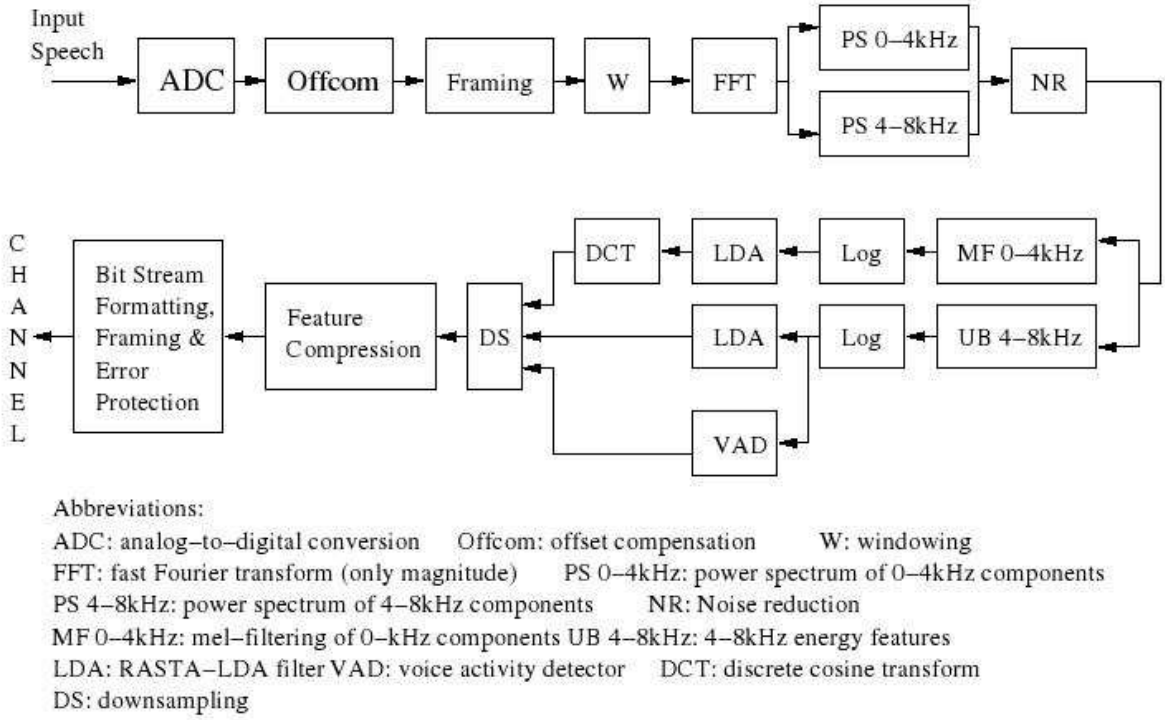

Figure 2.17: QIO block diagram

A description about microphone arrays properties and beamforming can be found in the following Sections of the chapter.

With this basis, it is straight-forward to see the necessity of designing and implementing a source localization system in order to get the appropiate position we must steer our array to. This Master Thesis focuses on this key issue and presents three possible methods for speaker localization: based on Time Delay Estimation (TDE), based on the Steered Response Power (SRP) and based on Spectral-Estimation Sub-Spaces. This last method demonstrates not to be suited for wide-band signals such as speech. It is because of this that this Master Thesis mainly concentrates on the design and implementation of the two first methods whose characteristics and implementation schemes are described in the following Sections of the chapter. Both of them output position estimates based on the analysis of the CSP function (which is the Fourier transform of the GCC function). TDE methods are defined as indirect methods since they require two steps for localization: First, a TDOA computation is done between the signals of a microphone pair based on their GCC/CSP function. Second, the source location itself is performed based on this TDOA value and the environment geometrical properties. These methods have the advantage of their simplicity and low computational load. On the other hand, SRP methods are defined as direct methods since they directly evaluate the array response power

when steered to different locations and take the maxima as the more likely estimate for the source position. In this chapter it was demonstrated how SRP methods are equivalent to the summation of all possible combinations of microphone pairs. This technique therefore benefits from the integration of spatially differenciated measurements and offers more accurate and robust results at the expense of heavier computational processes. The chapter also discusses the use of different prefilters prior to CSP computations among which the Phase Transform (PHAT) prewhitening filter was chosen to output the best results.

In the later Sections of the chapter some implementation details are discussed. Specially conflictive is the necessity to transform real time delay units in seconds to integer time delay units in samples when performing the digital implementation of the algorithms. It was seen how this transformation implies an unaccuracy effect that can be corrected either with different rounding techniques, when implementing the system through time-domain GCC evaluations (TSRP) or by directly evaluating the frequency-domain CSP functions (FSRP).

In the next Sections some possible techniques that may help improve the basic SRP algorithm performance are presented. A hyerarchical, *coarse to fine* search was implemented as a way to decrease the inherent computational load associated to SRP searchs. *Noise masking* was also implemented in order to discard speech frames with low SNR. A study about the *confidence* we can have on the final estimates given the system known inputs is carried out in order to exploit those system set-up conditions that can be more advantageous. *Interpolation* is also taken into account as a way to reach to sub-sample resolution in the digital computations. Low-pass, high-pass and band-pass *filtering* was implemented in order to check the contributions of the different frequency bands. Finally, considerations about the *geometrical information* were taken in order to discard those dead areas in the search space.

Later on, an introduction about some *tracking algorithms* is presented. They offer promising results since they are able to integrate temporally differenciated measurements: The instantaneous spatial estimates from different speech frames are filtered in order to get a smooth trajectory. Kalman filter and Particle Filtering (PF) methods are introduced. Although their design was considered, the design, implementation and exhaustive evaluation itself of the GCC-PHAT and SRP-PHAT algorithms and their improvement techniques left no room for the implementation and testing of these tracking algorithms. Their development is open for future research works continuing this present Master Thesis.

Finally, an introduction about VAD systems is presented. They are a fundamental first step in every localization system since no location estimate can be performed during

the silent periods of an utterance. An accurate distinction between speech and silence is therefore crucial for a good performance in our system. The main methods to design a VAD, based on energy and based on CSP analysis, are introduced. Finally, the chosen solution used in this Master Thesis was an already available free software tool known as QIO. The motivation and details for this solution are explained in the last Section of the chapter.

Chapter 3

Experimental results

3.1 Introduction

This chapter is aimed at evaluating the techniques introduced in the theoretical review. A whole different range of experiments, done under different conditions and applied to different databases, will be shown in order to analize their results and get to certain conclusions about the algorithms performance.

3.2 Evaluation strategy

As described in the theoretical review, the SRP algorithm is the most powerful one in terms of speaker localization performance. Therefore, the main experimental corpus of this thesis will be aimed at testing this technique under different scenarios in order to analyze those key features and strategies considered relevant to characterize the algorithm behaviour.

In order to be able to properly interpret the performance and compare all the different results from all the different experiments, we need to stick to a certain evaluation strategy.

3.2.1 Main Evaluation Metrics: The CHIL Evaluation Plan

In particular we have decided to mainly follow the CHIL Evaluation Plan, see [OM06]. The CHIL team is a consortium of internationally renowned research labs in Europe and the US. Using their Evaluation Plan will allow us not only to have an unified, standarized view of all our results but also to be able to directly compare the performance of our algorithm versus those techniques currently operating in the international research scene.

In particular we will make use of their audio technology metrics for acoustic person tracking. According to this basis, our localization algorithm must yield a set of spatial (x,y,z) coordinates related to the speaker position estimate every time frame. These position estimates will be compared, by means of the Euclidean distance, to the ones labelled in a transcription file containing the real positions, or ground truth, of the speaker.

In this CHIL Evaluation Plan, the localization errors are classified in two classes:

- *Gross Errors.*
- *Fine Errors.*

Whenever our estimation lies less than a certain threshold to the ground truth position of the speaker we will consider it as a Fine Error. This threshold is set to be 500 mm in an accurate lecture scenario.

A complete description of the CHIL Evaluation strategies can be found at [OM06]. There are two mainly distinct set of metrics. The first set evaluates the person tracking abilities of the system while the second one focuses on more specific metrics for acoustic source localizations.

Among this second set, the most relevant metrics used in this Master Thesis are:

- *Pcor* or localization rate, it provides the number of fine errors over the total number of frames for which the localization system has produced an estimation.
- *Bias*, it provides the mean distance in mm along each of the 3D coordinates (x,y,z) between the ground truth position of the speaker and the system estimation. It can be referred just to the fine error estimates or to both fine plus gross error estimations.
- *Average error*, it provides the mean distance in mm between the estimated positions and the real positions of the speaker. It can be again be referred to either just fine errors or to fine plus gross error estimates. When referred to just fine errors, this metric is equivalent to the MOTP.
- *RMSE*, it provides the RMSE (Root Mean Square Error) of the location estimates in case of both fine and fine plus gross error estimates.
- *False Alarm rate*, it provides the number of false alarms divided by the number of total frames. A false alarm occurs whenever the localization system yields an estimate given that no speaker was in fact active at that moment. False alarms are generally due to errors in the VAD (Voice Activity Detector).

- *Deletion rate*, it provides the number of deletions divided by the total number of frames. A deletion occurs whenever the localization system does not output an estimate given that a speaker is in fact active at that moment. Deletions are generally due to errors in the VAD (Voice Activity Detector).

On the other hand, among the first set, the most relevant metrics used in this Master Thesis are:

- *MOTP*, Multiple Object Tracking Precision, represents the precision of the system, in mm, when it comes to determine the exact position of a tracked person in the room. It is calculated as the total Euclidian distance error between fine estimates and ground truths averaged by the total number of fine errors. It shows the ability of the tracker to just find correct positions and it is independent of tracking errors.
- *A-MOTA*, Audiovisual Multiple Object Tracking Accuracy, represents the tracking accuracy of the system, in %, when it comes to keep correct correspondences over time between the speaker estimation and its ground truth. It is calculated as the sum of all gross errors plus all the deletions averaged by the total number of ground truth points:

$$A - MOTA = 1 - \frac{gross\_errors + deletions}{ground\_truths} \tag{3.1}$$

3.2.2 Other Evaluation Metrics

Apart from the CHIL metrics we have also taken into account some other measurements.

A system to convert cartesian (x,y,z) coordinates into polar (r, azimuth, elevation) was developed. Applying this transformation to both the SRP estimates and the speaker ground truth positions allowed us to ouput an error in terms of azimuth and elevation degrees. This metric is relevant since it permits a direct comparison of our results to those obtained by Lathoud when working with the AV16.3 corpus [Lat06b].

Also, distance error along the z-coordinate seems to be less critical and more difficult to derive in an accurate way (specially when working with linear arrays lined along the XY plane, since their directivity pattern is symmetrical along the z-coordinate). Therefore, localization system performance will be evaluated not only in 3D, (x,y,z) coordinates, but also in 2D, (x,y) coordinates.

3.2.3 Sample Results Table

In order to properly compare the performance of the different experiments carried out, most of the experimental results in this Master Thesis are displayed through a "standard

results table“ showing and comparing the key evaluation parameters presented above.

	Experiment 1	Experiment 2	Experiment 3
Pcor	$56.0 \pm 1.0\%$	$76.0 \pm 0.9\%$	$86.0 \pm 0.7\%$
Rel. error reduction		35.7%	53.6%
Bias fine (x:y:z) [mm]	$-6 : -57 : -99$	$3 : -26 : -64$	$-8 : -31 : -61$
Bias fine+gross (x,y,z) [mm]	$25 : -189 : -106$	$48 : 119 : -73$	$22 : 21 : -68$
Bias AEE fine [mm] = MOTP	230	209	206
Rel. AEE reduction		9.1%	10.4%
Bias fine+gross [mm]	585	503	408
Rel. BIAS f+g reduction		14.0%	30.3%
A-MOTA	$11 \pm 0.6\%$	$51 \pm 1.0\%$	$71 \pm 0.9\%$
Rel. error reduction		363.6%	545.5%
Loc. frames	9390	9390	9390
Ref. duration (s)	496.0	496.0	496.0

Table 3.1: Instance of the standard results table used in this Master Thesis

As shown in Table 3.1, the ”standard results table“ presents several columns, each of them holding the evaluation parameters considered in different rows. We now offer a brief description of its contents:

- *CHIL evaluation metrics*

  Pcor, or localization rate, represents the experiment fine error rate in %.

  Bias fine (x:y:z), represents the average distance error in mm commited along each of the three spatial coordinates (x,y,z) when taking into account just those frames with fine errors.

  Bias fine+gros (x:y:z), represents the average distance error in mm commited along each of the three spatial coordinates (x,y,z) when taking into account all the analyzed frames, both with fine and with gross errors.

  Bias AEE fine = MOTP, represents the average total distance error in mm between the estimated position and the ground truth location of the speaker when taking into account just those frames with fine errors.

  Bias fine+gross, represents the average total distance error in mm between the estimated position and the ground truth location of the speaker when taking into account all the analyzed frames, both with fine and with gross errors.

  A-MOTA, represents the tracking accuracy of the system in %.

- *Loc. frames*, represents the total number of frames taken into account during the experiment considered and gives a clue about the relevancy of the results obtained.
- *Ref. duration*, represents the total length, in seconds, of the speech considered in the experiment, that is to say, it measures the total duration of those fragments marked by the Voice Activity Detector (VAD) to contain speech out of the set of audio files involved in the experiment.
- *Relative reduction*, appearing in scriptsize at the columns of the Pcor, Bias fine, Bias fine+gross and A-MOTA, it represents in the improvements (positive magnitudes) or degradations (negative magnitudes) of the considered parameters in % when compared in relative terms to the experiment placed at the first column.
- *Statistical reliability*, appearing next to the values hold in the Pcor and A-MOTA rows and preceded by a $\pm$ symbol, it represents a measure of the statistical relevance of the figure obtained in each parameter, that is to say, it tells to us how much we can rely on the experimental value to be an accurate one. In other words, this parameter sets a probability range within which we are sure, up to a certain degree, that the parameter considered, either the Pcor or the A-MOTA, will lie regardless of how many other new experiments or audio files are considered. We compute this statistical reliability according to the next formula [Lew07]:

$$margin[\%] = \alpha\sqrt{\frac{P(1-P)}{N}} \tag{3.2}$$

where:

P is the probability value, in %, of the parameter considered.

N is the total number of elements, frames in our particular case, out of which the probability value P has been computed.

α is a parameter setting the statistical confindence of the resulting probability range, $P \pm margin$:

$\alpha = 2.58 =:99\%$ confidence.

$\alpha = 1.96 =:95\%$ confidence. This is the value used in this Master Thesis.

$\alpha = 1.64 =:90\%$ confidence.

3.2.4 Tunable parameters

The set of experiments was thought to cover a range of parameters and situations as wide as possible. In order to do so, the software developed was designed so that the

fundamental features of the SRP algorithm could be easily modified. Next, we offer a list and brief description of those key parameters that were selected so that their influence on the system performance could be measured under different values and conditions:

- *Sampling frequency,* f_s, it is the rate at which the analog speech signal was sampled to be turned into a digital audio file. Its value it is fixed and determined by the speech acquisition hardware used during the recordings of each database, although it is always possible to downsample the given audio files in order to get some new ones at a lower sampling frequency. The details about the different sampling frequencies used at each of the different databases considered can be found at Section 3.3 in page 61. Experimental evaluation about different sampling frequencies is performed in Section 3.5.1 in page 73.

- *Frame size*, it is the number of samples contained by each of the single speech fragments in which the audio file is divided. All the computations leading to a localization estimate are individually done taking into account just the information contained at each of these fragments. Some results about the effects of the frame size can be found in Section 3.5.2 in page 80.

- *Frame shift*, it sets the rate at which localization estimates are output. This rate was set to be 40 ms in this Master Thesis as done in the Idiap Project. Therefore, every frame shift seconds (i.e. every 40 ms) a new fragment of ”frame size“ samples is created and analyzed. This ”frame size“ samples are constituted by ”frame shift“ new samples coming from the audio file plus "(frame size - frame shift)" samples coming from the previous analyzed fragment.

- *FFT size*, it sets the length of the FFTs performed during the algorithm computations. For convenience and performance reasons it is always set to be a multiple of 2 as described in [FJ06] pp. 27 and must be always greater or equal than the frame size chosen. More details about the effects and limitations of the FFT size can be found in Section 3.5.3 in page 88.

- *Window type*, it sets the smoothing function that will be applied to the frames considered before its computations. The different possible windowing functions considered are: Rectangular, Hamming, Hanning, Bartlett and Blackman. More details about them can be found at appendix B and experimental results in Section 3.5.4 in page 90.

- *Number of maximums*, it sets how many peaks out of the correlation function will be selected and stored for further processing. The algorithm employed at this Master Thesis only uses the first, most powerful peak as the most likely to indicate the

correct delay between signals captured by different microphones. Nevertheless, as pointed in [Var02] pp. 4-5, the primary peak can often result from distorted data while secondary peaks may contain the proper time delay candidate. Although not developed in this Master Thesis, the tuning of this parameter can lead to an improved version of our algorithm that could implement the TIDES (TIme DElay Selection) scheme proposed by Varma in [Var02] pp. 81-122.

- *Correlation type*, it determines which kind of correlation will be used during the algorithm computations: either a time-domain correlation, or a frequency-domain one (theoretically equivalent to the first one), or a frequency-domain one preceded by a whitening filter, typically a PHAT (PHAse Transform) one, see [Dib00] pp. 73-85.

- *Start and end of the audio file*, it determines at which point of the audio file, in seconds, will our algorithm start and end its computations. It can be useful to get just through certain parts of interest in the audio file.

- *Sound speed and temperature*, it allows to choose either the room temperature, T, fact that determines the sound speed in that media as explained in Appendix A, or the sound speed itself. Using an accurate sound speed demonstrates to be of great importance since it is responsible of turning distance units in meters to delay units in seconds according to the equation $distance = sound\_speed(T).time\_delay$. As we can appreciate in Figure 3.1, sound speed varies with temperature and choosing a wrong value can mislead our results.

- *Source mics file*, it contains all the audio filenames involved in a certain simulation altogether with the directory paths where to find them.

- *Simulation file*, it contains all the details depicting the specific environment in which the simulation was performed, specifically, the geometric bounds of the room and the microphone array configuration that captured the signal.

  Microphone array, there is a complete flexibility to define different array configurations. We can determine the number of microphones involved as well as their spatial coordinates within the room considered. Different array configurations can be tested and their results checked in Section 3.6 in page 108.

  Search space grid and bounds, there is also a complete flexibility to define which particular area within the room will be examined in search of the speaker. We can either search the whole room or concentrate on particular part of it. Moreover, we can also define how fine we want our search to be. Typically we apply a grid to the search area. The intersecting points of this grid will constitute the set over which the Steered Response Power (SRP) search will be performed. The distance separating

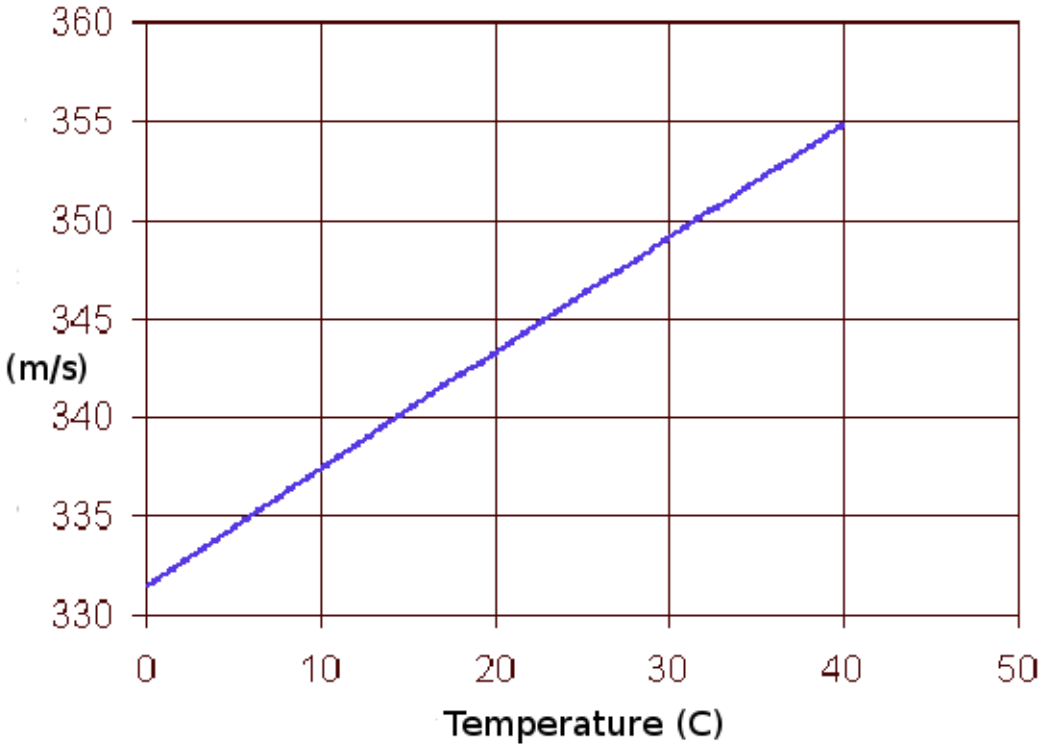

Figure 3.1: Sound speed as a function of temperature

the points in the grid is totally tunable in each of the three axis (x,y,z) (typically it will be the same along the three directions) and can vary from very short distances (i.e. 10 mm) to grosser ones (i.e. 300 mm).

- *Coarse to fine flag*, when activated, this flag makes the algorithm perform its search not in the typicall way but in a hyerarchical mood descending from a coarse level to finer and finer ones. More details about this technique are given in the theoretical introduction in Section 2.4.4.1 in page 39 and at the works by Zotkin and Duraiswami [RDD01] and [ZD04]. Experimental results can be found at Section 3.9.1

  Starting frequency, it determines the starting cut-off frequency that will be applied in the first step of the coarse to fine search according to the given spatial-frequency relationship, see Section 3.9.1. This parameter will also set how big the explored areas will be in the coarser level of the search.

  Maximum distance threshold, it determines how fine we want our coarse to fine search to get. The hyerarchical scheme will descend up to that point in which the distance separating two search points is lower than the threshold defined by this parameter.

- *Interpolation flag*, when activated, the system performs an interpolation scheme in search of improved results as those suggested by Varma at [Var02] pp. 60-68 and

explained in Section 2.4.4.4 in page 42. Experimental results can be found in Section 3.5.1 in page 73.

Interpolation rate, it sets the rate at which the interpolation will be done. Typically we will apply x2 or x3 interpolation to 16 KHz sampled databases in order to try to achieve performances resembling those of 32 KHz and 48 KHz sampled databases respectively.

- *Filter flag*, when activated, the system performs either a low-pass, or a high-pass or a band-pass filtering to the signal prior to any further processing. See theoretical background in Section 2.4.4.5 in page 44. Experimental results can be found in Section 3.9.4 in page 143.

  Low frequency, it sets the lowest frequency that won't be removed during the filtering proccess.

  High frequency, it sets the highest frequency that won't be removed during the filtering proccess. Obviously, this high frequency parameter cannot go further than $\frac{f_s}{2}$.

- *Frequency SRP flag*, this flag determines whether the SRP computations are performed in the frequency domain (FSRP), thus achieving sub-sample resolution, see 2.4.3.3 in page 37, or in the time domain (TSRP), then necessarily loosing precision due to the transformation from real time delay units in seconds to integer units in samples.

  Round flag, it just applies when choosing time domain SRP (TSRP) computations. In that case, we can elect among different ways of turning these real time delay units in seconds to integer units in samples. This flag determines whether to pick the closer integer towards zero, or just the closer integer or perform a linear interpolation between the two closest integer values. Each of these choices imply different results as depicted in Section 3.5.1 at page 73.

- *Noise masking flag*, when activated, the system performs a scheme according to which those speech samples having whose power lies under a certain threshold over the noise background are removed as described by DiBiase at [Dib00] pp. 31-33 and explained in Section 2.4.4.2 in page 40.

  Noise mask type, it sets the discarding strategy to follow: As level of reference we can take either the average noise power along all frequencies or the noise spectrum itself in order to directly compare frequency index by frequency index to the speech power spectrum. Experimental results can be found at 3.9.2 in page 133.

  Noise threshold, it sets how many dB must the speech signal overpass the selected noise power reference in order not to be discarded.

- *Dead areas flag*, when activated, every localization estimate output by the system lying within the limits of any of the dead areas (i.e. tables, wardrobes, etc.) defined in the *simulation file* is discarded and replaced by the last given localization estimate. For theoretical details see Section 2.4.4.6 in page 44. For experimental results check Section 3.9.5 in page 152.
- *Microphone distance weighting flag*, when activated, those results provided by microphone pairs further located each from another are priorized by applying a weighting proportional to their separation. More details can be found in Section 2.4.4.3 in page 41. Experimental results can be found in Section 3.9.3 in page 137.

3.3 Databases

We have used four different databases. Two of them were recorded in real life environments: AV16.3 and HIFI-MM1 and another two are simulated databases: Sony and Simulated HIFI-MM1.

3.3.1 IDIAP AV16.3

An audio-visual database recorded in the IDIAP research institute, Switzerland, with the aid of 16 microphones and 3 cameras, hence the name AV16.3. A complete description of this corpus can be found in [GLGP04]. This Master Thesis only makes use of the audio corpus which was recorded at a sampling frequency of 16KHz.

3.3.1.1 Geometry

For all recordings there are two circular arrays of radius 0.1m composed by 8 sennheiser microphones each. The centers of the two arrays are separated by 0.8m and the origin of coordinates is located in the middle point between the two arrays.

The IDIAP Meeting Room consists on a 8.2mx3.6mx2.4m rectangular room containing a centrally located 4.8mx1.2m rectangular table. Possible speakers' localizations distribute along a L-shaped area around the table as seen in Figure 3.3. A general description of the meeting room, depicted in Figure 3.2, can be found in [Moo02].

3.3.1.2 Contents

The complete database along with the corresponding annotation files containing the recordings ground truth is fully accesible online at [Lat06a].

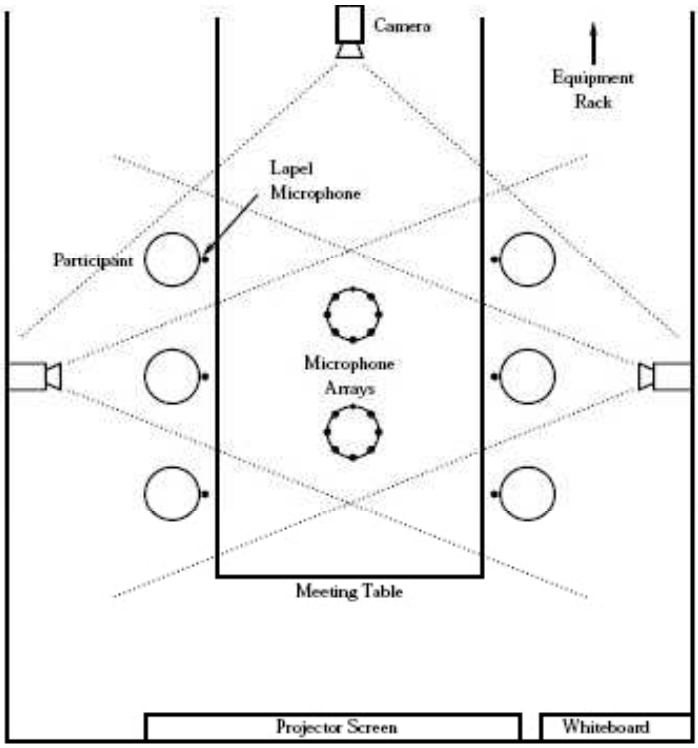

Figure 3.2: Idiap Smart Meeting Room

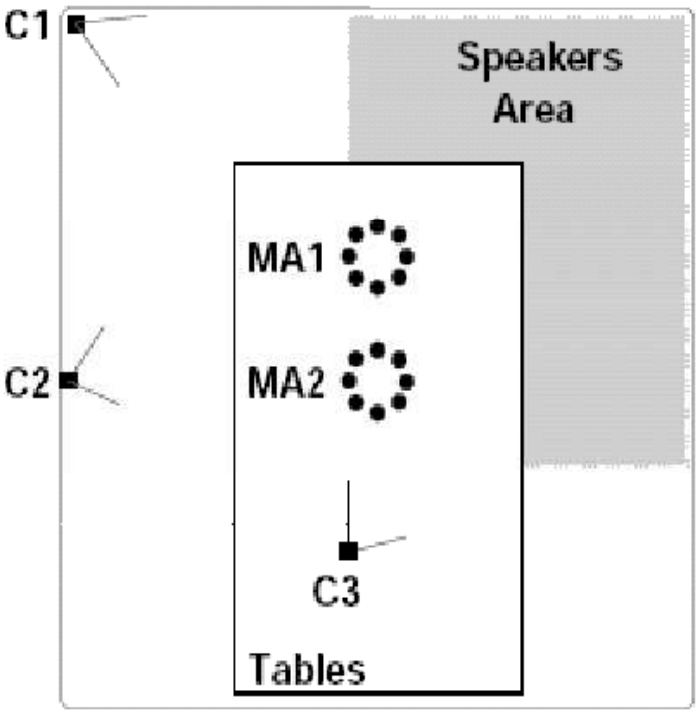

Figure 3.3: 3 m-long by 2 m-wide L-shaped area for speakers distribution in Idiap Room

sequence name	duration(s)	number of speakers	speaker behaviour
$seq01-1p-0000$	213	1	ST
$seq02-1p-0000$	188	1	ST
$seq03-1p-0000$	241	1	ST
$seq11-1p-0100$	32	1	MV
$seq15-1p-0100$	36	1	MV

Table 3.2: List of the annotated sequences. Tags mean: [ST]Static speaker, [MV]Moving speaker

The corpus is composed by several sequences or recordings which range in the number of speakers involved. However, in this Master Thesis we will just focus on those sequences containing a single speaker, those listed in the Table 3.2.

The sequences names are coded sistematically in order to offer a compact description of their contents. For example, "seq15-1p-0100" has three parts:

- "seq15" is the unique identifier of this sequence.
- "1p" means that just one speaker was recorded.
- "0100" four binary flags giving a quick overview of the content of this recording. From left to right:

  bit 1, 0 means "very constrained" (for instance, speaker facing the microphone array at all times), 1 means "mostly unconstrained".

  bit 2, 0 means "static motion", 1 means "dynamic motion".

  bit 3, 0 means "minor occlusions", 1 means "at least one mayor occlusion".

  bit 4, 0 means "little overlap", 1 means "significant overlap" between speakers.

3.3.1.3 Annotation

Every audio sequence is assigned a corresponding annotation file containing the real position (3D coordinates) of the speaker's mouth at every time frame in which that speaker was talking. The segmentation between the speech and speechless periods, that is to say the Voice Activity Detector (VAD) involved, was first checked manually at certain time instances by a human operator in order to ensure its corretness, and later extended to cover the rest of recording time by means of interpolation techniques. The frame time resolution was defined to be 40ms.

3.3.2 HIFI-MM1

An audio database was recorded at the EDECAN Room in the Speech Technology Group (ETSIT, UPM) at the same time this Master Thesis was being developed and closely related to it. The recording sampling frequency was set to be 48KHz and samples are 24 bits signed integers. Additionally, a downsampled 16KHz version of it was also generated in order to compare the performance of our algorithms at different sampling frequencies. It is composed by 1200 utterances, uttered by 12 different speakers (8 males and 4 females). The actual length of the corpus is around 35 minutes of continous speech.

3.3.2.1 Geometry

All the recordings were captured by two different microphone arrays. First, a linear array of four 200 mm equispaced sennheiser microphones located at the front wall of the room and placed simetrically around the origin of coordinates and, then, a L-shaped array of 3 crown microphones located at the corner to the left of the linear array.

This configuration allows us to perform experiments taking into account different combinations of arrays:

- 4 sennheiser + 3 crown
- 4 sennheiser
- 3 crown
- 3 sennheiser
- 2 sennheiser
- 2 crown

The Edecan Room consists on a rectangular 4mx4.4mx3m area depicted in Figure 3.4. During each recording speakers were only allowed to stay at one of five different fixed positions distributed around the room. In each of these fixed positions the speakers were allowed to orientate in four, equally spaced, different directions: from totally facing the array to totally backing it.

3.3.2.2 Contents

A total number of twelve different speakers were recorded. In each recording, there is always just one sole speaker who utters a single, simple, continous sentence. All the spea-

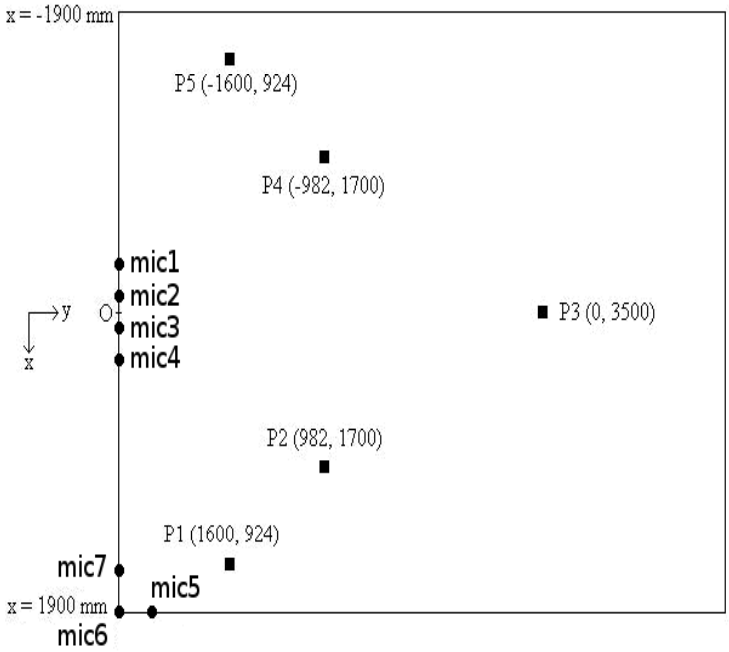

Figure 3.4: The Edecan Project Room sited at the Speech Technology Group (GTH) in the Technical University of Madrid (UPM)

kers were recorded at every possible position with every possible orientation towards the array.

The audio files were systematically coded in order to offer a compact description of their contents. Every file is named:

WWW-PX-OY-UZZUUU-chV.wav where:

- WWW are the speaker initials (speaker id).
- X is the speaker position (1 to 5).
- Y is the speaker orientation towards the array (1 to 4).
- ZZ is the scenario id.
- UUU is the utterance id.
- V is the channel number (0 = close-talk microphone, 1-4 = sennheiser microphones, 5-7 = crown microphones).

3.3.2.3 Annotation

All audio files were generated their corresponding reference files containing the real speakers' "mouth" localizations for every time frame given the actual position P1-5 he was occuping as well as his particular height in each case. All these labelling file follow the format used in the CHIL 2006 evaluation campaign for reference files in the Acoustic Person Tracking Task, see [OM06], namely, one line per time index with the following format:

Time_index(s) Num_simult_speakers Num_noise_sources Speake_ID X_pos Y_pos Z_pos

In the HIFI-MM1 corpus, *Num_simult_speakers* always equals 1, *Num_noise_sources* always equals 0 and *Speaker_ID* equals the corresponding speaker initials.

Of course, as corresponding to a speaker localization sytem, this labelling info was just generated to those periods of time within the audio file in which the speaker is actually talking and never to the silent ones. This segmentation was carried out thanks to the Voice Activiy Detector (VAD) software developed by ICSI (International Computer Science Institute) at Berkely University, California, OGI (Oregon Graduate Institute of Science and Technology) and Qualcomm Inc., [vLK07]. Out of the segments obtained with this software for each of our HIFI audio files, we just picked those forming a long,

continuous, consistent fragment since we have the a priori knowledge that all our recordings consisted basically on short command sentences uttered without interruptions.

3.3.3 Simulated SONY

An audio database recorded for the Sony company is currently available for research purposes. It contains more than 30 hours of clean speech, sampled at frequency of 48KHz with 16 bits resolution and recorded in a studio by a close-talk microphone.

Thanks to some of the applications developed within the Speech Technology Group at Technical University of Madrid (UPM), see [RL06], those audio files were processed in order to simulate they were in fact recorded at the Edecan Room and captured by different configuration of linear arrays.

3.3.3.1 Geometry

The recordings were simulated as if they would have been uttered by a speaker 1.75 meters tall located at the different positions specified in the Edecan Room depicted in Figure 3.4.

Two different possible array configurations were simulated:

- 4 sennheiser microphones + 3 crown microphones. It is the same array configuration used for the HIFI database recordings, see Figure 3.4. It allows us to use different subarray options: linear arrays of 4, 3 or 2 sennheiser microphones, a L-shaped array of 3 crown microphones and linear arrays of 2 crown microphones.
- 33 sennheiser microphones. A linear, 20 mm equispaced array. Out of these 33 microphones we can pick 11 microphones separated by a variable distance forming an harmonic linear array depicted in Figure 3.5. This configuration allows to form different equispaced linear subarrays of 5 microphones:

  Microphones 3, 4, 5, 6 and 7 form a subarray of elements equispaced 20 mm.

  Microphones 2, 3, 5, 7 and 8 form a subarray of elements equispaced 40 mm.

  Microphones 1, 2, 5, 8 and 9 form a subarray of elements equispaced 80 mm.

  Microphones 0, 1, 5, 9 and 10 form a subarray of elements equispaced 160 mm.

  Apart from these configurations there are many other possible subarrays than can be got out of this equispaced 33 elements set. For instance, uniform linear arrays of 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29 and 31 elements equispaced 20 mm could be also picked.

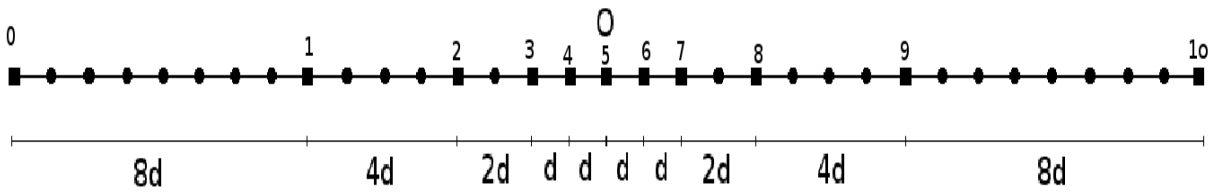

Figure 3.5: Microphone array configuration for Simulated HIFI and Simulated Sony databases. 33 equispaced linear array (circles) and 11 harmonic array (squares). d = 20 mm

3.3.3.2 Contents

208 different speakers were used during the recordings getting a final total amount of 8740 recordings containing simple command sentences.

The audio files are coded sistematically in order to offer a compact description of their contents:

- idXXXX_YYYYY-PZ-chA-4+3arr.wav
- idXXXX_YYYYY-PZ-chAA-33arr.wav

where:

- "XXXX" is the speaker id.
- "YYYYY" is the sentence id.
- "Z" is the simulated speaker position (1 to 5). See Figure 3.4.
- "AA" is the channel number. In the 4+3 array: ch0-3 (sennheiser), ch4-6 (crown). In the 33 array: ch00-32 (sennheiser), ch0,8,12,14,15,16,17,18,20,24,32 (harmonic subarray).

3.3.3.3 Annotation

For every audio sequence, a corresponding annotation file was generated so that it would also follow the CHIL standards described in [OM06]. These files contain the real position (3D coordinates) of the speaker's mouth at every time frame taking into account the position P1-5 she was standing at (see Figure 3.4) and her height (set to always be 1.75 m). The frame time resolution was defined to be 40ms.

3.3.4 Simulated HIFI

A **simulated version of HIFI database** was also generated in order to compare its performance against the one provided by the real recordings. The close-talk channel files were used as input to the simulation software in order to generate 48 KHz audio files hypothetically sited at the Edecan Room in the Speech Technology Group and captured by the same **array configurations** depicted at the beginning of the previous Section in page 67. The **annotation** files used are the same of those of the Real HIFI corpus.

However, regarding these simulated databases, both SONY and Simulated HIFI, we must take into account the limitations of the simulation software in game. This applications are based on a set of models about sound propagation (in particular, the simulations used in this Master Thesis contain a strong component derived from the direct-path propagation plus a serie of additional reverberated paths that vary on their number of reflections before reaching its target: the components holding from 1 to 5 reflections are computed according to the images method, see [RL06] pp.26-28, while components having from 6 to 20 reflections are computed according to the ray tracing method, see [RL06] pp. 28-29). Nevertheless, the results obtained by these systems are still open to improvements and do not always offer real results. In conclusion, we must be careful when examining the experiments performed under simulated databases: they can offer us hints about our system behaviour but they can never be as credible as those resulting from real recordings. Due to these reasons, as general rule, we will when possible prefer to perform our experiments under real databases rather than with simulated ones.

Finally, in Table 3.3 we show a summary about the main features of the different databases used during the experimental part of this Master Thesis.

Database	f_s	# Speakers	# Frames	Environment room	Microphone Arrays	Annotation
AV16.3	16 KHz	3 static 2 moving	7295 static 917 moving	Idiap Room	2 circular arrays of 8 mics each	CHIL format Every 40 ms
Real HIFI	48 KHz 16 KHz	12 static	9390 static	Edecan Room	4 senn mics equispaced 200 mm (linear array) 3 crown mics (L-shaped array)	CHIL format Every 40 ms
Simulated HIFI	48 KHz	12 static	9390 static	Edecan Room	4 senn mics equispaced 200 mm (linear array) 33 senn mics equispaced 20 mm (linear array) 11 senn mics (harmonic array)	CHIL format Every 40 ms
Simulated Sony	48 KHz	209 static	213950 static	Edecan Room	4 senn mics equispaced 200 mm (linear array) 33 senn mics equispaced 20 mm (linear array) 11 senn mics (harmonic array)	CHIL format Every 40 ms

Table 3.3: Main features of the different databases used in the experimentation

3.4 Baseline results (GCC-PHAT)

In this Section we will evaluate the basic localization algorithm over which this Master Thesis was developed: GCC. Some previous work developed at the Speech Technology Group in Technical University of Madrid (UPM) had set the basis of this algorithm, see [MH06]. However, this turns out to be a limited algorithm: it only takes into account the information given by one single pair of microphones and it is only able to determine just the direction of arrival (DOA) in which the speaker is talking. A theoretical introduction to this algorithm can be found in Section 2.4.1 in page 25.

Nevertheless, a reliable implementation of this strategy is fundamental to later on be able to implement SRP, a more powerful localization system that can be seen as the sumation of all the possible GCC combinations between two microphones, see [Dib00] pp. 78-80 and Section 2.4.3.2 in page 36.

3.4.1 AV16.3

After designing and implementing the algorithm, we made a basic testing of it with those static sequences of AV16.3 database. ST-AV16.3 choice was motivated because of its simplicity and good recording conditions (very constrained, static and without occlusions or overlaps). The evaluation metrics used here cannot follow the CHIL standards since we are not yet able to determine a exact position but only the time difference of arrival (TDOA) between the microphone pairs involved. These TDOAs estimations are given in discrete time units, that is to say, in number of samples. Under this basis, we will evaluate three different metrics:

- *Absolute error*, in number of samples. It is computed as:

$$absolute\_error = |estimated\_tdoa - true\_tdoa| \tag{3.3}$$

- *Relative error*, in %. It is computed as:

$$relative\_error = 100\frac{|estimated\_tdoa - true\_tdoa|}{true\_tdoa} \tag{3.4}$$

- *Histograms*, as done in paper [Dib00], histograms are an useful tool to show both the absolute and the relative error statistical distribution. The histograms displayed at Figure 3.7 were formed as follows: as explained in Section 2.4.1 in page 25, each microphone pair is used to output a TDOA estimation every time frame. We run the same localization experiment for every possible microphone pair combination and

then averaged the absolute and relative errors commited by each one of them along all the time indexes. Given the averaged error info of each of the microphone pairs we can build a *histogram* displaying several bars, each of them having a height equivalent to the number of microphone pairs whose error metric fits the range covered by that bar. Based on this graphic, we can later plot an *accumulated histogram* function reflecting at each abscissa the percentage of microphone pairs whose averaged error metrics are lower than that given abscissa.

Therefore, in order to make use of these metrics, we need to translate the speaker ground truth position given in (x,y,z) coordinates to TDOAs between any possible mic pair combination. A bash script was developed to perform this task. After this, we can easily compare our GCC estimations to the ground truth ones and get some graphics showing some hints about the algorithm performance. We will test separately every possible mic pair combination in order to get some useful information about which mic pairs perform better and why.

Finally, a comparison between this basic GCC algorithm and SRP will be shown. A script was developed converting SRP (x,y,z) estimations into GCC TDOAs estimations related to every microphone pair so that they could be directly compared.

Results are gathered in Figure 3.6 where the *absolute* and *relative errors* for each of the microphone pairs considered are displayed with crosses in the case of GCC-PHAT and circles in the case of SRP-PHAT. Figure 3.7 represents the *histogram* bars and the *accumulated histogram* functions derived from the GCC-PHAT (solid lines) and the SRP-PHAT (dashed lines) methods.

We get two important conclusions from Figures 3.6 and 3.7. First, the overwhelming improvement of the SRP method over the GCC one in terms of absolute error, relative error and error distribution. Second, as shown in Figure 3.6, we can observe that the absolute error has a clear tendency to grow as the distance between the mic pair considered also grows. This is due to the fact that the more distant any microphone pair is, the bigger the time delay differences (TDOAs) between them will be. However, we can at the same time appreciate that the relative error follows the opposite effect: it tends to descend as the distance between microphones grows. Therefore, the TDOA error achieved by distant microphones estimations, when compared to the real TDOA they should have worked out, demonstrates to be lower and makes distant microphones to be, in principle, more reliable in their estimates. This can be due to the fact that the bigger the TDOAs between microphone pairs are, the smaller the quantization effect will be when translating real time delay units in seconds to discrete, digital time delay units in samples. A theoretical explanation about this fact can be found in Section 2.4.1.3 in page 29.

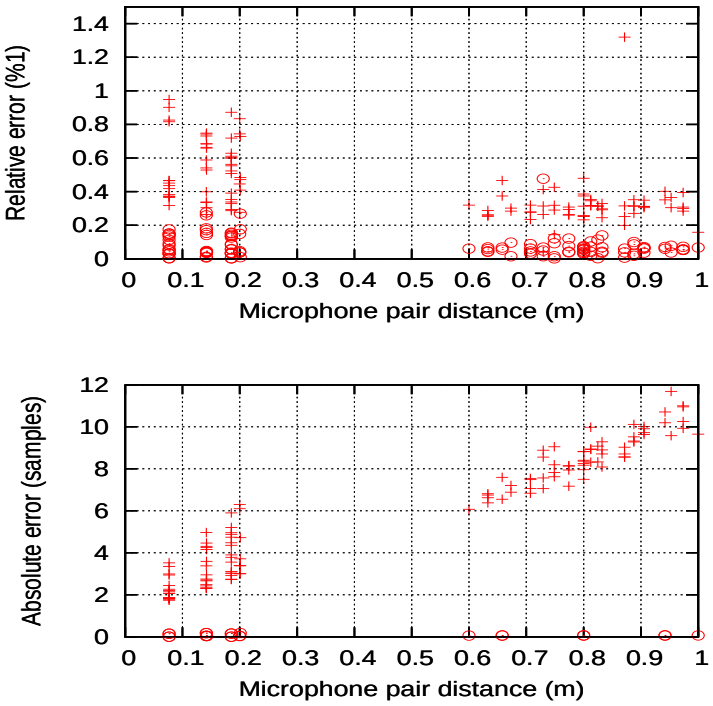

Figure 3.6: Absolute error (in samples, top) and relative error (in %1, bottom). Performance comparison between GCC (crosses) and SRP (circles)

3.5 Basic algorithm characterization (SRP)

In this Section we will evaluate the accuracy of the SRP-PHAT localization algorithm. To start with, the effect of four relevant parameters on the SRP-PHAT performance will be studied: the sampling frequency, the frame size, the FFT size and the window type applied.

3.5.1 Sampling frequency and interpolation techniques

In this Section we will concentrate on measuring the impact of the f_s in the localization results. We will try to determine whether a higher f_s implies better perfomances as well as the effectivity of trying to emulate higher frequencies by the use of interpolation schemes.

3.5.1.1 HIFI

We chose HIFI database since it was not only recorded at 48 KHz but also a downsampled 16 KHz version of it was also generated. This way, in Table 3.4 we will measure the

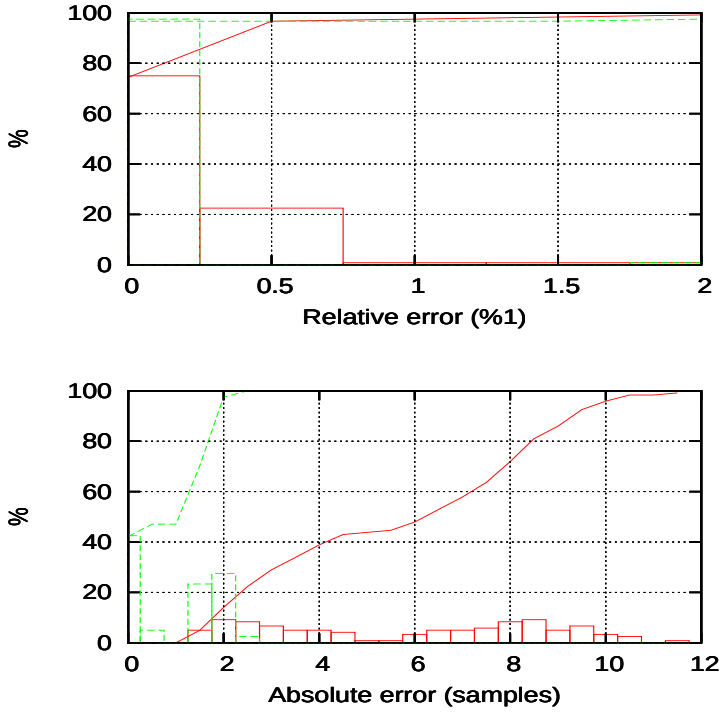

Figure 3.7: Absolute error (in samples, top) and relative error (%, bottom) histograms (boxes) and accumulated histograms (lines). Comparison between GCC (solid) and SRP (dashed)

effect of the sampling frequency by performing several experiments, first, at the original sampling frequency rate, 48 KHz. Secondly, at the downsampled 16 KHz version of the original database and, finally, trying to simulate the original 48 KHz rate based on a x3 interpolated version of the 16 KHz downsampled database.

	fs= 16 KHz	fs= 16x3 KHz	fs= 48 KHz
Pcor	$56.0 \pm 1.0\%$	$76.0 \pm 0.9\%$	$86.0 \pm 0.7\%$
Rel. error reduction		35.7%	53.6%
Bias fine (x:y:z) [mm]	$-6 : -57 : -99$	$3 : -26 : -64$	$-8 : -31 : -61$
Bias fine+gross (x,y,z) [mm]	$25 : -189 : -106$	$48 : 119 : -73$	$22 : 21 : -68$
Bias AEE fine [mm] = MOTP	230	209	206
Rel. AEE reduction		9.1%	10.4%
Bias fine+gross [mm]	585	503	408
Rel. BIAS f+g reduction		14.0%	30.3%
A-MOTA	$11 \pm 0.6\%$	$51 \pm 1.0\%$	$71 \pm 0.9\%$
Rel. error reduction		363.6%	545.5%
Loc. frames	9390	9390	9390
Ref. duration (s)	496.0	496.0	496.0

Table 3.4: Real HIFI. Interpolation techniques. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 150 mm

It is interesting to note the effect of the interpolation: trying to "predict" the 48 KHz samples that would lie in between those obtained at 16 KHz results in a better performance but never as good as the original, 48 KHz sampled data. However, the main conclusion deduced is that the greater the sampling frequency is, the better the performance of the algorithm will be. This is due to the fact that a higher f_s allows us to make a more accurate conversion between the real time delay units in seconds among mic pairs and their corresponding delays in sample units which are the ones we can actually use in our computations.

It is therefore important to take awareness of the negative effect of loosing precision in the conversion from real time units to integer sample ones. We then thought we could try to compensate this effect by applying three different techniques:

- *Frequency-domain version of SRP.* Implies working with real time units in the frequency domain. As explained in the theoretical background, see Section 2.4.3.3 in page 37, the operation of performing correlation and picking out of its samples the one placed at the proper delay is equivalent to that of performing beamforming with that same appropiate delay, with the difference that, in this last case, we can perfectly make use of the exact real time delay value instead of an integer version of it.

- *Interpolation.* Instead of picking the correlation value placed at the appropiate in-

teger version of the time delay (in samples), we can try to infer its real value by performing a linear interpolation between the adjacent samples.

- *Rounding.* It can be seen as a simpler way to interpolate, based on a step function. Instead of selecting just the integer value towards zero of the real delay we will pick that integer which is closest to the real delay.

In Tables 3.5 and 3.6, we performed several experiments fixing all their parameters except those determining which of the previous techniques to choose. The f_s used were 16 and 48 KHz respectively. This way we can measure the effectivity and impact of each of these schemes aimed at obtaining some kind of sub-sample resolution.

	No Rounding	Rounding	Interpolation	Frequency SRP
Pcor	$53.0 \pm 1.0\%$	$56.0 \pm 1.0\%$	$70.0 \pm 0.9\%$	$79.0 \pm 0.8\%$
Rel. error reduction		6.4%	36.2%	55.3%
Bias fine (x:y:z) [mm]	$-42 : 1 : -14$	$-6 : -57 : -99$	$-63 : 32 : -31$	$-13 : -16 : -40$
Bias fine+gross (x,y,z) [mm]	$4 : -24 : -63$	$25 : -189 : -106$	$21 : -5 : -26$	$36 : 52 : -36$
Bias AEE fine [mm] = MOTP	171	230	230	198
Rel. AEE reduction		-34.5%	-34.5%	-15.8%
Bias fine+gross [mm]	593	585	545	465
Rel. BIAS f+g reduction		1.3%	8.1%	21.6%
A-MOTA	$5 \pm 0.4\%$	$11 \pm 0.6\%$	$40 \pm 1.0\%$	$58 \pm 1.0\%$
Rel. error reduction		6.3%	36.8%	55.8%
Loc. frames	9390	9390	9390	9390
Ref. duration (s)	367.0	496.0	496.0	367.0
Run time (real-time units)	0.13	0.13	0.13	115.00
Rel. run-time reduction		0.0%	0.0%	99.9%

Table 3.5: Real HFI. Rounding techniques. fs = 16 KHz. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 150 mm

	No Rounding	Rounding	Interpolation	Frequency SRP
Pcor	$66.0 \pm 1.0\%$	$78.0 \pm 0.8\%$	$69.0 \pm 0.9\%$	$85.0 \pm 0.7\%$
Rel. error reduction		35.3%	8.8%	55.9%
Bias fine (x:y:z) [mm]	$-33 : -41 : -47$	$-79 : 13 : -32$	$-72 : 8 : -12$	$-66 : -23 : -20$
Bias fine+gross (x,y,z) [mm]	$-10 : -28 : -80$	$-28 : 80 : -45$	$-12 : -88 : -4$	$-32 : 72 : -24$
Bias AEE fine [mm] = MOTP	185	217	226	231
Rel. AEE reduction		-17.3%	-22.2%	-24.9%
Bias fine+gross [mm]	495	506	631	444
Rel. BIAS f+g reduction		-2.2%	-27.5%	10.3%
A-MOTA	$32 \pm 0.9\%$	$56 \pm 1.0\%$	$38 \pm 1.0\%$	$71 \pm 0.9\%$
Rel. error reduction		35.3%	8.8%	57.4%
Loc. frames	9390	9390	9390	9390
Ref. duration (s)	367.0	496.0	496.0	496.0
Run time (real-time units)	0.25	0.25	0.25	63.00
Rel. run-time reduction		0.0%	0.0%	99.6%

Table 3.6: Real HIFI. Rounding techniques II. fs = 48 KHz. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame size: 320 ms. Grid step in which the room was divided: 250 mm

As expected, the frequency-domain version of SRP method (FSRP) yields the best results as it works with the exact time delays in the frequency domain. Nevertheless, this method, compared to the other three options (all of them performed in the time domain), turns out to be extremely heavy from the computacional point of view due to the large number of complex multiplications that must be carried out (as many as FFT points times the number of space locations the beamformer is pointed to). Particularly, as reflected in the Tables 3.5 and 3.6, the computational cost of the FSRP method is extremely higher than the schemes carried out in the time domain. A more detailed study about this computational load can be found in Section 3.8 in page 123.

3.5.1.2 AV16.3

We do not have any 48 KHz sampled version of AV16.3 database whose $f_s = 16KHz$. Therefore, we will try to perform x2 and x3 interpolations to check if we can achieve better performances with this method.

Both Tables 3.7 and 3.8, compare the localization results got with the plain, 16 KHz, ST-AV16.3 database to those achieved by x2 and x3 interpolated versions of it. Nevertheless, Table 3.7 just takes into account the interpolation scheme effects since any kind of

rounding is performed, while Table 3.8 reflects what happens when both rounding and interpolation are applied at the same time.

	16 KHz No Rounding	Interpolation x2	Interpolation x3
Pcor	$87.0 \pm 0.8\%$	$88.0 \pm 0.7\%$	$89.0 \pm 0.7\%$
Rel. error reduction		1.1%	2.3%
Bias fine (x:y:z) [mm]	$75 : 13 : -119$	$50 : 9 : -47$	$44 : 17 : -9$
Bias fine+gross (x,y,z) [mm]	$23 : -45 : -135$	$-25 : -66 : -57$	$-40 : -67 : -25$
Bias AEE fine [mm] = MOTP	227	171	144
Rel. AEE reduction		24.7%	36.6%
Bias fine+gross [mm]	350	303	279
Rel. BIAS f+g reduction		13.4%	20.3%
A-MOTA	$73 \pm 1.0\%$	$75 \pm 1.0\%$	$78 \pm 1.0\%$
Rel. error reduction		2.7%	6.8%
Loc. frames	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0

Table 3.7: ST-AV16.3. Interpolation techniques II. Frame size: 640 ms. Grid step in which the room was divided: 50 mm

	16 KHz Rounding	Interpolation x2	Interpolation x3
Pcor	$95.0 \pm 0.5\%$	$89.0 \pm 0.7\%$	$89.0 \pm 0.7\%$
Rel. error reduction		-6.3%	-6.3%
Bias fine (x:y:z) [mm]	$22 : 7 : 27$	$21 : -0 : 39$	$26 : 4 : 45$
Bias fine+gross (x,y,z) [mm]	$-27 : -60 : 17$	$-56 : -86 : 24$	$-56 : -80 : 28$
Bias AEE fine [mm] = MOTP	100	152	141
Rel. AEE reduction		-52.0%	-41.0%
Bias fine+gross [mm]	191	282	272
Rel. BIAS f+g reduction		-47.6%	-42.4%
A-MOTA	$90 \pm 0.7\%$	$78 \pm 1.0\%$	$80 \pm 0.9\%$
Rel. error reduction		-13.3%	-11.1%
Loc. frames	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0

Table 3.8: ST-AV16.3. Interpolation techniques. Frame size: 320 ms. Grid step in which the room was divided: 100 mm

From Tables 3.7 and 3.8, we can note that interpolation only leads to an improvement in the mean distance error when no rounding technique is applied to real time delays.

From this fact, we can therefore infer that both interpolation in the frequency domain and rounding in the time domain are somehow equivalent techniques in the sense that both of them aim at getting a better sub-sample resolution or, in other words, a more precise use of the real time delay between mic pairs. Again, this performance-equivalent techniques turn out to have drastically different effects on the computational load: while frequency interpolating x2 and x3 result in neccessarily having to use twice and three times as big FFT transforms (with consequent twice and three times bigger execution times), rounding in the time domain allows not to increase the FFT size and keep the execution time. For instance, some execution times for the experiments above are reflected in Table 3.9.

16 KHz with no rounding	70 ms per frame 1.75 times real-time
16 KHz with rounding	72 ms per frame 1.8 times real-time
16 KHz x 2	120 ms per frame 3 times real-time
16 KHz x 3	220 ms per frame 5.5 times real-time

Table 3.9: Static AV16.3. Instance of computational loads for interpolation schemes

3.5.1.3 HIFI vs. AV16.3

Finally we can make an interesting observation about how well our algorithms work depending on the chosen database. As we can check, AV16.3 performance rates are much better than HIFI ones. This is mainly due to two reasons:

- *Recording conditions.* AV16.3 database was recorded in a smart room designed to somehow limit reverberation and background noise, see [Moo02]. On the other side, HIFI database was recorded in the same laboratory this Master Thesis was developed: a room with big windows yielding high reflecting coefficients and large background noise since surrounded by lots of PCs whose fans introduce a strong low frequency noise.
- *Array geometry.* AV16.3 database was recorded by two circular arrays of 8 microphones each resulting in a total of 16 microphones who can be combained into 120 different mic pairs. Meanwhile, HIFI recordings were captured by a sole linear array of 4 elements which only allows 6 different single combinations and presents worse directivity pattern conditions as far as the main lobe width and the frequency aliasing of the array are considered.

3.5.2 Frame size

In this Section we will concentrate on measuring the impact of the frame size in the localization results. We will try to determine whether a bigger frame size implies better perfomances as well as measuring how the processing delay derived from big frame sizes can affect localization when dealing with moving speakers.

3.5.2.1 AV16.3

Here, the choice of the AV16.3 database was due to the fact that it includes both static and moving speakers. Consequently, this fact allows us to measure the effect of the frame length on our algorithm performance under different circumstances. We will first concentrate on those recordings containing a static speaker. Table 3.10 and Figure 3.8 will show the localization results of these experiments in which the only parameter to vary was the frame size. Values considered ranged from 80, 160, 200, 320, 500, 640 to 1000 ms.

	Frame Size = 40 ms	320 ms	640 ms
Pcor	$81.0 \pm 0.9\%$	$87.0 \pm 0.8\%$	$94.0 \pm 0.5\%$
Rel. error reduction		7.4%	16.0%
Bias fine (x:y:z) [mm]	$58 : 17 : 63$	$62 : 18 : 65$	$62 : 19 : 63$
Bias fine+gross (x,y,z) [mm]	$-76 : -174 : 39$	$-62 : -155 : 46$	$7 : -45 : 54$
Bias AEE fine [mm] = MOTP	197	191	190
Rel. AEE reduction		3.0%	3.6%
Bias fine+gross [mm]	490	423	291
Rel. BIAS f+g reduction		13.7%	40.6%
A-MOTA	$61 \pm 1.1\%$	$74 \pm 1.0\%$	$87 \pm 0.8\%$
Rel. error reduction		21.3%	42.6%
Loc. frames	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0

Table 3.10: ST-AV16.3. Frame size effect. Grid step in which the room was divided: 150 mm

We can conclude that, since we have temporal frame to observe the speaker position, the bigger the length of the frame, the greater the estimates we will get. Also, as the frame length increases we will also need to compute more points in our FFT transforms therefore increasing the computational time required as seen in Figure 3.9.

Next, the effects when considering speakers in movement will be studied. Measurements were made at frame lengths of 80, 160, 200, 320, 400, 500 and 640 ms. The aggregated results from both seq11 and seq15 were gathered at Table 3.11 and then disglosed

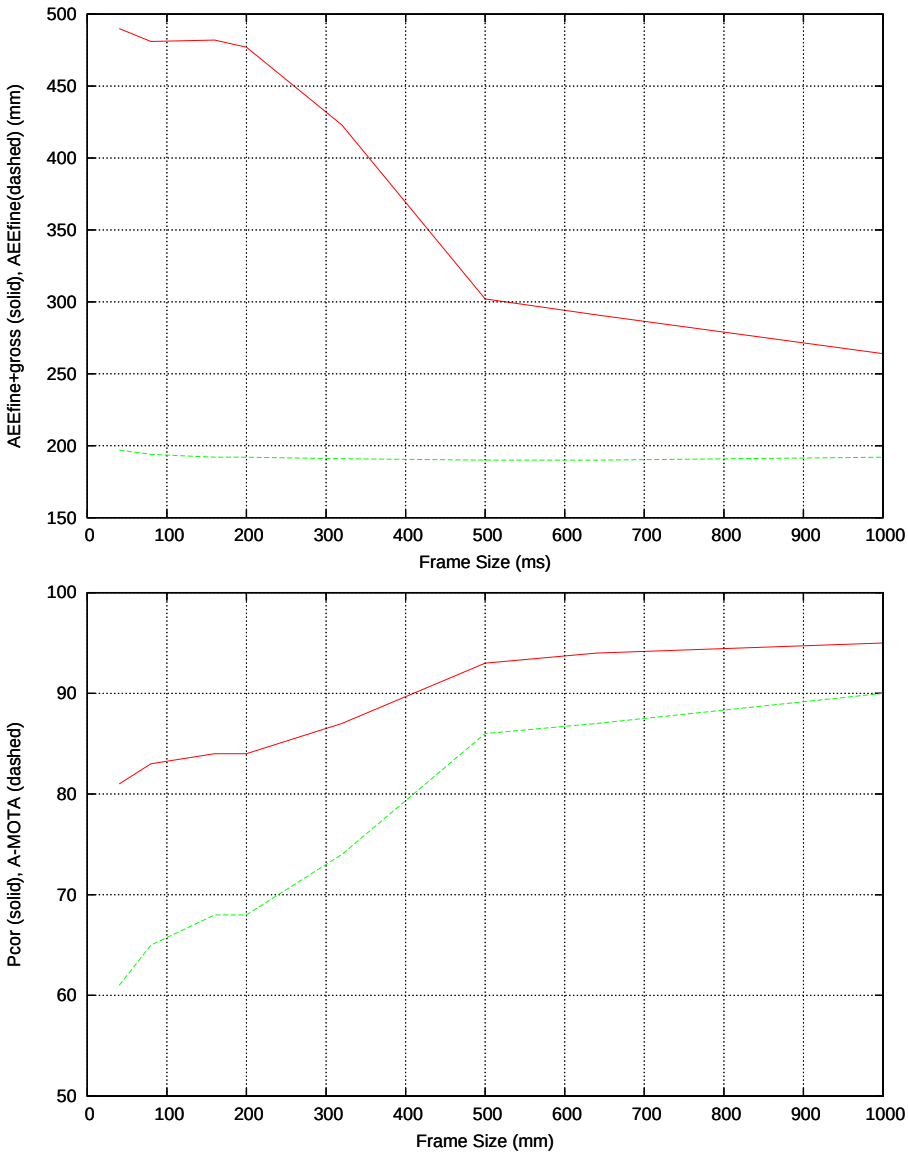

Figure 3.8: Static AV16.3 average error and localization rate as a function of the frame size considered. Top: Average error fine+gross (solid), average error fine (dashed). Bottom: Pcor (solid), A-MOTA (dashed).

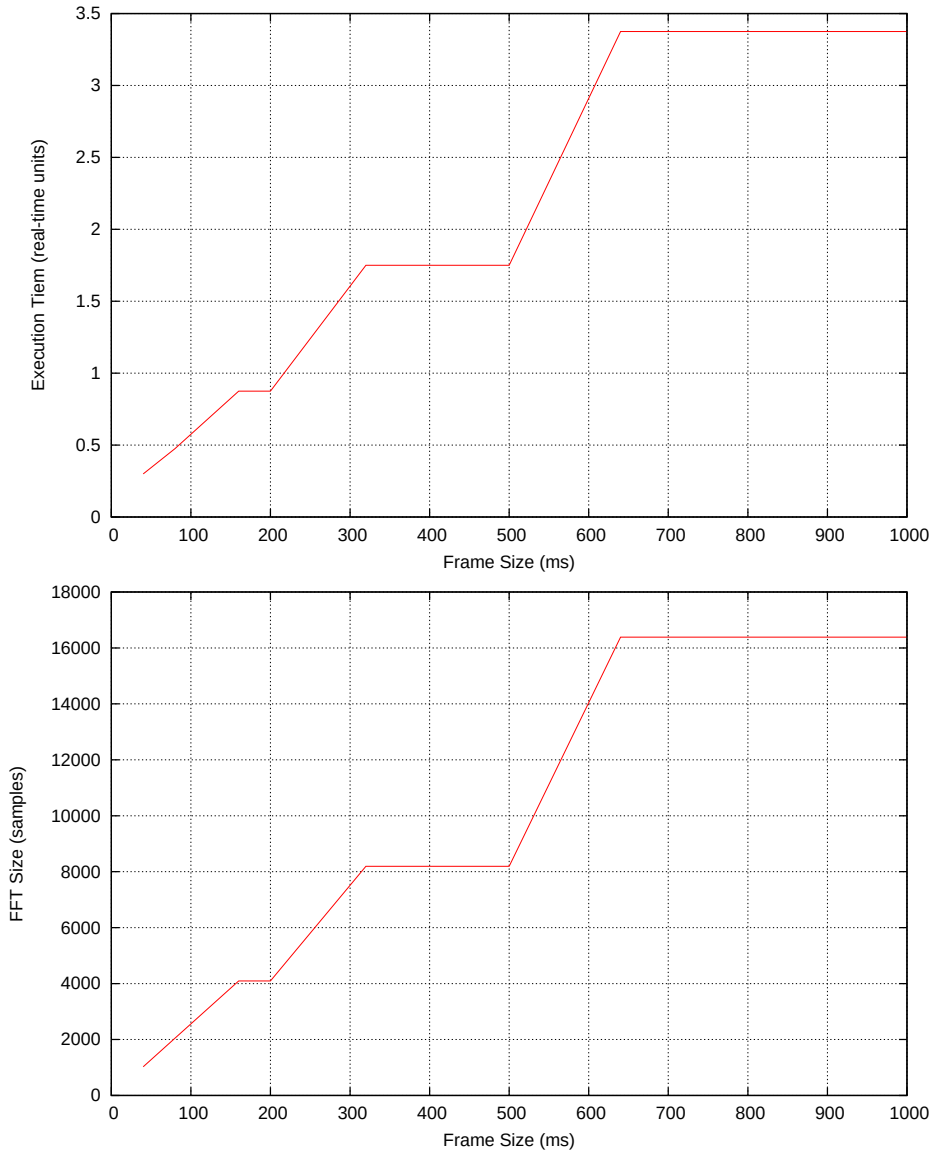

Figure 3.9: Static AV16.3 run time and FFT size as a function of the frame size

in Table 3.12, for seq11 alone, and Table 3.13, for seq15 alone, since their results came up to diverge significally. Finally, Figure 3.10 shows the Pcor, A-MOTA and average error figures as a function of the varying frame size.

	seq11 y 15. 40 ms	320 ms	640 ms
Pcor	$62.0 \pm 3.1\%$	$71.0 \pm 2.9\%$	$69.0 \pm 3.0\%$
Rel. error reduction		14.5%	11.3%
Bias fine (x:y:z) [mm]	$53 : 15 : 59$	$49 : 67 : 60$	$10 : 86 : 41$
Bias fine+gross (x,y,z) [mm]	$-95 : -429 : 8$	$-115 : -278 : 13$	$-118 : -185 : 9$
Bias AEE fine [mm] = MOTP	218	228	259
Rel. AEE reduction		-4.6%	-18.8%
Bias fine+gross [mm]	800	660	636
Rel. BIAS f+g reduction		17.5%	20.5%
A-MOTA	$23 \pm 2.7\%$	$42 \pm 3.2\%$	$37 \pm 3.1\%$
Rel. error reduction		82.6%	60.9%
Loc. frames	917	917	917
Ref. duration (s)	42.0	42.0	42.0

Table 3.11: MV-AV16.3. Frame size effect. Grid step in which the room was divided: 150 mm

	seq11. 40 ms	320 ms	640 ms
Pcor	$75.0 \pm 3.9\%$	$87.0 \pm 3.0\%$	$83.0 \pm 3.4\%$
Rel. error reduction		16.0%	10.7%
Bias fine (x:y:z) [mm]	$63 : -5 : 68$	$54 : 22 : 68$	$26 : 28 : 51$
Bias fine+gross (x,y,z) [mm]	$90 : -6 : 64$	$90 : 38 : 71$	$67 : 59 : 74$
Bias AEE fine [mm] = MOTP	201	214	253
Rel. AEE reduction		-6.5%	-25.9%
Bias fine+gross [mm]	386	272	319
Rel. BIAS f+g reduction		29.5%	17.4%
A-MOTA	$50 \pm 4.5\%$	$74 \pm 3.9\%$	$67 \pm 4.2\%$
Rel. error reduction		48.0%	34.0%
Loc. frames	481	481	481
Ref. duration (s)	21.1	21.1	21.1

Table 3.12: MV-AV16.3 seq11. Frame size effect. Grid step in which the room was divided: 150 mm

	seq15. 40 ms	320 ms	640 ms
Pcor	$48.0 \pm 4.7\%$	$55.0 \pm 4.7\%$	$54.0 \pm 4.7\%$
Rel. error reduction		14.6%	12.5%
Bias fine (x:y:z) [mm]	$42 : 35 : 50$	$44 : 114 : 51$	$-6 : 146 : 31$
Bias fine+gross (x,y,z) [mm]	$-284 : -861 : -50$	$-325 : -601 : -47$	$-308 : -434 : -58$
Bias AEE fine [mm] = MOTP	235	242	265
Rel. AEE reduction		-3.0%	-12.8%
Bias fine+gross [mm]	1224	1058	960
Rel. BIAS f+g reduction		13.6%	21.6%
A-MOTA	$-4 \pm 1.8\%$	$10 \pm 2.8\%$	$7 \pm 2.4\%$
Rel. error reduction		-350.0%	-275.0%
Loc. frames	436	436	436
Ref. duration (s)	20.6	20.6	20.6

Table 3.13: MV-AV16.3 seq15. Frame size effect. Grid step in which the room was divided: 150 mm

In this case, the performance does increase as we also increase our frame length but this just happens up to a certain value of it. This effect is due to the fact that the speakers are talking at the same time they are moving and, therefore, if we take a too long frame, the speaker's position will be different between the beginning and the end of it. We can try to make some basic calculations about how long our frame can be before this changing position effect starts having a significal importance on our algorithm localization rate. As explained above, whenever our estimation lies less than 500 mm from the true speaker localization we can consider it as a fine error. Taking into account that a human speaker walks at approximately 5 Km/h, then:

$$500mm * \frac{1m}{1000mm} * \frac{1h}{5km} * \frac{1km}{1000m} * \frac{3600s}{1h} = 0.36s \tag{3.5}$$

Therefore, taking frames longer than 360 ms should start worsening the performance. We can check this assumption by looking at Figure 3.10. As we can see, the localization rates grow up to 320-400 ms and after that they start to slowly decrease: the positive effect of having more data to locate is cancelled by the changing position of the speaker.

In Figure 3.10, we can also note that the performance obtained with moving speaker in seq11 is obviously better than the one got with seq15. At first, we thought this might be due to different, more abrupt behaviour of the speaker in seq15. Nevertheless, a close inspection of the videos corresponding to these sequences refuted this theory. Finally, we were able to find out the real reason behind this phenomena by playing at the same time

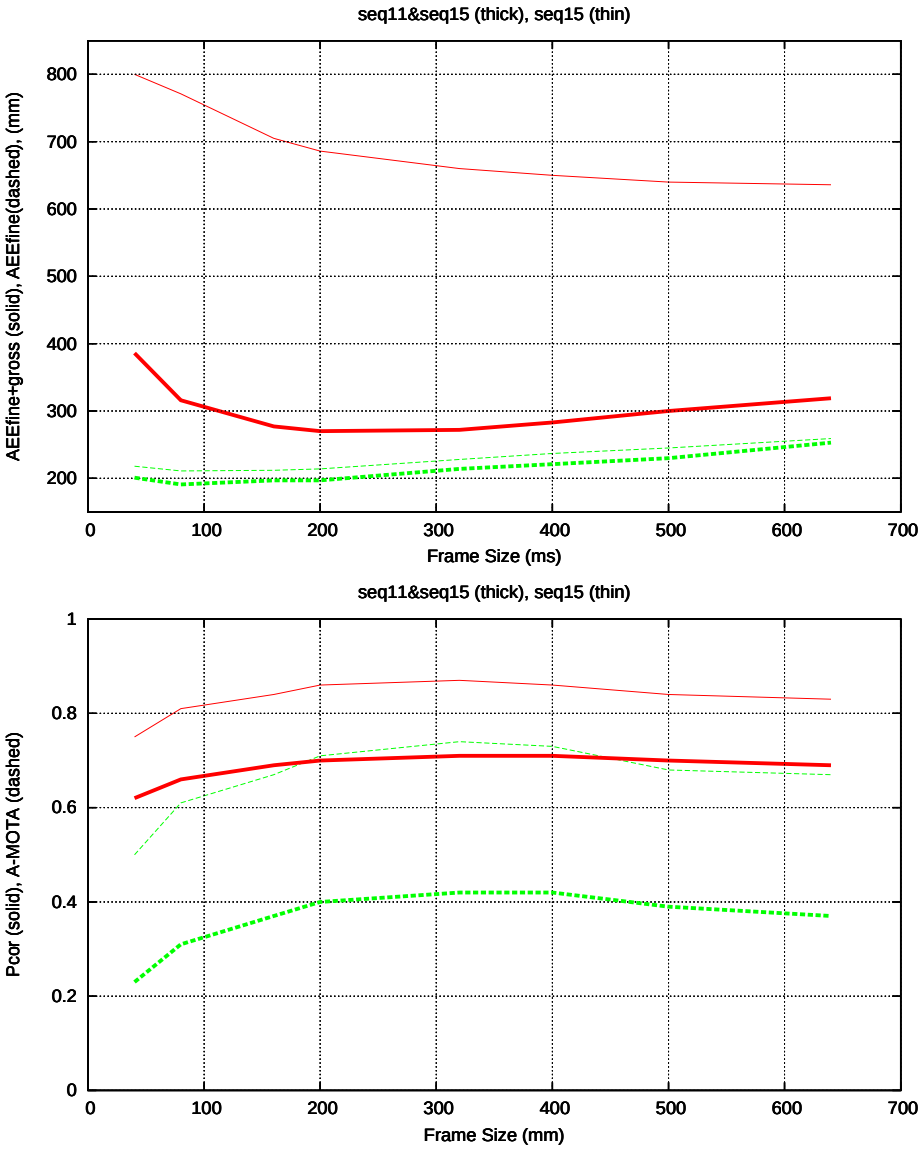

Figure 3.10: Moving AV16.3 average error and localization rate as a function of the frame size considered. Top: Average error fine+gross (solid) and average error fine (dashed). Bottom: Pcor (solid) and A-MOTA (dashed). Aggregated results for seq 11 and 15 are shown in thick lines, while seq15 alone is shown with thin lines.

the video recording of the sequence and a graphical representation of our localization system estimates corresponding to it: either the VAD used by the Idiap Institute or the human operator in charge of labelling seq15 demonstrated to commit a mistake since they were including long speechless periods of time in their reference ground truth file. Thus, although our system localization estimates were proper during the speech periods, they were obviously wrong during these specific, wrongly marked speechless periods of time, finally leading to a dramatic drop in the overall performance. This fact showed us the extreme importance of having an effective VAD precisely marking which are the speech periods, able to be localized, and which are the speechless ones, discarded in terms of system evaluation.

3.5.2.2 HIFI

We will also check the frame size effect on the HIFI database to see if the conclusions got with AV16.3 are of general validity. Experiments at frame sizes of 40, 80, 160, 320, 500, 640 and 1000 ms were performed and their results depicted in Table 3.14 and Figure 3.11.

	Frame Size = 40 ms	320 ms	640 ms
Pcor	$65.0 \pm 1.0\%$	$88.0 \pm 0.7\%$	$84.0 \pm 0.7\%$
Rel. error reduction		35.4%	29.2%
Bias fine (x:y:z) [mm]	$-8 : -27 : -60$	$-15 : -11 : -27$	$-7 : -34 : -61$
Bias fine+gross (x,y,z) [mm]	$25 : 15 : -69$	$20 : 43 : -28$	$50 : 19 : -69$
Bias AEE fine [mm] = MOTP	213	196	207
Rel. AEE reduction		8.0%	2.8%
Bias fine+gross [mm]	723	386	418
Rel. BIAS f+g reduction		46.6%	42.2%
A-MOTA	$30 \pm 0.9\%$	$76 \pm 0.9\%$	$69 \pm 0.9\%$
Rel. error reduction		153.3%	130.0%
Loc. frames	9390	9390	9375
Ref. duration (s)	367.0	367.0	367.0

Table 3.14: Real HIFI. Frame size effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Grid step in which the room was divided: 150 mm

As we can observe in Figure 3.11, the localization rate in HIFI database also grows as the frame size considered grows but just to a certain point located around 320 ms. From this point onwards, increments of the frame size considered do not translate into localization rate improvements but tend to very slowly descend. This phenomena is due to the fact that all the recordings involved in the HIFI database consist on really short com-

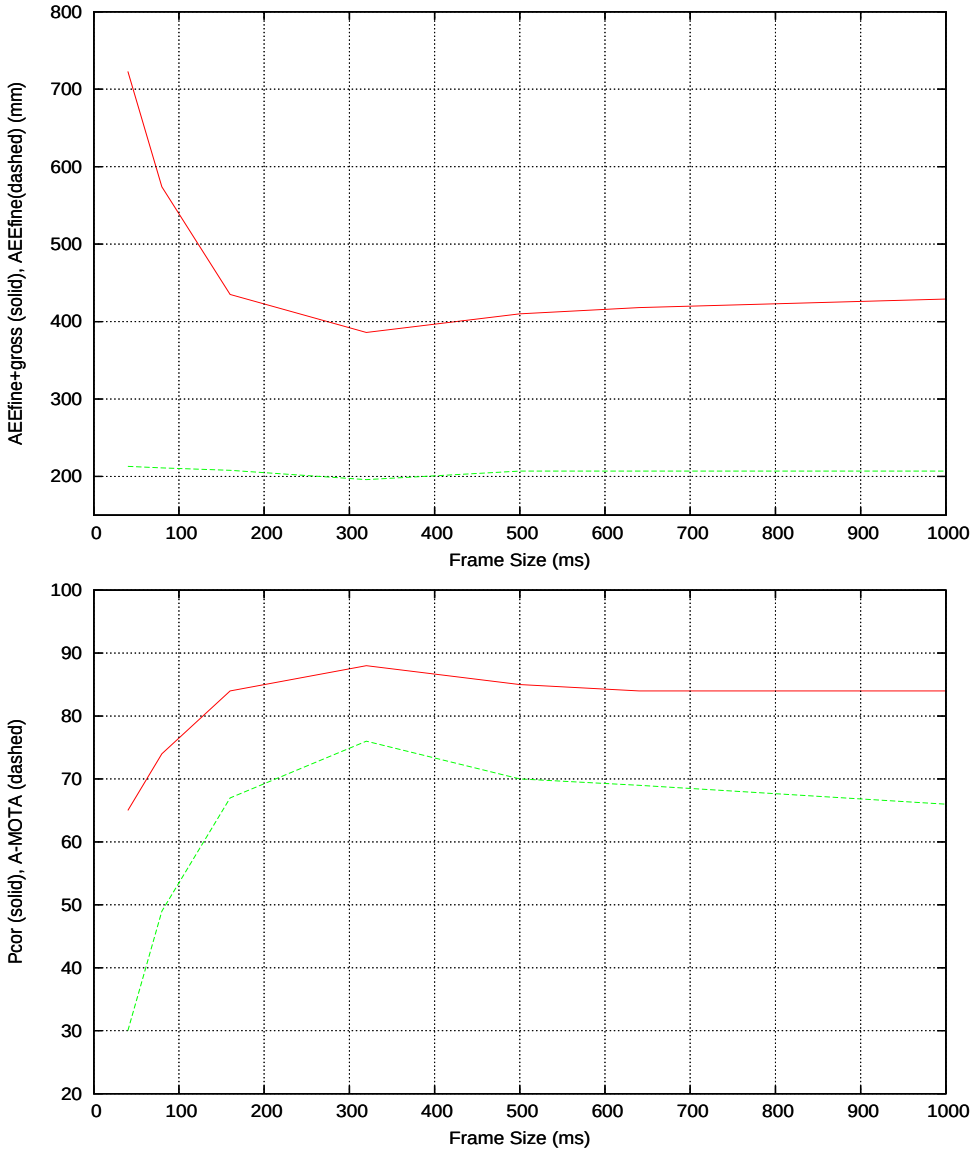

Figure 3.11: Real HIFI average error and localization rate as a function of the frame size considered

mand sentences (from Table 3.14 we can deduce the average duration of each recording to be approximately 1.65 seconds being 223 the files considered and 367 seconds the total duration of the reference). Therefore, increments of the frame size will imply a dramatic descend in the number of frames required to analyze the complete file. The last of these frames will usually be strongly zero-padded as it will not cover enough speech data as to fill a complete frame size and will therefore present poor localization results. In addition, there will also be a high probability that these long frames cover not only speech periods but also a significant part of silence periods that will consequently seriously corrupt the file overall localization results.

3.5.3 FFT size

We will try to evaluate our algorithm depending on the number of samples of the FFT transforms we use. In theory, a larger FFT transform result in a more accurate representation of the signal spectrum.

It is important to note that the FFT size imposes an important limitation on how far our speaker search can reach. Specifically, the FFT size imposes the length of our correlation fuctions where we perform the search of the appropiate power value placed at the appropiate time-delay (in sample units) where we expect our speaker to be. Consequently, if our correlation function has $N = fftSize$ points we will only be able to search among those time-delays (in samples) included in the range $[-\frac{N-1}{2}, \frac{N}{2}]$. This fact, depending on both the corresponding sampling frequency we are working with and the speed of sound, can be translated into the maximum distance any speaker can stand from our array in order to be localized. Next, some basic computations are presented, in the most restrictive conditions, in order to check how this phenomenon can threathen our algorithm ability to locate:

1. *At 16 KHz* The minimum frame size, M, used in any experiment was 40 ms. This imposes a minimum FFT size, N, of:

   $M_{min} * f_s = 40ms * 16KHz = 640$ samples
   Since $(N \geq M) \wedge N = 2^n$: $N_{min} = 1024$ and $time\_delays \in [-511, 512]$ samples
   $delay_{max} = \frac{512 samples}{16KHz} * speed\_sound = 32ms * 345\frac{m}{s} = 11.04$ m

2. *At 48 KHz* The minimum frame size, M, used in any experiment was 40 ms. This imposes a minimum FFT size, N, of:

$M_{min} * f_s = 40ms * 48KHz = 1920$ samples
Since $(N \geq M) \wedge N = 2^n$: $N_{min} = 2048$ and $time\_delays \in [-1023, 1024]$ samples
$delay_{max} = \frac{1024 samples}{48KHz} * speed\_sound = 21.33ms * 345\frac{m}{s} = 7.36$ m

In both cases, these limits are large enough in order not to impose any restriction in the room environments considered.

Now, we will concentrate on the hypothetic effect of a finer frequency resolution of the Fourier transform on our algorithms. In order to do so we will make an experiment picking a frame size just as big as the FFT size chosen to later compare its results to the ones obtained when applying a FFT transform twice as big.

3.5.3.1 AV16.3

In next Table 3.15, we compare the results obtained in AV16.3 in the two cases previously mentioned: when using a FFT size just as big as the frame size considered and when picking a FFT size twice as big.

	FFT Size = 8192	FFT Size = 16384
Pcor	$88.0 \pm 0.7\%$	$88.0 \pm 0.7\%$
Rel. error reduction		0.0%
Bias fine (x:y:z) [mm]	$-46 : -30 : -81$	$-46 : -30 : -81$
Bias fine+gross (x,y,z) [mm]	$-110 : -131 : -95$	$-111 : -130 : -95$
Bias AEE fine [mm] = MOTP	223	224
Rel. AEE reduction		-0.4%
Bias fine+gross [mm]	356	356
Rel. BIAS f+g reduction		-0.0%
A-MOTA	$76 \pm 1.0\%$	$76 \pm 1.0\%$
Rel. error reduction		0.0%
Loc. frames	7295	7295
Ref. duration (s)	600.0	600.0

Table 3.15: ST-AV16.3. FFT size effect. fs = 16 KHz. Frame Size = 512 ms equivalent to 8192 samples at fs = 16KHz

As we can see there is no impact of the FFT size on our algorithm performance as long as this value keeps bigger than the frame size. In this case, the only effect of doubling the FFT size is a doubled computation time from the 70 ms/frame (1.75 times real-time) of the 8192 transform to the 140 ms/frame (3.5 times real-time) of the 16384 one.

We did not see any differential facts in the rest of the databases, compared to AV16.3, that could have an influence in our algorithm about the localization results with the FFT size. This is why we decided to conclude here our experimentation about this issue and proceed with new experiments.

3.5.4 Windowing

When dividing the recorded signal into frames, we can either directly process them (equivalent to applying a rectangular window) or try to previously apply a windowing function in order to smoothen them and prevent possible border-effects that may appear in the frequency domain if the time-domain signal terminates too abruptly.

3.5.4.1 AV16.3

In the next Tables 3.16 and 3.17, we will measure the performance of different types of windows: rectangular, Hamming, Hanning, Blackman and Bartlett, applied to ST-AV16.3 and MV-AV16.3 corpus respectively, in order to determine that one yielding the best results.

	Rectangular	Hamming	Hanning	Blackmann	Barlett
Pcor	$88.0 \pm 0.7\%$	$88.0 \pm 0.7\%$	$86.0 \pm 0.8\%$	$85.0 \pm 0.8\%$	$74.0 \pm 1.0\%$
Rel. error reduction		0.0%	-2.3%	-3.4%	-15.9%
Bias fine (x:y:z) [mm]	$-47 : -30 : -81$	$-47 : -30 : -81$	$-46 : -29 : -80$	$-46 : -29 : -80$	$-42 : -27 : -77$
Bias fine+gross (x,y,z) [mm]	$-111 : -131 : -95$	$-111 : -131 : -95$	$-132 : -162 : -97$	$-138 : -168 : -98$	$-251 : -207 : -98$
Bias AEE fine [mm] = MOTP	224	224	223	224	224
Rel. AEE reduction		-0.0%	0.4%	-0.0%	-0.0%
Bias fine+gross [mm]	357	357	395	404	558
Rel. BIAS f+g reduction		-0.0%	-10.6%	-13.2%	-56.3%
A-MOTA	$76 \pm 1.0\%$	$76 \pm 1.0\%$	$72 \pm 1.0\%$	$71 \pm 1.0\%$	$47 \pm 1.1\%$
Rel. error reduction		0.0%	-5.3%	-6.6%	-38.2%
Loc. frames	7295	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0	600.0

Table 3.16: ST-AV16.3. Windows effect. fs = 16 KHz. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm

	Hamming	Hanning	Blackmann	Barlett
Pcor	$66.0 \pm 3.1\%$	$66.0 \pm 3.1\%$	$66.0 \pm 3.1\%$	$62.0 \pm 3.1\%$
Rel. error reduction		0.0%	0.0%	−6.1%
Bias fine (x:y:z) [mm]	$-82 : 66 : -104$	$-84 : 66 : -105$	$-84 : 64 : -105$	$-88 : 86 : -100$
Bias fine+gross (x,y,z) [mm]	$-210 : -312 : -142$	$-211 : -314 : -144$	$-217 : -328 : -146$	$-240 : -319 : -145$
Bias AEE fine [mm] = MOTP	270	272	273	314
Rel. AEE reduction		−0.7%	−1.1%	−16.3%
Bias fine+gross [mm]	644	652	661	727
Rel. BIAS f+g reduction		−1.2%	−2.6%	−12.9%
A-MOTA	$32 \pm 3.0\%$	$32 \pm 3.0\%$	$32 \pm 3.0\%$	$24 \pm 2.8\%$
Rel. error reduction		0.0%	0.0%	−25.0%
Loc. frames	917	917	917	917
Ref. duration (s)	42.0	42.0	42.0	42.0

Table 3.17: MV-AV16.3. Windows effect. fs = 16 KHz. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm

From results in Tables 3.16 and 3.17, we can conclude that the window function showing the best performance is Hamming, both under static and moving speakers. We will therefore Hamming window to be the used-by-default one in all the experiments performed.

3.5.4.2 HIFI

In the following Table 3.17, we limit to compare the performance of rectangular and Hamming windows in the case of the HIFI corpus to extend the validity of the results previouly got.

	Rectangular	Hamming
Pcor	$71.0 \pm 0.9\%$	$72.0 \pm 0.9\%$
Rel. error reduction		1.4%
Bias fine (x:y:z) [mm]	$-14 : 20 : 22$	$-13 : 28 : 22$
Bias fine+gross (x,y,z) [mm]	$-28 : 85 : 21$	$-30 : 59 : 22$
Bias AEE fine [mm] = MOTP	252	253
Rel. AEE reduction		-0.4%
Bias fine+gross [mm]	602	580
Rel. BIAS f+g reduction		3.7%
A-MOTA	$42 \pm 1.0\%$	$44 \pm 1.0\%$
Rel. error reduction		4.8%
Loc. frames	9390	9390
Ref. duration (s)	496.0	496.0

Table 3.18: Real HIFI. Windows effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 500 ms. Grid step in which the room was divided: 150 mm

As we can see in Table 3.18, applying a Hamming window leads to slight improvements compared to the rectangular one.

3.5.5 Search space grid

When performing SRP we first divide our room in a serie of points where to aim our array in order to find out in which one of them the speaker was located. The way in which the room is divided demonstrated to be crucial in the algorithm performance. We typically chose to pick equispaced (x,y,z) points.

3.5.5.1 AV16.3

Here in Table 3.19, we show some results for 50, 100, 150, 200 and 250 mm equispaced grids in the case of the ST-AV16.3 database. Under this experimental conditions, the pointing is carried out by two circular uniform arrays of 8 microphones each, see Section 3.3.1.1 in page 61.

	50 mm	100 mm	150 mm	200 mm	250 mm
Pcor	$97.0 \pm 0.4\%$	$96.0 \pm 0.4\%$	$95.0 \pm 0.5\%$	$89.0 \pm 0.7\%$	$87.0 \pm 0.8\%$
Rel. error reduction		-1.0%	-2.1%	-8.2%	-10.3%
Bias fine (x:y:z) [mm]	$23 : 8 : 26$	$54 : 18 : 43$	$61 : 18 : 61$	$74 : -2 : 41$	$123 : 15 : 42$
Bias fine+gross (x,y,z) [mm]	$-9 : -36 : 18$	$23 : -22 : 37$	$27 : -19 : 54$	$61 : -34 : 46$	$112 : -33 : 48$
Bias AEE fine [mm] = MOTP	99	156	192	228	255
Rel. AEE reduction		-57.6%	-93.9%	-130.3%	-157.6%
Bias fine+gross [mm]	160	223	264	319	351
Rel. BIAS f+g reduction		-39.4%	-65.0%	-99.4%	-119.4%
A-MOTA	$93 \pm 0.6\%$	$91 \pm 0.7\%$	$90 \pm 0.7\%$	$78 \pm 1.0\%$	$75 \pm 1.0\%$
Rel. error reduction		-2.2%	-3.2%	-16.1%	-19.4%
Loc. frames	7295	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0	600.0

Table 3.19: ST-AV16.3. Grid spacing effect. fs = 16 KHz. Frame Size= 640 ms

As expected, as the search grid is defined with more a more finesse the localization results turn out to be more and more precise.

3.5.5.2 HIFI

Now, we will analyze which is the effect that the grid spacing of the search space has on the Real HIFI corpus. In next Table 3.20, we evaluate the results for 25, 50, 100, 150 and 250 mm equispaced grids. In this case, the pointing is carried out by the uniform linear array of 4 sennheiser microphones equispaced 200 mm, see Figure 3.5 in page 68.

	250 mm	150 mm	100 mm	50 mm	25mm
Pcor	$78.0 \pm 0.8\%$	$88.0 \pm 0.7\%$	$71.0 \pm 0.9\%$	$69.0 \pm 0.9\%$	$69.0 \pm 0.9\%$
Rel. error reduction		12.8%	−9.0%	−11.5%	−11.5%
Bias fine (x:y:z) [mm]	$-78 : 11 : -31$	$-15 : -11 : -27$	$-18 : -64 : -68$	$-12 : -51 : -121$	$-10 : -87 : -143$
Bias fine+gross (x,y,z) [mm]	$-24 : 71 : -44$	$20 : 43 : -28$	$19 : -257 : -84$	$20 : -221 : -128$	$17 : -240 : -145$
Bias AEE fine [mm] = MOTP	216	196	193	203	228
Rel. AEE reduction		9.3%	10.6%	6.0%	−5.6%
Bias fine+gross [mm]	497	386	555	527	538
Rel. BIAS f+g reduction		22.3%	−11.7%	−6.0%	−8.2%
A-MOTA	$57 \pm 1.0\%$	$76 \pm 0.9\%$	$41 \pm 1.0\%$	$37 \pm 1.0\%$	$37 \pm 1.0\%$
Rel. error reduction		33.3%	−28.1%	−35.1%	−35.1%
Loc. frames	9390	9390	9390	9390	9390
Ref. duration (s)	367.0	367.0	367.0	367.0	367.0

Table 3.20: Real HIFI. Grid spacing effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms

We can here appreciate some odd results. The localization rate increases, as expected, when reducing the grid from 250 mm to 150 mm. However, if we keep on reducing the grid spacing, contrary to what we should expect, we appreciate a descending tendency in the localization rate. We can make a closer inspection of these details by looking at Figure 3.12

The unexpected results in Table 3.20 and Figure 3.12 can be explained as a consequence of the poor directivity pattern properties of the 4 sennheiser microphones linear array which was used to capture the audio data. A close inspection of it can be done in Figures 3.13, 3.14, 3.15, 3.16, 3.17.

As reflected in Figures 3.13, 3.14, 3.15, 3.16, 3.17 and as referred in [AM01], as the frequency gets higher the main lobe beam-width tends to decrease, see Figure 2.7 in page 18. However, since the intermicrophone distance is quite high in this case, spatial aliasing, that is, the appearance of grating lobes in undesired directions of space, starts occuring from relatively low frequencies, in particular according to Nyquist principle, see Section 2.3.1 in page 18:

$$d < \frac{\lambda_{min}}{2} =: f_{max} < \frac{c}{2d} = \frac{345\frac{m}{s}}{0.4m} = 862.5Hz \tag{3.6}$$

where d is the intermic distance (200 mm),c is the speed of sound ($345\frac{m}{s}$) and λmin and f_{max} are, respectively, the minimum wavelength and the maximum frequency allowed to avoid spatial aliasing.

We therefore have the following effect: On the one hand, at the low frequencies range, we are free of the undesired effects of spatial aliasing but the width of our main lobe it is too wide to get an accurate beamforming. Our Steered Response Power (SRP) algorithm cannot sort out well between different positions if they are too close because the directivity pattern does not point precisely to just one point but integrates power coming from different locations sited at gross areas of space. In Table 3.21 we show some rough computations about the differences, in degrees, between spatial points equispaced 25, 50, 100, 150 and 250 mm. As we can check, as the grid inter-spacing becomes more and more finesse, the difference in degrees gets too small to make any significal effect on the power patterns captured by directivities shown in Figures 3.13 and 3.14. On the other hand, on the high frequency bands, which demonstrate to output the best localization results in general, see Section 3.9.4 in page 143, we see much finer main lobes and we are thus able to get proper pointings to even very close spatial locations. Nevertheless, the effect of the grating lobes, pointing at totally undesired directions of space and integrating the energy coming from them, seriously corrupt the results.

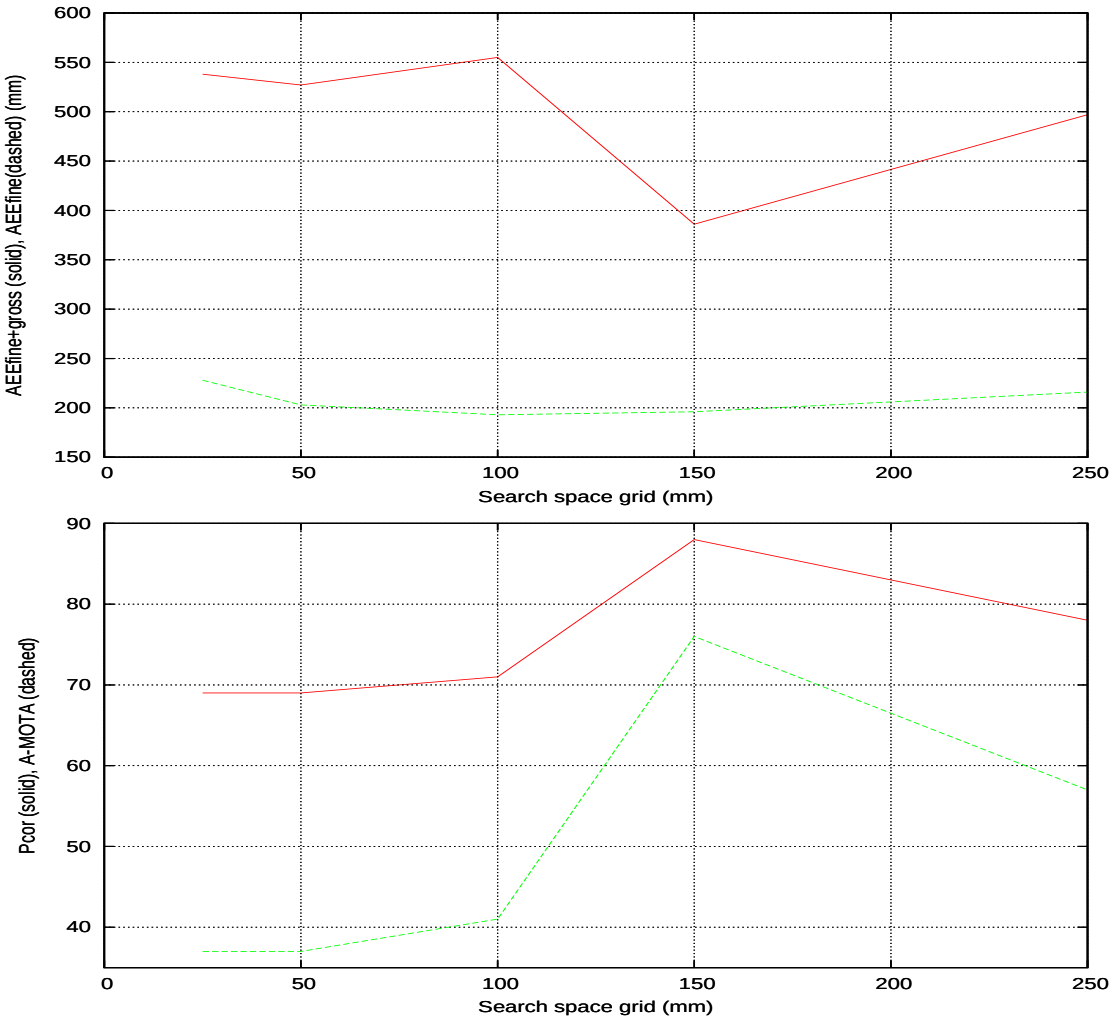

Figure 3.12: Real HIFI. Search grid effect. Top: Average error fine+gross (solid) and average error fine (dashed) as function of the grid spacing. Bottom: Pcor (solid) and A-MOTA (dashed) as function of the grid spacing.

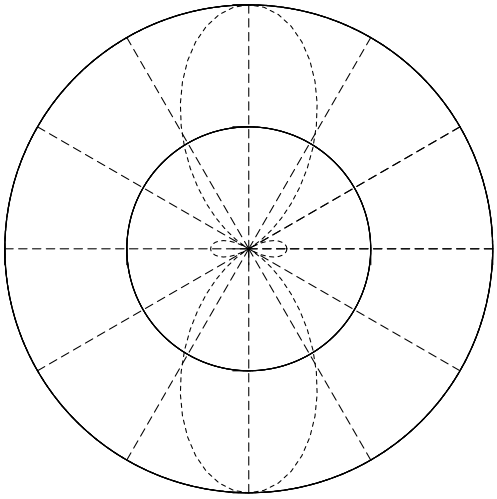

Figure 3.13: Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 500 Hz.

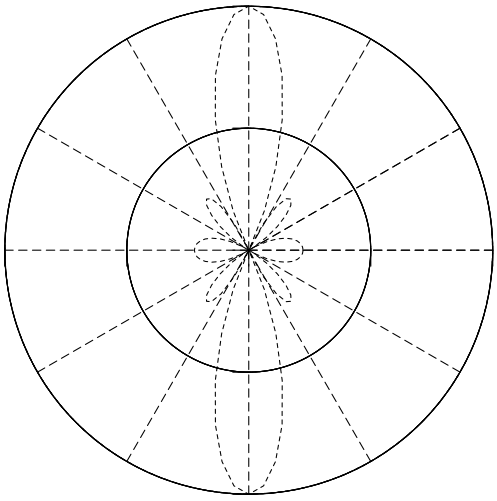

Figure 3.14: Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 1000 Hz.

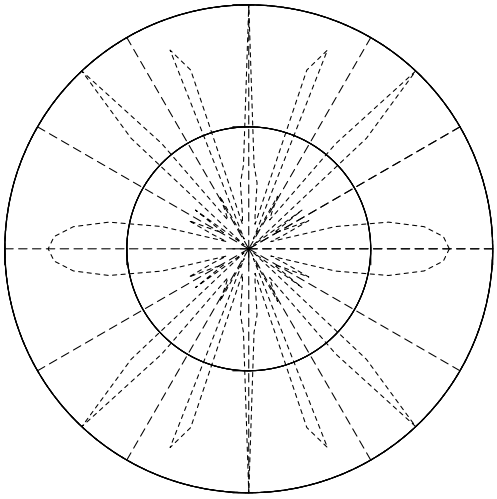

Figure 3.15: Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 5000 Hz.

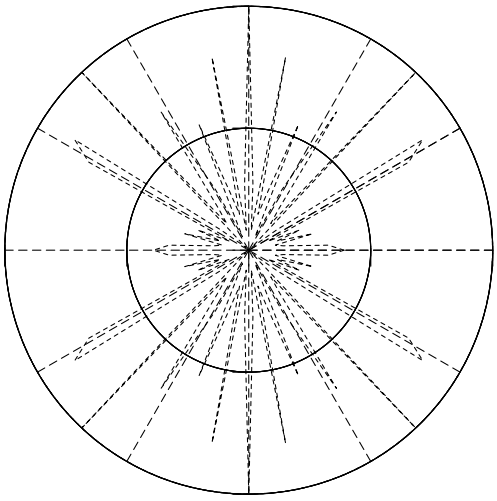

Figure 3.16: Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 10000 Hz.

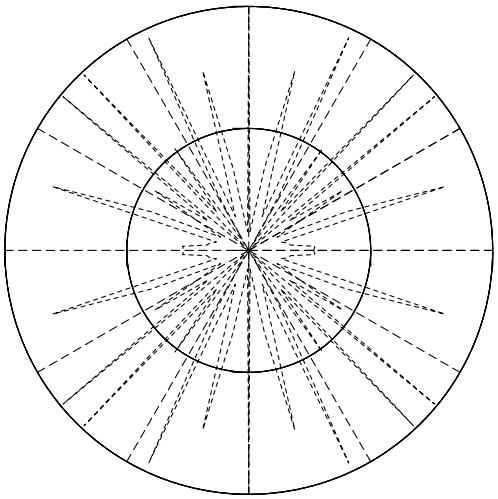

Figure 3.17: Directivity pattern of the linear array formed by 4 sennheiser microphones equispaced 200 mm at 20000 Hz.

Grid inter-spacing	Maximum difference between adjacent points $(*)$
250 mm	8 degrees
150 mm	5 degrees
100 mm	3 degrees
50 mm	1.5 degrees
25 mm	0.8 degrees

$(*)$ Results are approximate and have been calculated with respect to the average distance of any possible speaker in the 223 recordings considered with respect to the origin of coordinates

Table 3.21: HIFI. Maximum difference in degrees between adjacent points as a function of grid inter-spacing.

Some suggested solutions to this problem might be the use of linear arrays with a greater effective length, thus getting a much thiner main lobe, and smaller intermic distances between them (and therefore having a much higher number of elements) so that spatial aliasing is avoided as much as possible, see [AM01] and Figures 3.5.5.2, 3.19, 3.20 and 3.21. As a trade-off for this solution, we will need a much higher processing time, see Section 3.8 in page 123, in order to take into account the information coming from so many microphones. In literature, it has also been suggested the use of a Constant Directivity Beamforming (CDB) technique, see [BW01] pp. 3-17 and Section 2.3.2.3 in page 22. Basically, it consists on a well-designed harmonic array with variable intermic distance so that several different subarrays of equispaced elements can be formed, each of them

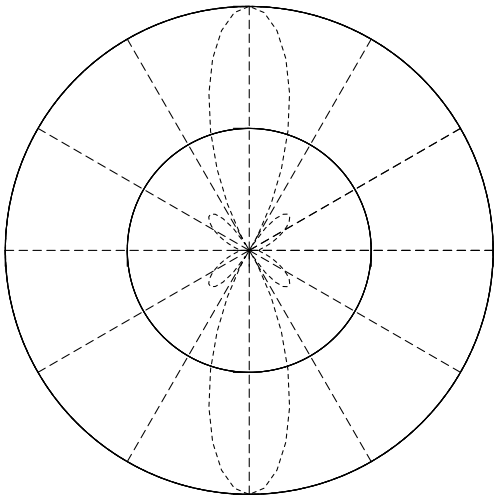

Figure 3.18: Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 1000 Hz.

designed to cover a specific frequency range. When their responses are properly combined after some post-filtering they can offer a directivity pattern with constant response and no-aliasing over wide frequency bands.

3.5.5.3 SONY

In Table 3.22, we show the results for 500 and 150 mm equispaced grids in the case of Sony database, which was chosen to check the effects of the search space division in a simulated corpus.

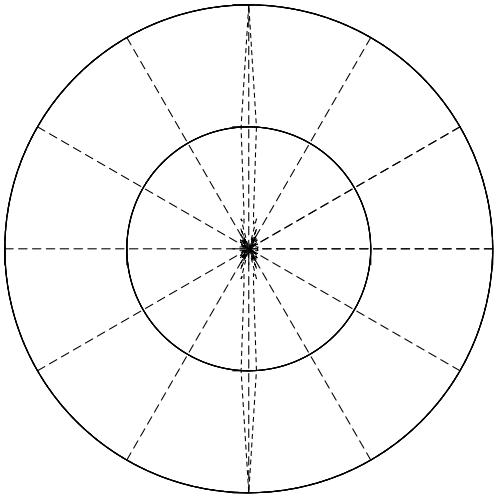

Figure 3.19: Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 5000 Hz.

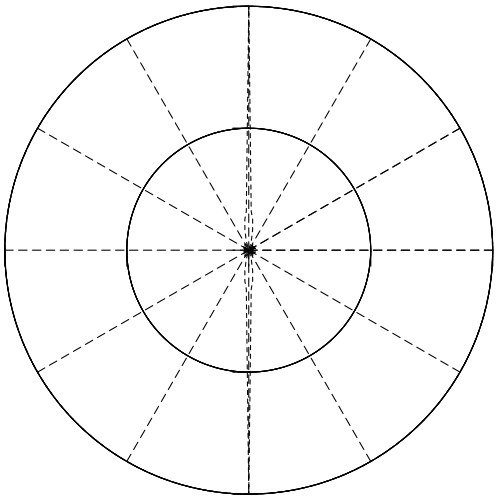

Figure 3.20: Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 10000 Hz.

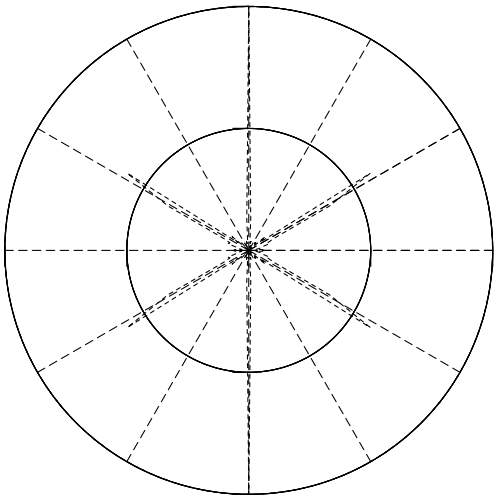

Figure 3.21: Directivity pattern of the linear array formed by 33 sennheiser microphones equispaced 20 mm at 20000 Hz.

	50 mm	150 mm
Pcor	$96.0 \pm 0.1\%$	$0.0 \pm 0.0\%$
Rel. error reduction		-100.0%
Bias fine (x:y:z) [mm]	$-82 : -175 : -349$	$-32 : -48 : -161$
Bias fine+gross (x,y,z) [mm]	$-101 : -149 : -348$	$237 : 429 : -345$
Bias AEE fine [mm] = MOTP	400	256
Rel. AEE reduction		36.0%
Bias fine+gross [mm]	432	632
Rel. BIAS f+g reduction		-46.3%
A-MOTA	$88 \pm 0.1\%$	$-71 \pm 0.1\%$
Rel. error reduction		-180.7%
Loc. frames	213950	635404
Ref. duration (s)	17690.0	18083.0

Table 3.22: Simulated SONY. Grid spacing effect. fs = 48 KHz. Sequences considered= 8740 recordings from 20 different speakers placed at one single, static position (P2). Frame Size= 640 ms

This last result warns us about how crucial a proper space griding can be if the real speaker position "does not fit well" in the search space grid. In the first case, when taking 150 mm equispaced points and due to the static position of the speaker, the algorithm tends to almost always pick a point located at approximately 600 mm from the target,

therefore leading to poor success localization rates. However, if we build a more precise grid with 50 mm equispaced points, the algorithm is able to find a new, closer point, this time located at around 400 mm from the target, therefore leading to a drastic improvement in the localization rate.

3.5.6 Comparison between real and simulated data

In this Section we will compare our algorithm operation under real and simulated conditions. The databases chosen to experiment with were HIFI and its simulated version.

In order to increase the amount of data and conditions under which our algorithm was being tested, we decided to include two simulated databases, SONY and simulated HIFI. Simulating a database basically means to take a close-talk recording of the audio data and then proccess it as if those recordings were in fact uttered at any chosen recording room and captured any chose array geometry inside it. In our particular case, the environment chosen to be simulated was the HIFI recording room located at the Speech Technology Group laboratory in the Technical University of Madrid (UPM). The array geometry inside it consists on 4 linear equispaced sennheiser microphones placed in the front wall of the room and 3 L-shaped crown microphones placed in one corner, see Figure 3.4 in page 65. The simulated recordings obtained at each one of these microphones are the result of adding an attenuated, line-of-sight version of the close-talk recording plus several delayed reverberations of it generated by the method of ray-tracing and the contribution of a background noise, according to the model depicted in Section 2.2.4 in page 11.

3.5.6.1 HIFI

Here in Table 3.23, we compare the Real HIFI and the Simulated HIFI results when captured by the linear array of 4 sennheiser microphones equispaced 200 mm.

	Real HIFI	Simulated HIFI
Pcor	$88.0 \pm 0.7\%$	$54.0 \pm 1.0\%$
Rel. error reduction		-38.6%
Bias fine (x:y:z) [mm]	$-15 : -11 : -27$	$-8 : -19 : -54$
Bias fine+gross (x,y,z) [mm]	$20 : 43 : -28$	$-72 : 453 : -97$
Bias AEE fine [mm] = MOTP	196	143
Rel. AEE reduction		27.0%
Bias fine+gross [mm]	386	715
Rel. BIAS f+g reduction		-85.2%
A-MOTA	$76 \pm 0.9\%$	$9 \pm 0.6\%$
Rel. error reduction		-88.2%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.23: Real HIFI vs. Simulated HIFI. Microphone array: 4 sennheiser microphones equispaced 200 mm. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms. Grid step in which the room was divided: 150 mm

In Table 3.24, we compare the Real HIFI and the Simulated HIFI results when captured by the linear array of 4 sennheiser microphones equispaced 200 mm plus the L-shaped array formed by 3 crown microphones.

	Real HIFI	Simulated HIFI
Pcor	$72.0 \pm 0.9\%$	$70.0 \pm 0.9\%$
Rel. error reduction		-2.8%
Bias fine (x:y:z) [mm]	$-13 : 28 : 22$	$-14 : 3 : -13$
Bias fine+gross (x,y,z) [mm]	$-30 : 59 : 22$	$-25 : 214 : -42$
Bias AEE fine [mm] = MOTP	253	238
Rel. AEE reduction		5.9%
Bias fine+gross [mm]	580	519
Rel. BIAS f+g reduction		10.5%
A-MOTA	$44 \pm 1.0\%$	$40 \pm 1.0\%$
Rel. error reduction		-9.1%
Loc. frames	9390	9390
Ref. duration (s)	496.0	367.0

Table 3.24: Real HIFI vs. Simulated HIFI II. Microphone array: 4 sennheiser microphones equispaced 200 mm plus 3 L-shaped crown microphones. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms. Grid step in which the room was divided: 150 mm

From Tables 3.23 and 3.24 we can derive that our localization algorithm performs better with real than with simulated data. There is, nevertheless, an important difference between the two databases behaviour: while real data shows real poor performances associated to the L-shaped array, the simulated ones seems to work better. We associate this event to the fact that, in the real recording situation, the L-shaped array area, located in a room corner, implies strong and performance-limitting reverberation effects that, on the other hand, are not present in the simulated version.

3.6 Array geometry evaluation

We will test the robustness of our localization estimates under different array geometries. The objetive is to assess the best configuration in terms of number of microphones, intermicrophone distance and array geometry.

3.6.1 Number of microphones

In this Section we will concentrate on the effect that different array configurations, with different number of microphones, have on the algorithm performance.

Now, the focus will be put on just taking into account the effect that a varying number

of microphones has on the performance when keeping the rest of the parameters (database, frame size, inter-space distance, etc.) constant.

3.6.1.1 Simulated HIFI

HIFI database was simulated to be captured, for instance, by a linear array of 33 elements equispaced 20 mm. As shown in Figure 3.5 in page 68, this fact allows us to use different subarray configurations of it including:

- *33 elements linear array*, equispaced 20 mm, effective longitude, L = 660 mm.
- *11 elements linear harmonic array*, the spacing between the elements varies in such a way that allows 4 different configurations of linear subarrays of 5 elements each equispaced 160, 80, 40 and 20 mm respectively.
- *5 elements linear array*, 4 different configurations equispaced 160, 80, 40 or 20 mm respectively. In Table 3.25 we will focus on that one with the maximum equispaced distance, 160 mm, implying an effective length, L = 800 mm.
- *3 elements linear array*, multiple configurations, we will just focus on that one with the maximum equispaced distance, 320 mm, implying an effective length, L = 960 mm.
- *2 elements linear array*, multiple configurations, we will just focus on that one with the maximum equispaced distance, 640 mm, implying an effective length, L = 1280 mm.

In Table 3.25, we expose the results of all these different array configurations presented above. They all have the particularity of being equally long phisically. Their effective lengths, L, are different though. Let's recall that:

$$L = Nd \tag{3.7}$$

where L is the effective length of the array, N is the number of elements in the array and d is the intermic distance between the elements.

	33 mics	11 harmonic mics	5 mics	3 mics	2 mics
Pcor	$90.0 \pm 0.6\%$	$86.0 \pm 0.7\%$	$75.0 \pm 0.9\%$	$73.0 \pm 0.9\%$	$31.0 \pm 0.9\%$
Rel. error reduction		-4.4%	-16.7%	-18.9%	-65.6%
Bias fine (x:y:z) [mm]	$1 : -56 : -20$	$1 : -95 : 8$	$-2 : -35 : -31$	$1 : -24 : -23$	$2 : 0 : -112$
Bias fine+gross (x,y,z) [mm]	$-25 : 138 : -32$	$-12 : 120 : -14$	$7 : -85 : -45$	$5 : -99 : -49$	$-45 : -694 : -142$
Bias AEE fine [mm] = MOTP	103	182	173	164	113
Rel. AEE reduction		-76.7%	-68.0%	-59.2%	-9.7%
Bias fine+gross [mm]	315	425	341	364	1023
Rel. BIAS f+g reduction		-34.9%	-8.3%	-15.6%	-224.8%
A-MOTA	$81 \pm 0.8\%$	$72 \pm 0.9\%$	$50 \pm 1.0\%$	$46 \pm 1.0\%$	$-38 \pm 1.0\%$
Rel. error reduction		-11.1%	-38.3%	-43.2%	-146.9%
Loc. frames	9390	9390	9390	9390	9390
Ref. duration (s)	367.0	367.0	367.0	367.0	367.0

Table 3.25: Simulated HIFI. Array geometry effect I. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

The previous Table 3.25 gives us an interesting hint: as the number of elements of the array increases and as the intermicrophone distance between them decreases we obtain better localization rates. There are two main reasons behind this phenomena: First, the greater the number of elements in the array, the lower the side lobes level in its directivity array and, consequently, the lower the energy coming from directions different to the steered, main-lobe one, see Figure 2.5 in page 17. Secondly, the greater the number of elements in the array when keeping its longitude constant, the lower the intermic distance will be and, consequently, the more the spatial aliasing is avoided, specially at high frequency ranges, those demonstrating to show better performance. More results about these effects can be found in Sections 3.5.5 in page 93 and 3.9.4 in page 143 of this Master Thesis. For a theoretical backgroud check Section 2.3 in page 15 or the McCowan book, [AM01]. However, this effect also has its trade-off in form of a higher computation time as referred in Section 3.8 in page 123.

In Table 3.26, we can now check the results for the Simulated HIFI database when captured by an array having both different number of microphones and different effective length, specifically, the linear array of 4 elements equispaced 200 mm, having a effective length, L = 800 mm, and being the one used in the Real HIFI corpus.

	4 mics
Pcor Rel. error reduction	$71.0 \pm 0.9\%$
Bias fine (x:y:z) [mm]	$4 : -97 : 0$
Bias fine+gross (x,y,z) [mm]	$9 : -172 : 0$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	107
Bias fine+gross [mm] Rel. BIAS f+g reduction	506
A-MOTA Rel. error reduction	$42 \pm 1.0\%$
Loc. frames	9390
Ref. duration (s)	367.0

Table 3.26: Simulated HIFI. Array geometry effect II. Microphone array: Linear array of 4 elements equispaced 200 mm : effective longitude, L= 800 mm. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

In this Table 3.26, we can note an interesting effect: although having a greater number of microphones and a smaller intermicrophone distance between them, the performance

shown is worse than the one obtained with the 3-elements array described in Table 3.25. This event warns out about a new, interesting conclusion: It is also very important to take into account the effective length, L, of the arrays involved, see Figure 2.6 in page 17. Since L is longer in the 3-elements array than in the 4-elements one, the width of the main lobe is wider in this last one. Consequently, the 3-elements array is able to achieve a more accurate pointing to the selected spatial locations.

As we can see, there are several factors having an influence when talking about array geometry effects. Now, just one of them, the effective length, L, will be considered: In Table 3.27, we kept a constant intermic distance of 20 mm and tried with different equispaced linear arrays of 3, 5, 7, 9, 11, 15, 17, 21, 25 and 33 elements, each of them therefore having an effective length 40 mm longer than the previous one. Results can also be closely inspected at Figure 3.22.

	33 (L=660 mm)	25 (L=500 mm)	21 (L=420 mm)	17 (L=340 mm)	11 (L=220 mm)	5 (L=100 mm)
Pcor	$90.0 \pm 0.6\%$	$85.0 \pm 0.7\%$	$54.0 \pm 1.0\%$	$53.0 \pm 1.0\%$	$28.0 \pm 0.9\%$	$26.0 \pm 0.9\%$
Rel. error reduction		-5.6%	-40.0%	-41.1%	-68.9%	-71.1%
Bias fine (x:y:z) [mm]	$1 : -56 : -20$	$1 : -23 : 27$	$-34 : -112 : -57$	$2 : 42 : 21$	$71 : 20 : 0$	$-7 : 99 : -162$
Bias fine+gross (x,y,z) [mm]	$-25 : 138 : -32$	$-25 : 251 : -1$	$-44 : 195 : -50$	$-52 : 732 : -81$	$-14 : -264 : 0$	$-114 : -789 : -159$
Bias AEE fine [mm] = MOTP	103	121	186	185	169	247
Rel. AEE reduction		-17.5%	-80.6%	-79.6%	-64.1%	-139.8%
Bias fine+gross [mm]	315	431	656	932	1087	1191
Rel. BIAS f+g reduction		-36.8%	-108.3%	-195.9%	-245.1%	-278.1%
A-MOTA	$81 \pm 0.8\%$	$71 \pm 0.9\%$	$8 \pm 0.5\%$	$6 \pm 0.5\%$	$-43 \pm 1.0\%$	$-48 \pm 1.0\%$
Rel. error reduction		-12.3%	-90.1%	-92.6%	-153.1%	-159.3%
Loc. frames	9390	9390	9390	9390	9390	9390
Ref. duration (s)	367.0	367.0	367.0	367.0	367.0	367.0

Table 3.27: Simulated HIFI. Number of microphones effect with growing effective length. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

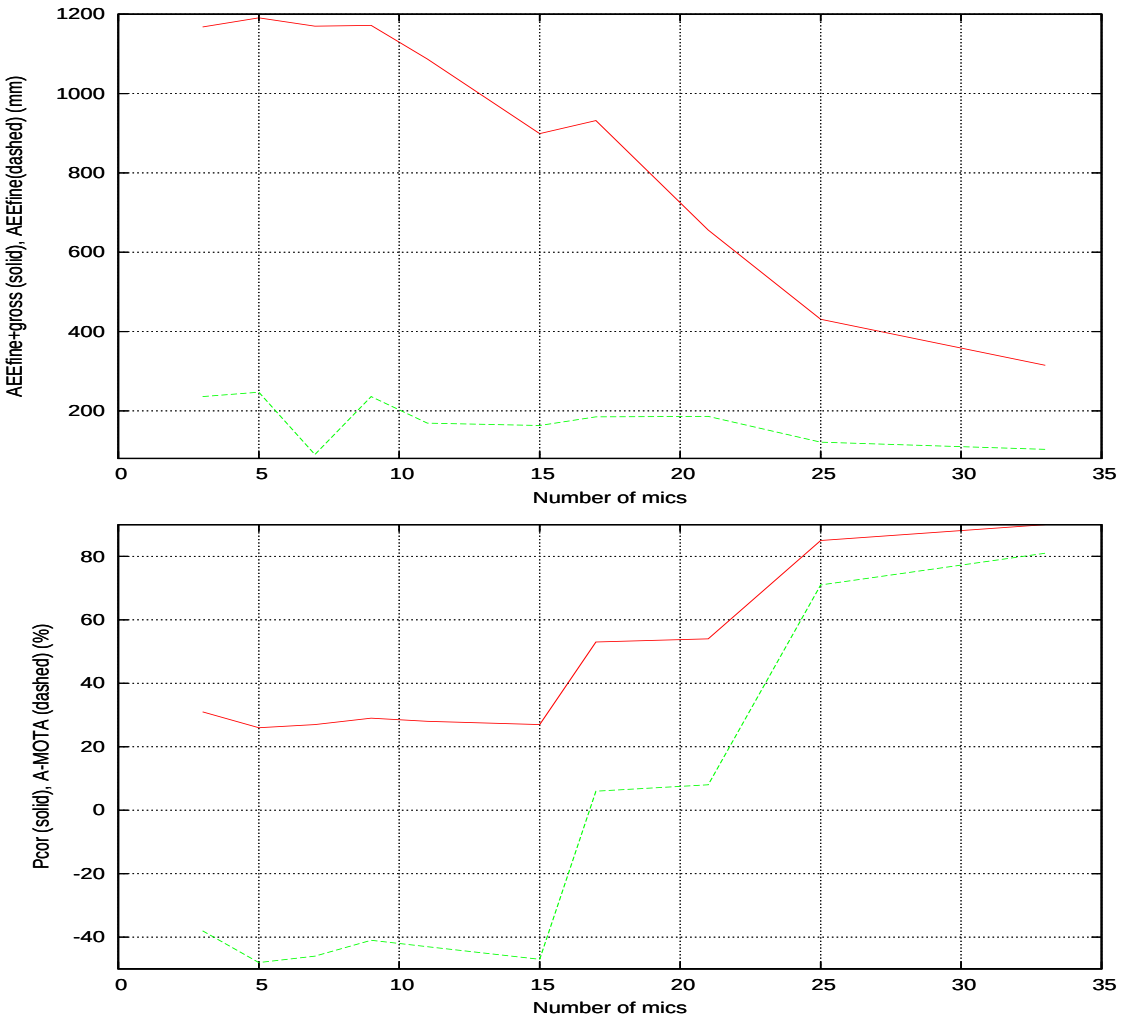

Figure 3.22: Simulated HIFI. Effective length and number of mics effect.

It is curious to see, both in Table 3.27 and Figure 3.22, that localization rates do grow with a growing number of microphones but they just do it in a step function mood, that is to say, for instance, increments from 3 to 15 elements result in approximately equal localization rates but adding an extra pair of microphones, 17 instead of 15, results in an abrupt improvement. The same thing happens when reaching 21 and 25 elements in the array. It seems that there are only significal improvements in the localization rates just when the effective length is long enough as to determine main lobes thin enough as to be capable to precisely distinguish between adjacent localization points (it is important to point out that, as exposed in 3.21 in page 102, the differences in degrees between adjacent points are quite low, specially when talking about 50 mm equispaced localizations as it is

the case).

3.6.1.2 Real HIFI

The real HIFI database, as depicted in Figure 3.4 in page 65, was captured by a linear array of 4 sennheiser elements equispaced 200 mm, placed at the back wall of the room, plus a L-shaped array composed by 3 crown microphones placed at one corner. This basis allows us to try different array configurations in order to check their performance:

- *4+3 array*, aggregated results when taking into account both the 4 sennheiser linear array and the 3 crown L-shaped array.
- *4 sennheiser array*, results when just taking into account the 4 sennheiser linear array.
- *3 crown array*, results when just taking into account the 3 crown L-shaped array.
- *3 sennheiser array*, results when taking into account just 3 of the 200 mm equispaced sennheiser microphones in the linear array. This allows us to directly compare their records with the L-shaped array ones.
- *2 sennheiser array*, results when taking into account just 2 of the 200 mm equispaced sennheiser microphones in the linear array.
- *2 crown array*, results when taking into account just 2 of the crown microphones in the L-shaped array.

In Tables 3.28, 3.29 and 3.30 we expose the results of all these different array configurations presented above. We particularly concentrate on respectively comparing in each one of them the performance of the 2, 3 and 4-elements, sennheiser, linear arrays versus their equivalent crown, L-shaped ones.

	2 senn mics	2 crown mics
Pcor Rel. error reduction	$42.0 \pm 1.0\%$	$0.0 \pm 0.0\%$ -100.0%
Bias fine (x:y:z) [mm]	$-69 : -101 : -157$	$132 : -176 : -175$
Bias fine+gross (x,y,z) [mm]	$115 : -604 : -159$	$-38 : 207 : -159$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	276	281 -1.8%
Bias fine+gross [mm] Rel. BIAS f+g reduction	909	1410 -55.1%
A-MOTA Rel. error reduction	$-15 \pm 0.7\%$	$-100 \pm 0.0\%$ 566.7%
Loc. frames	9375	9375
Ref. duration (s)	496.0	496.0

Table 3.28: Real HIFI. senn vs. crown mics effect II. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm

The idea of using a 3 crown L-shaped array located at one corner of the room was at first believed to be useful since its geometry theoretically allowed to have more precision when determining direction of arrival angles. However, the experimental outputs did not fit the theoretical assumptions. The reason behind this unexpected result may have to do with the power accoustic conditions at the L-shaped array. Since it is located at one corner of the room, it suffers from high reverberation and reflections between its elements signals.

It is also worth to note in the last Table 3.28 that the SRP-PHAT algorithm in that experiment, since we are just taking into account one microphone pair, is totally equivalent to the results GCC-PHAT method would have obtained. Once again, see Section 3.4, it is stated clearly the overwhelming supremacy of SRP compared to GCC.

	3 senn mics	3 crown mics
Pcor	$55.0 \pm 1.0\%$	$11.0 \pm 0.6\%$
Rel. error reduction		-80.0%
Bias fine (x:y:z) [mm]	$-8 : -8 : -94$	$96 : 165 : -148$
Bias fine+gross (x,y,z) [mm]	$83 : -180 : -114$	$-176 : 136 : -139$
Bias AEE fine [mm] = MOTP	259	302
Rel. AEE reduction		-16.6%
Bias fine+gross [mm]	612	1552
Rel. BIAS f+g reduction		-153.6%
A-MOTA	$10 \pm 0.6\%$	$-79 \pm 0.8\%$
Rel. error reduction		-890.0%
Loc. frames	9375	9375
Ref. duration (s)	496.0	496.0

Table 3.29: Real HIFI. senn vs. crown mics effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm

The results taking into account three microphones, that is, three possible microphone pair combinations, confirmed the uneffectiveness of the crown, L-shaped arrays compared to that of the sennheiser, linear ones. It is also important to note that sennheiser microphones offer a much better recording quality than the crown ones.

	4 senn mics	4+3 mics
Pcor Rel. error reduction	$84.0 \pm 0.7\%$	$72.0 \pm 0.9\%$ -14.3%
Bias fine (x:y:z) [mm]	$-7 : -34 : -61$	$-12 : 26 : 22$
Bias fine+gross (x,y,z) [mm]	$50 : 19 : -69$	$-22 : 83 : 22$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	207	250 -20.8%
Bias fine+gross [mm] Rel. BIAS f+g reduction	418	580 -38.8%
A-MOTA Rel. error reduction	$69 \pm 0.9\%$	$45 \pm 1.0\%$ -34.8%
Loc. frames	9375	9375
Ref. duration (s)	367.0	496.0

Table 3.30: Real HIFI. Array geometry effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm

Finally, as expected from the previous experiments and as reflected in Table 3.30, we can appreciate the localization performance decreases when using both arrays, that is to say, the 4+3 array. This is due to the poor performance of the L-shape array as well as a result of mixing, in the same localization process, different types of microphones and different types of data recorded under different conditions.

3.6.2 Intermicrophone distance

Likewise as done in Section 3.6.1, we can also keep constant the number of microphones composing the linear array and focus on just varying the intermicrophone distance between them to see how this single parameter affects the localization.

3.6.2.1 Simulated HIFI

Here, simulated HIFI database was chosen to be captured by different array configurations. This allows us to perform experiments on the four different, equispaced, linear subarrays that can be formed out of the 11 elements harmonic array, each of them composed by 5 sennheiser microphones as described above in page 109. The results of these experiments were reflected in Table 3.31 where we compare their performance when having an intermicrophone distance of 160, 80, 40 and 20 mm respectively.

	160 mm (L=800 mm)	80 mm (L=400 mm)	40 mm (L=200 mm)	20 mm (L=100 mm)
Pcor	$75.0 \pm 0.9\%$	$33.0 \pm 1.0\%$	$27.0 \pm 0.9\%$	$26.0 \pm 0.9\%$
Rel. error reduction		-56.0%	-64.0%	-65.3%
Bias fine (x:y:z) [mm]	$-2 : -35 : -31$	$6 : -47 : 5$	$3 : -1 : -162$	$-7 : 99 : -162$
Bias fine+gross (x,y,z) [mm]	$7 : -85 : -45$	$-42 : 30 : -91$	$-81 : -606 : -144$	$-114 : -789 : -159$
Bias AEE fine [mm] = MOTP	173	102	174	247
Rel. AEE reduction		41.0%	-0.6%	-42.8%
Bias fine+gross [mm]	341	709	1237	1191
Rel. BIAS f+g reduction		-107.9%	-262.8%	-249.3%
A-MOTA	$50 \pm 1.0\%$	$-34 \pm 1.0\%$	$-47 \pm 1.0\%$	$-48 \pm 1.0\%$
Rel. error reduction		-168.0%	-194.0%	-196.0%
Loc. frames	9390	9390	9390	9390
Ref. duration (s)	367.0	367.0	367.0	367.0

Table 3.31: Simulated HIFI. Intermic distance effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

As we can see in Table 3.31 and following the suggestions reflected in Figure 3.6 in page 73, the bigger the distance between the microphones is, the better the localization rate will be. There are two reasons behind this phenomenon, first, as the intermicrophone distance increases, the total effective length of the linear array considered, L, also rises and, therefore, the beamwidth associated to its directivity pattern becomes thiner finally leading to a more accurate focus on the selected spatial point while performing beamforming. Secondly, as the distance between microphones becomes bigger, the time delay differences between them also increase consequently minimizing the rounding effect when translating the real units time delays in seconds to integer units time delays in samples.

3.7 Speaker position influence

We are interested in measuring what impact the speaker position has on the localization algorithm performance. In principle, the bigger the signal time delay differences between microphones are, the better our system can precisely sort out the speaker location. In fact, too small time delay differences among microphones imply a significant, random rounding effect when trying to translate real time units into discrete, digital units. Therefore, in principle, the ideal locations are either those lying as asymmetrically from the microphone array as possible or those being perpendicular to it as it is explained in the theoretical introdution to this Master Thesis at the end of Section 2.4.1.3 in page 29.

3.7.1 Real HIFI

The HIFI database election in this case was due to the fact that its possible speaker locations are restricted to just 5 sites distributed simetrically around the array forming different angles with respect to it, see 3.3.2 in page 64. The following Table 3.32 shows the different localization rates when individually referred, respectively, to just one out of the five possible recording positions in HIFI database.

The positions considered: P1, P2, P3, P4 and P5 are the recording positions distributed around the recording array in the Edecan Room as described in Figure 3.4 in page 65. Let's remind some facts about these positions properties in order to explain the results obtained:

- P1 and P5, are symmetric positions with respect to the center of the array. They are both separated 1848 mm from it and are the most tilted positions, forming an angle of approximately 30 degrees with the array plane.

- P2 and P4, are symmetric positions with respect to the center of the array. They are both separated 1963 mm forming an angle of approximately 60 degrees with the array plane.
- P3, is the perpendicular and more distant position with respect to the array plane.

	Position 3	Position 1	Position 5	Position 2	Position 4
Pcor	$92.0 \pm 1.2\%$	$90.0 \pm 1.3\%$	$88.0 \pm 1.5\%$	$84.0 \pm 1.7\%$	$85.0 \pm 1.6\%$
Rel. error reduction		-2.2%	-4.3%	-8.7%	-7.6%
Bias fine (x:y:z) [mm]	$-61 : -29 : -154$	$-107 : 16 : -25$	$84 : 6 : 59$	$-27 : 13 : -43$	$42 : -59 : 34$
Bias fine+gross (x,y,z) [mm]	$-28 : -185 : -142$	$-227 : 142 : -24$	$315 : 149 : 45$	$-115 : 103 : -38$	$167 : 22 : 25$
Bias AEE fine [mm] = MOTP	187	217	197	168	207
Rel. AEE reduction		-16.0%	-5.3%	10.2%	-10.7%
Bias fine+gross [mm]	344	393	478	334	386
Rel. BIAS f+g reduction		-14.2%	-39.0%	2.9%	-12.2%
A-MOTA	$83 \pm 1.7\%$	$80 \pm 1.8\%$	$75 \pm 2.0\%$	$68 \pm 2.2\%$	$71 \pm 2.0\%$
Rel. error reduction		-3.6%	-9.6%	-18.1%	-14.5%
Loc. frames	1970	1929	1785	1761	1945
Ref. duration (s)	77.0	75.0	70.0	69.0	76.0

Table 3.32: Real HIFI. Position effect. fs = 48 KHz. Sequences considered: Recordings from 12 different speakers placed at static positions P1 (44 recordings), P2 (43 recordings), P3 (45 recordings), P4 (45 recordings) and P5 (46 recordings). Frame Size= 320 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 150 mm

We can observe that the best localization rates correspond to those of the P3. This is due to the fact that it is centered with respect to the array. Consequently the signals uttered from this position will have likewise time delay arrivals to all the microphones in the array. Thus, time delay differences between the microphone pairs will tend to be zero and the Steered Response Power (SRP) algorithm will tend to focus the main lobe in the perpendicular, correct direction. The second best localizations are P1 and P5. They both have similar rates as it was natural to suppose given their symmetric nature. Although being not centered and quite close to the array, their localization rates are comparable to that of the P3. This may be due to two facts: On the first place, we must take into account the signal to noise ratio influence, lower in P3. Secondly, being the more tilted positions means asymmetric times of arrivals to the microphones, that is to say, the closest microphone to P1, for instance, will receive its signal much earlier than the last one, therefore leading to high time differences of arrival which turns out to be benefitial as demonstrated in equation 2.43 in page 31. To end with, the worst positions result to be P2 and P4, they are not centered, thus requiring a proper choice of the angle to steer at, and not as tilted as to create big time delay differences in the microphone array so that this choice can be done accurately.

3.8 Computational demands

We must not forget the need of reaching real-time computation performance in our algorithms in order to be able to apply them to real life solutions. If we analyze the SRP algorithm, there are four main factors affecting its execution time.

- *Time-domain vs. frequency-domain beamforming*, as explained in the theoretical introduction in Section 2.4.3.3 in page 37, both methods are equivalent in theory. When the time comes to apply them to practise two main differences between them arise: First, the frequency-domain method allows to directly use the real time delay units in seconds while the time-domain one forces a rounding conversion to integer sample units consequently implying a loose of precision in the localization. Second, the frequency-domain method implementation implies a large amount of complex multiplications (as many as the FFT size times the total number of microphone pair combinations times the total number of points where to focus). On the other hand, the time-domain method is much lighter since it substitutes this large amount of operations by IFFT transforms (as many as microphone pair combinations) and simple sumations (as many as the total number of points where to focus).
- *Frame size*, its effect is significant in the execution time required as long as increasing the length of the frames processed eventually forces to an increase in the number of

points of the FFT transforms involved.

- *Number of microphones*, as the number of microphones involved raises we have more and more possible microphone pairs to consider. Let's remind the total number of possible combinations of two elements out of a set of N elements:

$$\binom{N}{2} = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2} \tag{3.8}$$

  The number of operations implied will increase by this same factor.

- *Grid spacing*, the way in which we divide the search space has an important influence on the run time since it defines the final set of points where our array will focus in search of the speaker localization. The bigger the number of points taken into account, the longer the computations will take.

The effect (both in performance and run time) of FSRP (frequency-domain SRP) vs. TSRP (time-domain SRP) implementations can be checked in Table 3.33, whose experiments are referred to the Real HIFI database.

The effect (both in performance and run time) of the Frame/FFT size can be checked in Table 3.34 as well as in Figure 3.9 in page 3.9 in the case of ST-AV16.3, with frame sizes ranging from 40 to 1000 ms and FFT sizes varying from 1024 to 16384 points.

The effect (both in performance and run time) of the number of microphones can be checked in Table 3.35, whose experiments are referred to the Simulated HIFI databases and range from the use of 2 to 33 microphones.

Finally, the effect (both in performance and run time) of the grid spacing can be checked in Table 3.36, whose experiments are referred to the MV-AV16.3 database and range from 250 mm inter-spacing (equivalent to 540 different spatial locations) to 50 mm (equivalent to 56867 spatial locations).

The general conclusion we can appreciace is a clear trade-off between the algorithm efficiency and robustness vs. the required CPU time spent.

	Frequency SRP	Time SRP
Pcor Rel. error reduction	$85.0 \pm 0.7\%$	$78.0 \pm 0.8\%$ -46.7%
Bias fine (x:y:z) [mm]	$-66 : -23 : -20$	$-79 : 13 : -32$
Bias fine+gross (x,y,z) [mm]	$-32 : 72 : -24$	$-28 : 80 : -45$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	231	217 6.1%
Bias fine+gross [mm] Rel. BIAS f+g reduction	444	506 -14.0%
A-MOTA Rel. error reduction	$71 \pm 0.9\%$	$56 \pm 1.0\%$ -51.7%
Loc. frames	9390	9390
Ref. duration (s)	496.0	496.0
Run time (real-time units) Rel. run-time reduction	62.00	0.25 -24700.0%

Table 3.33: Real HIFI. SRP vs. FSRP computational load. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 320 ms : FFT Size= 16384 at fs = 48 KHz. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 250 mm : 429 locations

	Frame size= 640 ms	320 ms	160 ms	80 ms	40 ms
Pcor	$94.0 \pm 0.5\%$	$87.0 \pm 0.8\%$	$84.0 \pm 0.8\%$	$83.0 \pm 0.9\%$	$81.0 \pm 0.9\%$
Rel. error reduction		-116.7%	-166.7%	-183.3%	-216.7%
Bias fine (x:y:z) [mm]	$62 : 19 : 63$	$62 : 18 : 65$	$60 : 18 : 64$	$61 : 18 : 65$	$58 : 17 : 63$
Bias fine+gross (x,y,z) [mm]	$7 : -45 : 54$	$-62 : -155 : 46$	$-82 : -208 : 40$	$-72 : -196 : 42$	$-76 : -174 : 39$
Bias AEE fine [mm] = MOTP	190	191	192	194	197
Rel. AEE reduction		-0.5%	-1.1%	-2.1%	-3.7%
Bias fine+gross [mm]	291	423	482	481	490
Rel. BIAS f+g reduction		-45.4%	-65.6%	-65.3%	-68.4%
A-MOTA	$87 \pm 0.8\%$	$74 \pm 1.0\%$	$68 \pm 1.1\%$	$65 \pm 1.1\%$	$61 \pm 1.1\%$
Rel. error reduction		-100.0%	-146.2%	-169.2%	-200.0%
Loc. frames	7295	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0	600.0
Run time (real-time units)	3.38	1.80	0.88	0.48	0.30
Rel. run-time reduction		-87.8%	-284.1%	-604.2%	-1026.7%

Table 3.34: ST-AV16.3. Frame size computational load effect. Microphone array: Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm : 56867 locations

	33 mic (L=660mm)	11 harmonic mic	5 mic (L=800mm)	3 mic (L=960mm)	2 mic (L=1280mm)
Pcor	$90.0 \pm 0.6\%$	$86.0 \pm 0.7\%$	$75.0 \pm 0.9\%$	$73.0 \pm 0.9\%$	$31.0 \pm 0.9\%$
Rel. error reduction		-40.0%	-150.0%	-170.0%	-590.0%
Bias fine (x:y:z) [mm]	$1 : -56 : -20$	$1 : -95 : 8$	$-2 : -35 : -31$	$1 : -24 : -23$	$2 : 0 : -112$
Bias fine+gross (x,y,z) [mm]	$-25 : 138 : -32$	$-12 : 120 : -14$	$7 : -85 : -45$	$5 : -99 : -49$	$-45 : -694 : -142$
Bias AEE fine [mm] = MOTP	103	182	173	164	113
Rel. AEE reduction		-76.7%	-68.0%	-59.2%	-9.7%
Bias fine+gross [mm]	315	425	341	364	1023
Rel. BIAS f+g reduction		-34.9%	-8.3%	-15.6%	-224.8%
A-MOTA	$81 \pm 0.8\%$	$72 \pm 0.9\%$	$50 \pm 1.0\%$	$46 \pm 1.0\%$	$-38 \pm 1.0\%$
Rel. error reduction		-47.4%	-163.2%	-184.2%	-626.3%
Loc. frames	9390	9390	9390	9390	9390
Ref. duration (s)	367.0	367.0	367.0	367.0	367.0
Run time (real-time units)	40.00	11.80	1.13	0.38	0.18
Rel. run-time reduction		-239.0%	-3439.8%	-10426.3%	-22122.2%

Table 3.35: Simulated HIFI. Number of mics computational load effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: The physical longitude of all the arrays compared is 660 mm, their effective lenghts are different though. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm : 37180 locations

	56867 loc (50 mm)	7770 (100 mm)	2450 (150 mm)	540 (250 mm)
Pcor	$77.0 \pm 2.7\%$	$76.0 \pm 2.8\%$	$69.0 \pm 3.0\%$	$70.0 \pm 3.0\%$
Rel. error reduction		-4.3%	-34.8%	-30.4%
Bias fine (x:y:z) [mm]	$-2 : 107 : 26$	$17 : 93 : 40$	$10 : 86 : 41$	$19 : 61 : 3$
Bias fine+gross (x,y,z) [mm]	$-144 : -178 : -23$	$-129 : -181 : -8$	$-118 : -185 : 9$	$-97 : -213 : -33$
Bias AEE fine [mm] = MOTP	227	236	259	262
Rel. AEE reduction		-4.0%	-14.1%	-15.4%
Bias fine+gross [mm]	577	585	636	630
Rel. BIAS f+g reduction		-1.4%	-10.2%	-9.2%
A-MOTA	$53 \pm 3.2\%$	$51 \pm 3.2\%$	$37 \pm 3.1\%$	$39 \pm 3.2\%$
Rel. error reduction		-4.3%	-34.0%	-29.8%
Loc. frames	917	917	917	917
Ref. duration (s)	42.0	42.0	42.0	42.0
Run time (real-time units)	6.85	3.55	3.38	3.15
Rel. run-time reduction		-93.0%	-102.7%	-117.5%

Table 3.36: MV-AV16.3. Grid spacing computational load effect. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each

3.9 Evaluation of aditional strategies

3.9.1 Coarse to fine strategy

For the moment, our SRP-PHAT algorithm always begins with creating a grid in our room that defines a serie of points equispaced a given value. The microphone array will later point at each of these points in search of the one with the highest power value.

As seen in Table 3.19 in page 95, the finer the grid in which we divide the room is, the better localization results we obtain. Nevertheless, this fact implies a growing computacional time as seen in Table 3.36, since a finer grid means a higher number of spatial points to compute and evaluate.

Therefore, we then aimed at implementing a new technique that could allow us to take advantage of the nice results of a fine grid without having to neccessarily spend more computational time. We made use of Zotkin and Duraiswami's proposal in [RDD01] and [ZD04]. Based on the observation that the wavelengths of the sound from a speech source are comparable to the dimensions of the space being searched and that the source is broadband, they developed an efficient algorithm performing a hierarchical search of the Steered Response Power (SRP) from a coarse level to a fine one that promised significant speedups by using this coarse-to-fine strategy in both space and frequency. More details about this technique can be found in the theoretical introduction of this Master Thesis in Section 2.4.4.1 in page 39.

Eventually, we decided to add this new technique to our algorithm and evaluate how it worked. In the first step, we can select an appropiate, relatively low cut-off frequency that will define a small number of gross search areas according to the frequency-spatial relatioship showed above. Once we have selected the gross area containing the strongest speech energy, we further divide it into 8 equispaced cubes, called octrees, and explore them themselves with a cut-off frequency twice as big as the previous one (since the areas to explore have become twice as small). We can continue like that until we get to search areas as fine and precise as desired. However, the computational time implied in order to get to these fine levels should not be here a drawback since we are discarding from the beginning large areas of space and just concentrate on accurately exploring those ones which are more likely to contain the speaker according to the previous algorithm steps.

After implementing this new technique according to Zotkin and Duraiswami's baselines, some experiments were carried out to check its results, but these turned out to be highly discouraging. The localization rates dropped dramatically as the result of our algorithm selecting the wrong gross area of space from the very starting steps. This led us to think there was something wrong with the frequency-spatial properties we had assu-

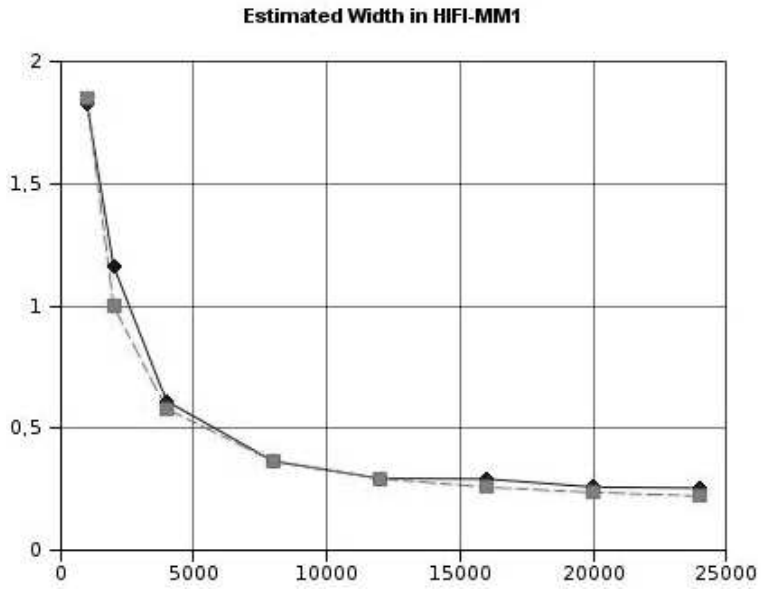

Figure 3.23: Experimental results (solid) and curve-fitting (dashed) of our array configuration peak width as a function of frequency

med from Zotkin and Duraiswami papers. Probably their array configuration properties do not apply to our own experimental conditions. We then decided to stablish our own hypothesis about the relationship between the cut-off frequency applied to the source and the consequent width of the explored region.

We then implemented a simple beamformer and closely analyzed a HIFI recording of a single, static speaker whose position was known and which captured by the uniform linear array configuration of 4 microphones used in this Master Thesis. We centered around different cut-off frequencies and sucesively pointed, first, directly to the speaker position and then slowly moving further from it. In each of these steps we plotted the speech energy received by our beamformer and set the peak width to be the distance between those two points distributed around the speaker position in which the power had descended 3 dB. The curve obtained, depicted in Figure 3.23, was different to that of Zotkin and Duraiswami. Curve fitting led us to our own frequency-spatial relationship:

$$b \simeq 0.15 + 5\lambda \tag{3.9}$$

New experiments were performed to check the validity of our new assumption, leading to more promising, but yet insufficient, results. Next in Table 3.37, we will show and compare some of them to their approximately equivalent non-coarse-to-fine experiments in terms of finesse. Both their localization performance and the computational time required will also be measured.

3.9.1.1 HIFI

HIFI database was chosen for the experimentation since this was the corpus whose frequency-spatial relationship, depicted in Figure 3.23, was most widely studied during this Master Thesis. AV16.3 frequency-spatial relationship was also studied but its uniform circular arrays did not offer such a regular behaviour in this sense as that showed by the uniform linear array in HIFI.

	Coarse-to-fine	No coarse-to-fine
Pcor	$58.0 \pm 1.0\%$	$71.0 \pm 0.9\%$
Rel. error reduction		22.4%
Bias fine (x:y:z) [mm]	$-47 : 16 : 1$	$-18 : -64 : -68$
Bias fine+gross (x,y,z) [mm]	$-144 : 184 : -3$	$19 : -257 : -84$
Bias AEE fine [mm] = MOTP	239	193
Rel. AEE reduction		19.2%
Bias fine+gross [mm]	736	555
Rel. BIAS f+g reduction		24.6%
A-MOTA	$16 \pm 0.7\%$	$41 \pm 1.0\%$
Rel. error reduction		156.2%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.37: Real HIFI. Coarse to fine strategy. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 500 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm. Grid step in which the room was divided: 100 mm

These experiments depicted in Table 3.37 made us think about two main conclusions:

- The localization performance of the coarse to fine algorithm is clearly lower than the traditional method. This is due to the fact that we are still commiting a high rate of mistakes in the first step of the method, when choosing the gross area in which to concentrate our search. This may be due to two facts: In the first place, we must not forget we are using a band-limitted, low-pass filtered version of the signal in order to be able to explore at once these relatively big areas, thus, a sensible part of the signal information is not used during this localization first step. Secondly, we also must remind we are using an array of just four elements equispaced d= 200 mm. This fact implies a strong spatial aliasing starting from relatively low frequencies as cited in [AM01] and depicted in Figures 3.13, 3.14, 3.15, 3.16 and 3.17 in page 100.

$d < \frac{\lambda_{min}}{2}$: No aliasing

$d = 0.2$ m $=: \lambda_{min} > 0.4$ m $: f_{min} = \frac{345\frac{m}{s}}{0.4m} = 862.5Hz$

That is to say, using versions of the source signal containing frequencies higher than approximately 900 Hz will imply a poor steering of our array. When pointing its main lobe to a certain point in space, several replicas of it will appear in different directions, therefore integrating energy coming from different search areas and misleading the obtained results. We could then think of using frequencies lower than this 900 Hz in the first coarse to fine step but, according to our $b \simeq 5\lambda$ ratio, that would define unnacceptably gross areas (cubes around 2 meters long, comparable to the dimensions of the whole room and, thus, useless).

- The computational time spent with our implementation of the coarse to fine algorithm is comparable, even slowly higher, than the traditional method. This is due to the fact that, although having to explore a significally smaller set of points, every exploration implies several correlation and IFFT processes, one for every step in the hyerarchical search until getting to the finer level, instead of the one-step scheme traditionally used.

Despite these dissapointing conclusions, we are still optimistic about the possibilities of this coarse to fine method. One of his most appealing properties, apart from the theoretical computational time save, is the fact that it minimizes the negative effect of spatial aliasing: no matter how close our microphones are in the array, its directivity pattern will eventually present replicated main lobes at different directions if the frequencies used are too high, or to put it in an equivalent way, if the areas explored are too fine. However, this problem, although inevitable with the traditional SRP algorithm, has almost no effect with a proper coarse to fine algorithm. As the search gets finer and finer and the aliasing becomes more likely to appear, we concentrate on more and more reduced pieces of space and, then, the aliased energy contributions coming from different spatial areas to the one selected will not mislead our results, that is to say, never a location coming from this aliased, wrong directions of the room could be selected as our system final estimation since those parts were already discarded during the first, non-aliased, steps of the technique.

What is more, and regarding the two inconvenient conclusions depicted above, there are some promising factors that may help to improve them in the future:

- *Localization performance*,

  The use of a better microphone array, with a better directivity pattern and a better behaviour with high frequencies will surely lead to improved results. The

larger the effective lenght of the array is, in order to get a more accurate main lobe, and the closer the microphones are between them, in order to avoid aliasing, the better the results will be.

In addition to this improvement, we can think of a simpler, but yet effective, coarse to fine algorithm in just two steps as described in [AG07], instead of implementing a multiple-steps coarse to fine algorithm as proposed in [ZD04]. With this new scheme, in the first step, we explore relatively gross areas with an appropiate cut-off frequency. In the second step we simply apply a fine grid to the reduced, selected area. The grid applied could be much finer than the one used in the traditional method as it takes into account a much smaller search area. This would yield better localization results without increasing the computational time required: The number of points finally evaluated in the two-steps can be comparable or even lower than the number of points to evaluate in the whole room even when dividing it with worse finesse.

- *Computational load*, the coarse to fine execution time could be lowered with an optimized implementation of the algorithm. For instance, since the signal is often filtered in order to search grosser areas of the space we do not neccessarily need to make full-length IFFT every step. We could instead make use of the cut-off frequency in order to define an IFFT size as small as possible in each of the steps. That would help us reducing the computational time required by the scheme.

3.9.2 Noise masking

As proposed by DiBiase in [Dib00] pp. 28-34, it is advantageous to take into account the Signal to Noise Ratio (SNR) since those frames with a significant amount of noise will lead to non-sense localization estimates. Following this idea, we implemented a flag in our system that, when activated, triggers a noise masking strategy in which speech samples lying underneath a certain noise-dependent threshold will not be taken into account during the localization computations. According to the theoretical introduction presented in Section 2.4.4.2 in page 40, this threshold can be designed according to two different strategies: fixed and adaptive.

In this Section, we willl try to measure how big the noise influence on our localization algorithm can be. In order to do so, we will use the more noisy database: HIFI. In addition, we will evaluate the performance of this noise masking strategy under its fixed and adaptive focusses.

3.9.2.1 Real HIFI

Real HIFI database was chosen in this Section as its recording conditions showed to be the most noisy ones among all the available databases. Here we will then test how well this noise masking strategy works. What is more, we will also try to assess how to design the best threshold under which start to discard speech samples. Table 3.38 compares experimental results without noise masking to those with different *fixed* thresholds. In Table 3.39 the *adaptive* threshold strategy is tested. We start by just discarding those speech samples lying underneath the noise level to, later on try and see what happens if we raise this adaptive threshold and keep just those speech frequency samples whose power exceeds in 12, 24 or even 36 dB the noise power.

	Without Noise Masking	Fixed threshold + 12dB	Fixed threshold + 24dB
Pcor	$84.0 \pm 0.7\%$	$83.0 \pm 0.8\%$	$76.0 \pm 0.9\%$
Rel. error reduction		-1.2%	-9.5%
Bias fine (x:y:z) [mm]	$-7 : -34 : -61$	$1 : -28 : -55$	$-2 : -22 : -51$
Bias fine+gross (x,y,z) [mm]	$50 : 19 : -69$	$89 : 118 : -66$	$105 : 163 : -64$
Bias AEE fine [mm] = MOTP	207	198	207
Rel. AEE reduction		4.3%	-0.0%
Bias fine+gross [mm]	418	479	559
Rel. BIAS f+g reduction		-14.6%	-33.7%
A-MOTA	$69 \pm 0.9\%$	$65 \pm 1.0\%$	$52 \pm 1.0\%$
Rel. error reduction		-5.8%	-24.6%
Loc. frames	9375	9375	9375
Ref. duration (s)	367.0	367.0	367.0

Table 3.38: Simulated HIFI. Fixed-threshold noise masking effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm

	Adaptive threshold+0dB	Adapt. threshold+12dB	Adapt. threshold+24dB	Adapt. threshold+36dB
Pcor	$84.0 \pm 0.7\%$	$84.0 \pm 0.7\%$	$85.0 \pm 0.7\%$	$83.0 \pm 0.8\%$
Rel. error reduction		0.0%	1.2%	−1.2%
Bias fine (x:y:z) [mm]	$-7 : -34 : -61$	$-7 : -34 : -61$	$-6 : -33 : -61$	$-0 : -29 : -52$
Bias fine+gross (x,y,z) [mm]	$50 : 20 : -70$	$52 : 19 : -70$	$54 : 15 : -68$	$53 : 36 : -61$
Bias AEE fine [mm] = MOTP	206	206	207	202
Rel. AEE reduction		−0.0%	−0.5%	1.9%
Bias fine+gross [mm]	419	417	417	443
Rel. BIAS f+g reduction		0.5%	0.5%	−5.7%
A-MOTA	$69 \pm 0.9\%$	$69 \pm 0.9\%$	$69 \pm 0.9\%$	$66 \pm 1.0\%$
Rel. error reduction		0.0%	0.0%	−4.3%
Loc. frames	9375	9375	9375	9375
Ref. duration (s)	367.0	367.0	367.0	367.0

Table 3.39: Simulated HIFI. Adaptive threshold noise masking effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm. Frame Size= 640 ms. Grid step in which the room was divided: 150 mm

As we can see, adopting a fixed threshold strategy demonstrates to be a wrong option since it deals with all samples in the same way without discriminating their frequency particularities. We must not forget that human speech quickly loses power as the frequency increases. Therefore, when averaging the noise power over all frequencies and taking this value as universal reference threshold, a large part of the high frequency components will be discarded thus leading to poorer localization estimates. In fact, high frequency sounds are the most likely to conduct to proper localization estimates as they present a much better behaviour as far as the array directivity pattern and beamwidth are concerned, see Section 3.9.4 in page 143. In other words, the higher the frequency component is, the thiner its associated beamwidth will be, thus resulting in a more accurate pointing to the selected search points.

On the other hand, a frequency-adaptive threshold strategy demonstrates to be more reasonable. However, it requires a proper tuning of the limit under which start discarding samples. In Table 3.39, we can observe in the first case that eliminating just those speech samples lying under the noise power yields almost the same results as if we were not applying a noise masking strategy. The same thing happens even when we choose to keep just those samples 12 dB above the noise level. The bound chosen in both cases demonstrates to be too low and finally results in an insufficient number of low power samples being eliminated as to affect the localization results. It is only when we keep just those samples raising 24 dB over the noise level that we obtain some slight improvement performance, we are able to localize better since we only take into account those significant parts of the signal which are clearly above the background noise level. Although the improvement is not spectacular in this particular experiment, this tool must be seriously taken into account when working in heavy noise conditions. Finally, as the last case demonstrates, if we choose to raise our threshold too much (36 dB over the noise level) we will begin to also discard valuable parts of our signal and get worse localization estimates in consequence.

3.9.3 Estimation of localization confidence

Some of the previous experiments suggested us the idea that the longer an array is, the better its localization estimates will be, see Sections 3.4 and 3.6 in pages 71 and 108 respectively. A longer array will be able to point more accurately to the selected search points. What is more, the more separated the microphones are within the array, the greater the time delay differences between them will be, thus leading to smaller rounding errors when translating real units in seconds to integer units in samples.

Now, we can make use of this a priori information in order to improve our localization rates. As explained above, the more a microphone pair is separated, the more likely is

to yield, in theory, a proper localization. We should then obtain some improvements by weighting the different contributions of all the microphone pairs according to the distance between them.

As demonstrated by Dibiase in [Dib00] pp. 78-80 and exposed in Section 2.4.3.2 in page 36, the final Steered Response Power (SRP) got when pointing to a certain spatial location is equivalent to the sumation, at the proper time-delay, of all the Generalized Cross Correlations (GCC) coming from all the possible mic pair combinations. Precisely, we will apply our weigthing to this sumation: each mic pair GCC will be multiplicated by a weigthing factor related to the distance between them. The mathematical expressions derived are:

- *Non-weigthed SRP algorithm:*

$$P = 2\Pi \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}(\tau_{ij}) \tag{3.10}$$

  where P is the SRP power estimation for a certain location, N is the total number of microphones in the array, $c_{ij}(\tau)$ is the Generalized Cross Correlation (GCC) function between mics i and j and τ_{ij} is the time delay difference between mics i and j form a certain spatial location.

- *Weigthed SRP algorithm:*

$$P = 2\Pi \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} c_{ij}(\tau_{ij}) \tag{3.11}$$

  where α_{ij} is a weigthing factor in the range (0,1] computed as follows:

$$\alpha_{ij} = \frac{d_{ij}}{d_{max}} \tag{3.12}$$

  where d_{ij} is the distance between mics i and j and d_{max} is the maximum distance between any mic pair.

3.9.3.1 Simulated HIFI

The choice of this database was motivated here by the fact that it uses array configurations composed by a large number of elements (as large as 33 for instance) where it is possible to find distant microphone pairs (as distant as 640 mm), close ones (as close as 20 mm) and all the intermediate values in between. With this he hope to get a fair, uniform weighting. Table 3.40 shows a comparison between the regular experiment with the

linear array of 33 sennheiser microphones equispaced 20 mm and the weighted one.

	33 mics (L=660 mm)	Weighted 33 mics (L=660 mm)
Pcor Rel. error reduction	$90.0 \pm 0.6\%$	$97.0 \pm 0.3\%$ 7.8%
Bias fine (x:y:z) [mm]	$1 : -56 : -20$	$1 : -56 : -11$
Bias fine+gross (x,y,z) [mm]	$-25 : 138 : -32$	$-1 : 6 : -15$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	103	111 −7.8%
Bias fine+gross [mm] Rel. BIAS f+g reduction	315	180 42.9%
A-MOTA Rel. error reduction	$81 \pm 0.8\%$	$93 \pm 0.5\%$ 14.8%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.40: Simulated HIFI. Microphone distance weighting effect. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 33 sennheiser microphones equispaced 20 mm (L = 660 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

Table 3.41 shows a comparison between the regular experiment with the harmonic linear array of 11 sennheiser microphones and the weighted one.

	11 harmonic mics	Weighted 11 harmonic mics
Pcor	$86.0 \pm 0.7\%$	$93.0 \pm 0.5\%$
Rel. error reduction		8.1%
Bias fine (x:y:z) [mm]	$1 : -95 : 8$	$1 : -94 : 0$
Bias fine+gross (x,y,z) [mm]	$-12 : 120 : -14$	$6 : 8 : 0$
Bias AEE fine [mm] = MOTP	182	121
Rel. AEE reduction		33.5%
Bias fine+gross [mm]	425	241
Rel. BIAS f+g reduction		43.3%
A-MOTA	$72 \pm 0.9\%$	$87 \pm 0.7\%$
Rel. error reduction		20.8%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.41: Simulated HIFI. Microphone distance weighting effect II. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear harmonic array of 11 sennheiser microphones. Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

Table 3.42 shows a comparison between the regular experiment with the linear array of 17 sennheiser microphones equispaced 20 mm and the weighted one.

	17 mics (L=340 mm)	Weighted 17 mics (L=340 mm)
Pcor	$27.0 \pm 0.9\%$	$55.0 \pm 1.0\%$
Rel. error reduction		103.7%
Bias fine (x:y:z) [mm]	$66 : 145 : -84$	$-39 : 70 : -29$
Bias fine+gross (x,y,z) [mm]	$165 : 706 : -88$	$114 : 329 : -46$
Bias AEE fine [mm] = MOTP	393	263
Rel. AEE reduction		33.1%
Bias fine+gross [mm]	1103	783
Rel. BIAS f+g reduction		29.0%
A-MOTA	$-47 \pm 1.0\%$	$9 \pm 0.6\%$
Rel. error reduction		-119.1%
Loc. frames	9375	9375
Ref. duration (s)	496.0	496.0

Table 3.42: Simulated HIFI. Microphone distance weighting effect IV. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 17 sennheiser microphones equispaced 20 mm (L = 340 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

Table 3.42 shows a comparison between the regular experiment with the linear array of 5 sennheiser microphones equispaced 160 mm and the weighted one.

	5 mics (L=800 mm)	Weighted 5 mics (L=800 mm)
Pcor	$75.0 \pm 0.9\%$	$75.0 \pm 0.9\%$
Rel. error reduction		0.0%
Bias fine (x:y:z) [mm]	$-2 : -35 : -31$	$-4 : -25 : -24$
Bias fine+gross (x,y,z) [mm]	$7 : -85 : -45$	$7 : -96 : -45$
Bias AEE fine [mm] = MOTP	173	164
Rel. AEE reduction		5.2%
Bias fine+gross [mm]	341	334
Rel. BIAS f+g reduction		2.1%
A-MOTA	$50 \pm 1.0\%$	$50 \pm 1.0\%$
Rel. error reduction		0.0%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.43: Simulated HIFI. Microphone distance weighting effect III. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Microphone array: Linear array of 5 sennheiser microphones equispaced 160 mm (L = 800 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

The experimental results confirmed the positive effect of this distance-dependent weigthing. However, this confidence on the localization results only works well with a high number of microphones so that there can be enough microphone pair combinations contributing.

The most positive aspect of this technique is its low computational demand. In practise, applying this weigthing does not result in any increase of the run time. The case comparing the experiments with 33 and 11 microphones respectively is specially spectacular. As we can see, the results using this a priori information on localization confidence with an 11 harmonic microphones array are better than even those thrown by a 33 microphones array with no weighting. What is more, the computational load saved by this operation is huge. Performing a localization search over a 50 mm grid with a 640 ms frame sampled at 48 KHz takes 1600 ms (40 real-time units) when using 33 microphones (528 possible microphone pair combinations). Meanwhile, performing the same operation with a distance-weigthed array of 11 harmonic elements (55 possible microphone pair combinations) will only take 190 ms (4.75 real-time units), a considerable reduction. This fact makes this technique crucial when aiming to reach real-time execution times with real life applications.

3.9.4 Filtering techniques

In this Section we will make a thorough study about how the different frequency bands in a speech signal contribute to the speaker localization. Theoretically, the high frequency components of the audio signals are more appropiate for localization purposes since they offer a much better response in terms of directivity pattern and thiner beamwidth and thus allow to a more accurate pointing in the search space. However, human speech presents a strong attenuation in power terms as the signal frequency increases. Therefore, it exists a clear trade-off between high frequency components, more accurate but with worse signal to noise ratios, and the low frequency bands concentrating most of the speech power but presenting unappropiate directivity patterns. The following experiments will attempt at giving us an insight about this issue. This information can be really valuable since it can set some kind of a priori knowledge that we could later use to improve our localization estimates based on the confidence on each frequency band. We could easily think of a weighting system, similar to the one applied with the intermicrophone distance in Section 3.9.3, that could give more importance to those frequency ranges which behave better when localizing.

During these experiments we will therefore apply low-pass, high-pass and band-pass filtering to different frequency ranges. We will also consider different databases, sampled at both 16 KHz and 48 KHz (and therefore having bandwidths of 8 and 24 KHz respectively), in order to get conclusions of general validity.

3.9.4.1 AV16.3

AV16.3 database, sampled at 16 KHz, allows us to study the range up to 8 KHz. Table 3.44 shows the low-pass filtering experiments performed with ST-AV16.3 in the frequency bands [0-1 KHz], [0-2 KHz], [0-4 KHz] and [0-8 KHz] (full-band).

	[0-8 KHz]	[0-4 KHz]	[0-2 KHz]	[0-1 KHz]
Pcor	$95.0 \pm 0.5\%$	$86.0 \pm 0.8\%$	$83.0 \pm 0.9\%$	$74.0 \pm 1.0\%$
Rel. error reduction		-9.5%	-12.6%	-22.1%
Bias fine (x:y:z) [mm]	$22 : 7 : 27$	$-31 : 1 : 121$	$-10 : -20 : 93$	$110 : 26 : 114$
Bias fine+gross (x,y,z) [mm]	$-27 : -60 : 17$	$-118 : -143 : 82$	$-84 : -169 : 59$	$-16 : -123 : 103$
Bias AEE fine [mm] = MOTP	100	193	214	263
Rel. AEE reduction		-93.0%	-114.0%	-163.0%
Bias fine+gross [mm]	191	377	387	493
Rel. BIAS f+g reduction		-97.4%	-102.6%	-158.1%
A-MOTA	$90 \pm 0.7\%$	$71 \pm 1.0\%$	$67 \pm 1.1\%$	$48 \pm 1.1\%$
Rel. error reduction		-21.1%	-25.6%	-46.7%
Loc. frames	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0

Table 3.44: ST-AV16.3. Low-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm

Table 3.45 shows the high-pass filtering experiments performed with ST-AV16.3 in the frequency bands [1-8 KHz], [2-8 KHz], [4-8 KHz] and [6-8 KHz].

	[1-8 KHz]	[2-8 KHz]	[4-8 KHz]	[6-8 KHz]
Pcor	$95.0 \pm 0.5\%$	$96.0 \pm 0.4\%$	$97.0 \pm 0.4\%$	$95.0 \pm 0.5\%$
Rel. error reduction		1.1%	2.1%	0.0%
Bias fine (x:y:z) [mm]	$22 : 7 : 26$	$23 : 7 : 27$	$30 : 12 : 14$	$22 : 7 : 27$
Bias fine+gross (x,y,z) [mm]	$-24 : -53 : 11$	$-17 : -30 : 14$	$14 : -9 : 5$	$-27 : -60 : 17$
Bias AEE fine [mm] = MOTP	100	101	106	100
Rel. AEE reduction		-1.0%	-6.0%	-0.0%
Bias fine+gross [mm]	187	176	156	191
Rel. BIAS f+g reduction		5.9%	16.6%	-2.1%
A-MOTA	$91 \pm 0.7\%$	$91 \pm 0.7\%$	$93 \pm 0.6\%$	$90 \pm 0.7\%$
Rel. error reduction		0.0%	2.2%	-1.1%
Loc. frames	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0

Table 3.45: ST-AV16.3. High-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm

Table 3.46 shows the band-pass filtering experiments performed with ST-AV16.3 in the frequency bands [1-2 KHz], [2-4 KHz] and [4-8 KHz] and compares them to the full-band experiment.

	[0-8 KHz]	[1-2 KHz]	[2-4 KHz]	[4-8 KHz]
Pcor	$95.0 \pm 0.5\%$	$80.0 \pm 0.9\%$	$84.0 \pm 0.8\%$	$97.0 \pm 0.4\%$
Rel. error reduction		-15.8%	-11.6%	2.1%
Bias fine (x:y:z) [mm]	$22 : 7 : 27$	$-38 : -23 : 84$	$-38 : 5 : 121$	$30 : 12 : 14$
Bias fine+gross (x,y,z) [mm]	$-27 : -60 : 17$	$-131 : -216 : 25$	$-117 : -86 : 73$	$14 : -9 : 5$
Bias AEE fine [mm] = MOTP	100	224	204	106
Rel. AEE reduction		-124.0%	-104.0%	-6.0%
Bias fine+gross [mm]	191	448	375	156
Rel. BIAS f+g reduction		-134.6%	-96.3%	18.3%
A-MOTA	$90 \pm 0.7\%$	$60 \pm 1.1\%$	$67 \pm 1.1\%$	$93 \pm 0.6\%$
Rel. error reduction		-33.3%	-25.6%	3.3%
Loc. frames	7295	7295	7295	7295
Ref. duration (s)	600.0	600.0	600.0	600.0

Table 3.46: ST-AV16.3. Band-pass filtering effect. fs = 16 KHz. Frame Size= 640 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 50 mm

It is interesting to note that we can achieve better localization results when high-pass filtering up to a certain point. Specifically, removing the frequencies lower than 1, 2 and even 4 KHz in AV16.3 database improves the performance of our system as expected, since these frequencies imply wide main lobes that do not allow a precise beamforming. As curiosity, it is remarkable that performing this high-pass operation practically removes every trace of human speech (whose power is mainly located in the first KHz band) but still yields better position estimates about where is the speaker talking. However, it is also important to note that, when removing all those frequencies lower than 6 KHz, the system starts to perform worse than all the previous high-pass filtering schemes, this is due to the fact that the speech energy in this band finally becomes as weak as to affect the performance. It is also remarkable that the frequency bands showing the worse behaviour are those ranging in the [0-1 KHz] and [1-2 KHz] limits: although containing most of the signal energy, their directivity patterns are not appropiate for localization purposes.

3.9.4.2 Real HIFI

HIFI database, sampled at 48 KHz, allows us to study the range up to 24 KHz. Table 3.47 shows the low-pass filtering experiments performed with Real HIFI in the frequency bands [0-1 KHz], [0-2 KHz], [0-4 KHz], [0-8 KHz], [0-16 KHz] and [0-24 KHz] (full-band).

Meanwhile, Table 3.47 shows the high-pass filtering experiments performed with Real HIFI in the frequency bands [1-24 KHz], [2-24 KHz], [4-24 KHz] and [8-24 KHz] and [16-24 KHz].

Finally, Table 3.47 shows the band-pass filtering experiments performed with Real HIFI in the frequency bands [1-2 KHz], [2-4 KHz], [4-8 KHz], [8-16 KHz] and [16-24 KHz].

We can somehow extract the same conclusions for the Real HIFI corpus: We can check that frequency bands performing the worst are either the low frequency ones, specifically the ranges [0-1 KHz] and [0-2 KHz], or those containing frequencies so high that no human speech power is present in them at all, particularly the range [16-24 KHz]. With this basis, we will expect that band-pass filtering between [2-16 KHz] to yield particularly good localization results, see Section 3.10 in page 153. However, we can appreciate that high-pass filtering schemes are not as effective in Real HIFI as they were in AV16.3. This is due to the bad properties of the capturing 4 elements array with respect to its directivity pattern response with frequency. High frequencies should perform better but, in this case, they imply having strong spatial aliasing and grating lobes that corrupt the localization estimates.

	[0-1 KHz]	[0-2 KHz]	[0-4 KHz]	[0-8 KHz]	[0-16 KHz]	[0-24 KHz]
Pcor	$27.0 \pm 0.9\%$	$55.0 \pm 1.0\%$	$70.0 \pm 0.9\%$	$82.0 \pm 0.8\%$	$85.0 \pm 0.7\%$	$84.0 \pm 0.7\%$
Rel. error reduction		103.7%	159.3%	203.7%	214.8%	211.1%
Bias fine (x:y:z) [mm]	$66 : 145 : -84$	$-39 : 70 : -29$	$-33 : -31 : -62$	$4 : -33 : -71$	$-3 : -22 : -52$	$-7 : -32 : -61$
Bias fine+gross (x,y,z) [mm]	$165 : 706 : -88$	$114 : 329 : -46$	$118 : 179 : -69$	$82 : 168 : -79$	$64 : 51 : -62$	$47 : 27 : -69$
Bias AEE fine [mm] = MOTP	393	263	209	199	195	206
Rel. AEE reduction		33.1%	46.8%	49.4%	50.4%	47.6%
Bias fine+gross [mm]	1103	783	583	501	415	426
Rel. BIAS f+g reduction		29.0%	47.1%	54.6%	62.4%	61.4%
A-MOTA	$-47 \pm 1.0\%$	$9 \pm 0.6\%$	$40 \pm 1.0\%$	$64 \pm 1.0\%$	$71 \pm 0.9\%$	$67 \pm 1.0\%$
Rel. error reduction		-119.1%	-185.1%	-236.2%	-251.1%	-242.6%
Loc. frames	9375	9375	9375	9375	9375	9375
Ref. duration (s)	496.0	496.0	496.0	496.0	496.0	496.0

Table 3.47: Real HIFI. Low-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm

	[1-24 KHz]	[2-24 KHz]	[4-24 KHz]	[8-24 KHz]	[16-24 KHz]
Pcor	$85.0 \pm 0.7\%$	$84.0 \pm 0.7\%$	$82.0 \pm 0.8\%$	$76.0 \pm 0.9\%$	$45.0 \pm 1.0\%$
Rel. error reduction		-1.2%	-3.5%	-10.6%	-47.1%
Bias fine (x:y:z) [mm]	$-7 : -32 : -61$	$-7 : -32 : -61$	$-6 : -30 : -61$	$-9 : -28 : -58$	$17 : -68 : -59$
Bias fine+gross (x,y,z) [mm]	$28 : -13 : -66$	$13 : 28 : -69$	$1 : 37 : -69$	$-9 : -25 : -64$	$8 : 487 : -101$
Bias AEE fine [mm] = MOTP	206	206	207	203	282
Rel. AEE reduction		-0.0%	-0.5%	1.5%	-36.9%
Bias fine+gross [mm]	424	427	449	518	1175
Rel. BIAS f+g reduction		-0.7%	-5.9%	-22.2%	-177.1%
A-MOTA	$69 \pm 0.9\%$	$68 \pm 0.9\%$	$64 \pm 1.0\%$	$53 \pm 1.0\%$	$-10 \pm 0.6\%$
Rel. error reduction		-1.4%	-7.2%	-23.2%	-114.5%
Loc. frames	9375	9375	9375	9375	9375
Ref. duration (s)	496.0	496.0	496.0	496.0	496.0

Table 3.48: Real HIFI. High-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm

	[1-2 KHz]	[2-4 KHz]	[4-8 KHz]	[8-16 KHz]	[16-24 KHz]
Pcor	$55.0 \pm 1.0\%$	$71.0 \pm 0.9\%$	$76.0 \pm 0.9\%$	$80.0 \pm 0.8\%$	$45.0 \pm 1.0\%$
Rel. error reduction		29.1%	38.2%	45.5%	−18.2%
Bias fine (x:y:z) [mm]	$-30 : 42 : -25$	$-30 : -37 : -57$	$8 : -25 : -72$	$-6 : -19 : -51$	$17 : -68 : -59$
Bias fine+gross (x,y,z) [mm]	$-233 : -186 : 3$	$115 : 39 : -63$	$21 : 273 : -84$	$-35 : -11 : -58$	$8 : 487 : -101$
Bias AEE fine [mm] = MOTP	238	197	204	193	282
Rel. AEE reduction		17.2%	14.3%	18.9%	−18.5%
Bias fine+gross [mm]	968	604	561	442	1175
Rel. BIAS f+g reduction		37.6%	42.0%	54.3%	−21.4%
A-MOTA	$9 \pm 0.6\%$	$42 \pm 1.0\%$	$51 \pm 1.0\%$	$61 \pm 1.0\%$	$-10 \pm 0.6\%$
Rel. error reduction		366.7%	466.7%	577.8%	−211.1%
Loc. frames	9375	9375	9375	9375	9375
Ref. duration (s)	496.0	496.0	496.0	496.0	496.0

Table 3.49: Real HIFI. Band-pass filtering effect. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame Size= 640 ms. Microphone array= Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm

3.9.4.3 SONY

SONY database, sampled at 48 KHz, allows us to study the simulated range up to 24 KHz. Table 3.47 shows the effects of low-pass filtering with a simulated database: SONY. The experiments were performed in the frequency bands [0-1 KHz], [0-2 KHz], [0-8 KHz], [0-16 KHz] and [0-24 KHz] (full-band).

We can again check that low-frequency bands included in [0-1 KHz], [0-2 KHz] are useless in terms of speaker localization. Only when taking bands including high frequencies and, thus, better directivity patterns responses, can we obtain proper results.

	[0-24 KHz]	[0-16 KHz]	[0-8 KHz]	[0-2 KHz]	[0-1 KHz]
Pcor	$96.0 \pm 0.1\%$	$92.0 \pm 0.1\%$	$36.0 \pm 0.2\%$	$17.0 \pm 0.2\%$	$1.0 \pm 0.0\%$
Rel. error reduction		-4.2%	-62.5%	-82.3%	-99.0%
Bias fine (x:y:z) [mm]	$-82 : -175 : -349$	$-82 : -174 : -348$	$-124 : -245 : -313$	$-306 : -52 : -327$	$-316 : 150 : -298$
Bias fine+gross (x,y,z) [mm]	$-101 : -149 : -348$	$-125 : -86 : -347$	$-278 : 496 : -331$	$-542 : 430 : -319$	$-329 : 882 : -325$
Bias AEE fine [mm] = MOTP	400	399	431	458	474
Rel. AEE reduction		0.2%	-7.8%	-14.5%	-18.5%
Bias fine+gross [mm]	432	481	1055	1069	1117
Rel. BIAS f+g reduction		-11.3%	-144.2%	-147.5%	-158.6%
A-MOTA	$88 \pm 0.1\%$	$81 \pm 0.2\%$	$-26 \pm 0.2\%$	$-63 \pm 0.2\%$	$-95 \pm 0.1\%$
Rel. error reduction		-8.0%	-129.5%	-171.6%	-208.0%
Loc. frames	213950	213950	213950	213950	213950
Ref. duration (s)	17690.0	8465.0	8465.0	17690.0	17690.0

Table 3.50: Simulated SONY. Low-pass filtering effect. fs = 48 KHz. Sequences considered: 8740 recordings from 20 different speakers placed at one single, static position (P2). Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Frame Size= 640 ms. Grid step in which the room was divided: 50 mm

3.9.5 Use of geometrical information

As depicted during the theoretical introduction in Section 2.4.4.6 in page 44, a possible improvement technique may consist on accurately defining the bounds of the search to fit the places where the speaker is likely to be as weall as taking into account dead areas in the room, such as big tables or wardrobes, where they speakers will rarely lie.

In this Section, we present some basic experiments to test the influence of this strategy on the final localization estimates.

3.9.5.1 AV16.3

AV16.3 geometry is ideal in order to test how much improvement these techniques may yield. On the first hand, we know the limits of the room in the Idiap Institute where the database was recorded: a rectangular room of 8.2mx3.6mx2.4m. However, we also know that, during the recordings, all the speakers where distributed along the L-shaped area depicted in Figure 3.3 in page 62. We can either perform a search over all the possible (x,y,z) coordinates contained in the room or just stick to those points with (x,y) coordinates belonging to the L-shaped area with z-coordinates ranging just in the habitual heights of a person either sitting or standing.

	Whole room search	L-shaped area search
Pcor	$73.0 \pm 1.0\%$	$81.0 \pm 0.9\%$
Rel. error reduction		11.0%
Bias fine (x:y:z) [mm]	$63 : 29 : 60$	$58 : 17 : 63$
Bias fine+gross (x,y,z) [mm]	$-116 : -318 : 113$	$-76 : -174 : 39$
Bias AEE fine [mm] = MOTP	218	197
Rel. AEE reduction		9.6%
Bias fine+gross [mm]	733	490
Rel. BIAS f+g reduction		33.2%
A-MOTA	$44 \pm 1.1\%$	$61 \pm 1.1\%$
Rel. error reduction		38.6%
Loc. frames	7295	7295
Ref. duration (s)	600.0	600.0

Table 3.51: ST-AV16.3. Use of geometrical information effect. fs = 16 KHz. Frame Size= 40 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm

As we can appreciate in Table 3.51, determining an accurate region where to search,

using any kind of previous information about the most likely places where the speaker can be, demonstrates to be crucial in order to improve the localization rates.

We will next analyze the results when ignoring those localization estimates placed at any of the dead areas defined. The Idiap room contains a big, centered rectangular, 4.8mx1.2m table. We can assume speakers talking around this table or even lying on it, but we will discard any position estimate located more than 200 mm into the table.

	Whole room search	Discarding dead areas
Pcor	$73.0 \pm 1.0\%$	$73.0 \pm 1.0\%$
Rel. error reduction		0.0%
Bias fine (x:y:z) [mm]	$63 : 29 : 60$	$63 : 29 : 60$
Bias fine+gross (x,y,z) [mm]	$-116 : -318 : 113$	$-116 : -318 : 113$
Bias AEE fine [mm] = MOTP	218	218
Rel. AEE reduction		-0.0%
Bias fine+gross [mm]	733	733
Rel. BIAS f+g reduction		-0.0%
A-MOTA	$44 \pm 1.1\%$	$44 \pm 1.1\%$
Rel. error reduction		0.0%
Loc. frames	7295	7295
Ref. duration (s)	600.0	600.0

Table 3.52: ST-AV16.3. Dead areas effect. fs = 16 KHz. Frame Size= 40 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 150 mm

As we can see there is no performance difference in this case between both cases. This is due to the fact that no estimation was determined to be inside the table. However, this technique may prove to be useful in rooms with large dead areas and under tougher noise or reverberation conditions that may lead the system to output wrong estimates belonging to these areas.

3.10 Selected final experiments

Finally, we have selected three different experiments that sum up the best strategies we can follow according to all the information collected through the previous experiments. These three experiments aim at giving a picture as complete as possible: They have been performed with three different databases, including real and simulated ones and cover three different frame sizes and estrategies.

It is important to note that these experiments do not only concentrate on localization performance but also take into account computational costs. Better localization rates than the ones shown in the following Tables 3.53, 3.54 and 3.55 were in fact achieved, but at the cost of necessarily having to spend a considerably higher computational time, far above from the real-time constraints.

As specified in Section 3.2.1 in page 52, z-coordinate errors demonstrated to be less critical since the arrays used were lined along the XY plane therefore having a symmetrical directivity patern along the z-axis. Thus, it turned out to be important to evaluate not only the 3D, (x,y,z), results, depicted in the first column of the following Tables, but also the 2D, (x,y), results, depicted in the second column of the following Tables, in order to properly characterize the algorithm behaviour.

3.10.1 AV16.3

The following Table 3.53 shows the most appropiate results for the AV16.3 database. The static sequences were selected. The frame size, 500 ms, was chosen to be the one showing the best performance vs. computational time ratio. Actually choosing a greater frame size would result in better localization estimates, however, as set in Figure 3.9 in page 82, increasing the size above this 500 ms would lead to neccesarily having to double the FFT size therefore doubling the computational time spent, see Section 3.8 in page 123. Also, the grid spacing selected, 100 mm, is not the ideal one as demonstrated in Section 3.5.5 in page 93: A finer grid would yield better estimates but would at the same time imply a slower application, see Section 3.8 in page 123. We also have made use of the a priori knowledge about the best frequency band in localization terms, [4-8 KHz], see Section 3.9.4 in page 143. Finally, the Hamming window and the rounding techniques were chosen as they demonstrated to be the most reasonable ones throughout the experiments.

	(x:y:z) error [mm]	(x:y) error [mm]
Pcor	$94.0 \pm 0.5\%$	$96.0 \pm 0.4\%$
Rel. error reduction		2.1%
Bias fine (x:y:z) [mm]	52 : 21 : 31	53 : 23 : 0
Bias fine+gross (x,y,z) [mm]	37 : 2 : 26	37 : 2 : 0
Bias AEE fine [mm] = MOTP	159	134
Rel. AEE reduction		15.7%
Bias fine+gross [mm]	224	191
Rel. BIAS f+g reduction		14.7%
A-MOTA	$90 \pm 0.7\%$	$93 \pm 0.6\%$
Rel. error reduction		3.3%
Loc. frames	7295	7295
Ref. duration (s)	600.0	600.0

Table 3.53: ST-AV16.3. Selected final experiment. fs = 16 KHz. Frame size= 500 ms. Microphone array= Two circular arrays of 8 microphones each. Grid step in which the room was divided: 100 mm. Windowing: Hamming. Band-pass filter: [4-8KHz]. Rounding applied.

The CPU load for this particular experiment was 65 ms per estimation. Given that estimations are output every 40 ms, this means 1.62 real-time units, really close to real-time performance.

3.10.2 Real HIFI

The following Table 3.54 shows the most appropiate results for the Real HIFI database. The sampling frequency selected was 48 KHz instead of 16 KHz since it leads to smaller rounding errors, see Section 3.5.1 in page 3.5.1. The frame size, 320 ms, was chosen because of outputting the best localization results, see Section 3.5.2 in page 80. Grid spacing, 150 mm, was also the one throwing the best estimates in the case of the Real HIFI database captured by the 4 elements linear array, see Section 3.5.5 in page 3.5.5. Moreover, the information about the best frequency bands was used: In Section 3.9.4 in page 143, we concluded that, in Real HIFI, bands lower than 1 KHz and higher than 16 KHz didn't work properly. Therefore, this will be the interval we will use here. Finally, the Hamming window and the rounding scheme were again the most reasonable techniques.

	(x:y:z) error [mm]	(x:y) error [mm]
Pcor Rel. error reduction	$88.0 \pm 0.7\%$	$88.0 \pm 0.7\%$ 0.0%
Bias fine (x:y:z) [mm]	$-1 : -21 : -50$	$-2 : -21 : 0$
Bias fine+gross (x,y,z) [mm]	$3 : 21 : -56$	$3 : 21 : 0$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	195	108 44.6%
Bias fine+gross [mm] Rel. BIAS f+g reduction	375	296 21.1%
A-MOTA Rel. error reduction	$77 \pm 0.9\%$	$77 \pm 0.9\%$ 0.0%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.54: Real HIFI. Selected final experiment. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame size= 320 ms. Microphone array: Linear array of 4 sennheiser microphones equispaced 200 mm (L = 800 mm). Grid step in which the room was divided: 150 mm. Windowing: Hamming. Band-pass filter: [2-16KHz]. Rounding applied.

The CPU load for this particular experiment was 11 ms per estimation. Given that estimations are output every 40 ms, this means 0.27 real-time units, a quite fair result.

3.10.3 Simulated HIFI

The following Table 3.55 shows the most appropiate results for the Simulated HIFI database. In this case, the higher possible sampling frequency, 48 KHz, a long frame size, 640 ms and an accurate grid spacing, 50 mm, were all selected in order to achieve the best possible localization results although this could imply longer computational times. A proper microphone array, having 11 elements distributed harmonically, see Section 3.6.1 in page 3.6.1, was also used. Apart from that, the microphone distance weigthing technique was applied as it demonstrated to largely improve the localization results in this case without having to increase the computational load, see Section 3.9.3 in page 137. Once again, the Hamming window and the rounding technique were selected.

	(x;y;z) error [mm]	(x;y) error [mm]
Pcor Rel. error reduction	$94.0 \pm 0.5\%$	$93.0 \pm 0.5\%$ -1.1%
Bias fine (x:y:z) [mm]	$1 : -93 : 0$	$1 : -94 : 0$
Bias fine+gross (x,y,z) [mm]	$6 : 8 : -9$	$6 : 8 : 0$
Bias AEE fine [mm] = MOTP Rel. AEE reduction	95	121 -27.4%
Bias fine+gross [mm] Rel. BIAS f+g reduction	193	241 -24.9%
A-MOTA Rel. error reduction	$87 \pm 0.7\%$	$87 \pm 0.7\%$ 0.0%
Loc. frames	9390	9390
Ref. duration (s)	367.0	367.0

Table 3.55: Simulated HIFI. Selected final experiment. fs = 48 KHz. Sequences considered: 223 recordings from 12 different speakers placed at 5 different, static positions. Frame size= 640 ms. Microphone array: Linear harmonic array of 11 sennheiser microphones. Grid step in which the room was divided: 50 mm. Windowing: Hamming. Microphone distance weigthing applied. Rounding applied.

The CPU load for this particular experiment was 150 ms per estimation. Given that estimations are output every 40 ms, this means 3.75 real-time units, much slower than a real-time application but still reasonable given the time-consuming conditions imposed and the localization results obtained. We could save computational time at the cost of slighty reducing the localization rate.

Chapter 4

Conclusions

This chapter makes a general revision of the content of this Master Thesis report. Special stress will be put on the fundamental conclusions reached through the development of this Master Thesis as well as the main contributions it contains.

Chapter 1 presented the main objectives aimed to achieve with this Master Thesis as well as a justification of their achievement and an introduction to the report structure.

Chapter 2 details the theoretical background over which this Master Thesis has been constructed. Special attention has been given to the speaker localization techniques: Starting with the basic technique to determine the Direction of Arrival (DOA) based on the Generalized Cross Correlation between microphone pairs with a Phase Transform prewhitening filter (GCC-PHAT) and ending up with a more robust algorithm based on the Steered Response Power (SRP-PHAT) of a microphone array when beamformed to point at different space locations. Insights about some possible improvement techniques that could be applied to this algorithms are also discussed as well as some key aspects about tracking algorithms, to be included in future research works, and Voice Activity Detectors (VAD) procedure, fundamental in order to determine whether there is an active speaker at a certain time period and therefore if a localization estimate during that period makes sense or not.

Chapter 3 describes the exhaustive experimentation carried out to test the two localization algorithms depicted above: GCC-PHAT and, mainly, the final implemented version of SRP-PHAT. These experiments were thought to cover as many cases as possible: they make use of both real databases such as AV16.3 and HIFI and simulated ones such as Sony and Simulated HIFI, and aim at evaluating the influence of the different *tunable parameters* on the system performance, as well as testing the different possible improvement techniques and assuring the statistical relevance of the results obtained. Next, we will highlight the main conclusions we can extract from the experiments conducted

in the different Sections of this Chapter 3:

- Early experiments demonstrated that localization estimates, both those given by GCC and SRP, improve when these algorithms are applied a prefilter rather than using their unfiltered versions. Out of all the possible prefilters tested, the Phase Transform (PHAT) pre-whitening filter showed the best performance rather than the Roth Processor, the Smoothed Coherence Transform (SCOT) or the Maximum Likelihood (ML) filter. More details about these prefilters can be found in Section 2.4.1.2 in page 27 or in IEEE paper [KC76].

- The more distant two microphones in a pair are, the more reliable their GCC-PHAT DOA estimations will be, that is to say, the smaller the relative error they will commit as demonstrated theoretically in equation 2.43 in page 31.

- SRP-PHAT estimations demonstrate to be much more accurate and robust when compared to those of the GCC-PHAT method as expected, since SRP can be defined as the sumation of all possible GCCs between all possible microphone pair combinations as demonstrated in paper [Dib00] pp. 78-80.

- Real databases offer better and more consistent localization results than simulated ones often missing more accurate models able to properly imitate the real recording conditions.

- *Tunable parameters effects*

  Sampling frequency, having a higher sampling frequency yields an error reduction in the localization estimates. As trade-off, it will also imply a higher computational load since it will be neccessary to take into account a higher number of samples.

  Frame size, in general, the longer the frame size the better localization estimates. Again, the trade-off is the CPU load increase. However this general hint is not valid in two cases: First, when localizing speakers in movement the results do increase as we increase the frame size but just to a certain length beyond which the estimates yielded will be worse since the speaker position starts to significally differ from the starting to the end point of the analyzed frame. Secondly, when analyzing too short utterances if frames are too long they will neccessary contain significal portions of silence or zero-padding that will make the estimation errors increase.

  FFT size, it determines the maximum distance our system can reach in search of a speaker as demonstrated in Section 3.5.3 in page 88. Apart from this, it does not affect the localization estimates accuracy. On the contrary, it does affect the computational load, doubling the FFT size implies doubling the CPU time spent.

It is important to note that the size of the FFT transform must always be equal or greater than that of the frame considered and, therefore, this is how the frame size has an influence on the execution time.

Window type, early experiments showed the convenience of applying windowing functions to the frames prior to their analysis in order to avoid wrong estimations. Among the different windowing functions Hamming demonstrated to perform better than Hanning, Blackman and Barlett.

Search space grid, in general, dividing the search space with more and more finesse yields more and more accurate localization estimate. Once more, this measure has a trade-off in form of heavier computational cost: Having more and more points in the search space where to steer at increases the execution time proportionally. However, this accuracy improvement is just valid when working with appropiate microphone arrays. Arrays whose elements are not close enough to each other, thus leading to spatial aliasing in high frequency bands, or whose effective length is not long enough, thus having too wide main lobes as to be steered accurately, will not fulfil the previous assertion. Moreover, localization estimates will just be proper if the speaker locations are properly taken into account to "fit well" into the designed grid.

- *Microphone array effects*

  Number of microphones and intermicrophone distance, the closer the elements are to each other in the array in order to avoid spatial aliasing and the longer its effective length is in order to present a thin, accurate main lobe, the better the localization estimates will be. Therefore the ideal array is that having small intermic distances but lots of elements so that its effective length is long enough. Moreover a significal number of elements in the array reduces the side lobes level and grants more accurate Steered Response Power (SRP) patterns. The trade-off of this type of arrays is, once again, the high computational cost associated to them, since it will be neccessary to take into account a microphone pair combation number that grows in a quadratic mood every time a new element is added to the set.

  L-shaped array, although its shape assures theoretically smaller geometric areas in which the speaker can stand, their localization estimates are seriously worsened by the strong reverberation conditions present at their corner location.

- Those speaker positions perpendicular to the microphone array are the ones showing the best localization results. Whenever they are not perpendicular, then, the more tilted they are with respect to the array the better for our localization purposes, which confirms what equation 2.43 in page 31 hinted.

- *Additional strategies effects*

  Coarse to fine search, this strategy requires to experimentally set and tune the proper relationship between the width of the energy peak and the source frequency for every specific array configuration.

  Noise masking, adopting a fixed threshold estrategy for all the frequencies proved to mislead the localization estimates as it leads to massive discard at high frequency bands, those containing less energy but being more effective in localization tasks. An adaptive threshold strategy setting a discard mask related to the energy spectra at each frequency component demonstrated to effectively work in noisy environments although it requires a prior, proper tuning of the threshold level to choose.

  Estimation of localization confidence, localization estimates result to be more accurate if we put more confidence on those microphone pairs whose elements are more distant. However, this confidence only improves the results when there is a large enough number of microphone pairs contributing.

  Filtering techniques, the use of pass-band frequency bands, when properly chosen, demonstrates to improve the localization estimates. Specifically, low frequency bands, typically under 2 KHz and associated to directivity patterns with too wide main lobes, are not advised to be taken into account as well as too high frequency bands, typically above 16 KHz, containing low speech power and prone to show spatial aliasing.

  Interpolation techniques, the experiments done within this field showed us the importance of reaching sub-sample resolution in our digital computations. As explained in Section 2.4.1.3 in page 29 the conversion from real time delay units in seconds to discrete, digital time delay units in samples neccesarily leads to imprecisions in our computations that must be avoided as much as possible. These imprecisions depend on 4 main facts: First, the sampling frequency of the signal. Second, an accurate selection of the speed of sound in the given environment. Third, the speaker position defining its DOA with respect to the array. Forth, the rounding function used to transform real time units into digital ones. 3 main solutions were proposed to this imprecision problem:

  FSRP, A first solution to the problem was to directly avoid it by straightahead using the exact real time delay units in seconds. This is possible in the Frequency SRP (FSRP) method described in Section 2.4.3.3 in page 37. The experimentation carried out with this method shows an important improvement in the localization rates but at the cost of prohibitively rising the execution time required.

  Interpolation, The second solution we thought of was to artificially rise the sam-

pling frequency of the digital signal by x2 and x3 interpolation processes in order to reduce imprecissions. Once again, the localization results improve although they are never as good as if they would have been obtained with signals originally sampled at rates x2 and x3 times higher. However, once again, this technique implies rising x2 and x3 the execution time required.

Rounding functions, Finally we thought of trying to smartly sort out the values of the missing subsamples in the correlation function based on their neighbouring samples values. This technique means no rise in the computational time spent and proves to effectively work throwing localization rates comparable, although slightly lower, than the previous methods.

Use of geometrical information, the use of any kind of a priori knowledge about the position speaker turns out to be of high importance: defining proper bounds where to search the speaker (as tight as possible but without discarding any possible location) as well as taking into account those dead areas where the speaker is not likely to be, such as armchairs, large tables, etc. help the system to discard wrong location outputs.

- Distance errors between the true speaker position and their corresponding estimates are generally much higher along the z-coordinate compared to those of the x and y-coordinates. This is due to the fact that the arrays used are XY lined and therefore have a symmetric directivity pattern along the z-axis which makes it difficult to make a difference between points lined in this direction. We can make use of this fact in two possible ways:

  Computational time, we can reduce the finesse of the search space grid in the z-direction without loss of accuracy in the system estimates but with a gain in the execution time since there will be less locations to evaluate.

  Localization rate, we can rise the localization rate by giving more finesse in the x and y directions and less in the z one. The computational time can be keep constant since the overall number of points to evaluate can be approximately the same as when giving equal finesse to all directions.

- Computational load in SRP algorithm increases linearly with the FFT size (and, consequently, with the frame size), the number of search points (which depends upon the grid spacing defined) and quadratically with the number of microphones in the array. FSRP implementation is also far more computationally expensive than TSRP. In general, we appreciate a trade-off between localization performance in the system and CPU time spent.

Chapter 5

Future work

5.1 Introduction

This chapter will give an overview about the future research lines opened after the completion of this Master Thesis. The main tasks in this Master Thesis were the design, implementation and evaluation of a speaker localization systems. The design process was completed succesfully and the evaluation of the implemented algorithms was exhaustive in terms of databases, tunable parameters, improvement techniques and environment conditions and geometries giving little room to further extend it in the future with new cases. Hence, most of the future lines proposed focus on giving hints about the design and implementation of new algorithms and improvement techniques that could be added to the system in order to improve its localization tasks.

5.2 Implementation of the Time Delay Selection (TIDES) algorithm

Described in Varma's Master Thesis [Var02] pp. 81-122, the Time Delay Selection (TIDES) algorithm proposes to take into account not only that peak value of the GCC (Generalized Cross Correlation) function but also some other weaker peaks. This technique is based on the fact that GCC function was observed to always contain a peak at the proper time delay difference between a microphone pair. However, this peak does not always turn out to be the strongest one since reverberation may make other peaks appear at wrong time delays.

This idea was taken into consideration during the achievement of this Master Thesis and the system was implemented to have a tunable parameter that could set how many

peaks we want to select out of the GCC functions. However, this issue was never exploited and opens a door to add a new improvement algorithm in the future that could lead to improved estimations.

We here suggest a basic outline for this algorithm that could consider selecting just the two strongest peaks in the GCC function, instead of just the strongest one as done in this Thesis. When compared, if one of these two peaks turns out to be dramatically stronger we will proceed as usual. However, if the two peaks result to be similar in strength we will make use instead of a weighted version of the values corresponding to the two peaks.

5.3 Improvement of the coarse to fine search method

As described in Section 3.9.1 in page 129, the experimental results obtained with the implemented coarse to fine search method did not fit the expectations, both in computational requirements and localization rates, we had deposited on it.

As far as the localization rates are concerned, we may think of adding a new, simplified version of our coarse to fine algorithm as the one described in Abad's Thesis [AG07]. Based on a two-steps technique instead of on the complex multi-step version based on octrees referenced in Duraiswami's article [ZD04]. This new technique would consider a gross search in the first step performed with a proper cut-off frequency associated to a energy peak width that could cover areas gross enough as to explore the environment just by looking at a few spots. After this step, that gross area, gross but yet significally smaller than the whole room, showing the greater Steered Response Power (SRP) would be selected to be explored with finesse and with the full-band version of the signal this time. This scheme is promising in both the computational requirements, since the number of points to explore can be much smaller than directly dividing the whole room with finesse, and the localization rates, since the second step search could be done with much more finesse than allowed when working with the whole room area.

Finally, as far as the computational requirements are concerned, the implemented coarse to fine algorithm could also be modified so that the IFFT transforms could have less points and, thus, run more efficiently: Whenever we are making use of a low-pass filtered version of the original signal in order to get grosser energy peaks to explore gross areas of space we do not need to apply IFFT transforms covering the full signal band. We could then reduce the IFFT size to just fit the band limitted by the corresponding cut-off frequency. This algorithm optimization would lead to some computational load save.

5.4 Further estimation of the localization confidence

As we described during the theoretical introduction in Section 2.4.4.3 in page 41, any kind of a priori information about how the reliability of our estimations is related to the prior experiment conditions can be used to improve our localization results by designing techniques that put more emphasis on those conditions which yield the best results.

The microphone distance weighting technique is a good example of this: Since the experiments in Section 3.4 in page 71 showed distant microphone pairs to output better estimates, we decided to give them more importance by simply applying a weighting function.

Therefore, this same thing could also be extended in the future to a set of different other features. For instance, as pointed out in Section 3.9.4 in page 143 when talking about the filtering schemes, the results output by the experiments in this section could be used to likewise design a weighting function that would integrate the information coming from all frequency bands but giving more importance to those showing better performances. This approximation would surely output better estimates than the ones obtained by simply using the pass-band filtered versions of the signal that demonstrated to perform better as it was done in the final experiments Section 3.10 in page 153.

However, the design of simple, linear weighting functions as proposed above does not seem to exploit all the potential of this localization confidence path. We can think of a much powerful tool to take into account all these facts. As future line of research we propose a new strategy with no precedent in the state-of-the-art literature: The design and implementation of a MuMe, [Jab94], Neural Network (NN) that can be trained in order to sort out itself the appropiate weighting function corresponding to the overall set of parameters selected to have some influence on the localization confidence. In order to do so, it is important first to make an in-depth study about all these possible issues that may have an effect on the localization confidence. We have already here clearly identified two of them: the inter-microphone distance and the signal frequency bands. There can be though many others such as the environment geometry, the signal to noise ratio, the speech spectral content, etc. All of them will be used as input during the training phase of the NN which will compare the results that our localization algorithm yields to the ground truth positions taking into account every possible variation of selected input parameters. During this training phase the NN is able to design a weighting function that perfectly shapes the contribution of all the input conditions so that the output they yield is as similar to the ground truth as possible. Once this weighting function has been found out we can later use it as a valuable a priori information that will help our estimates resemble more the true speaker localizations.

5.5 Use of better array configurations

As concluded in the array geometry Section 3.6 in page 108, there are some array configurations which are not suited for speaker localization tasks. For instance, the linear array of four 200 mm equispaced sennheiser microphones used during the HIFI database recordings offers a quite poor directionality at low frequencies, not being able thus to reject signals coming from undesired directions, and strong spatial aliasing at medium and high frequencies as can be seen in Figures 3.13, 3.14, 3.15, 3.16 and 3.17 in page 100. In addition, the L-shaped array of 3 crown microphones also offers a poor directivity and it is subject to strong reverberation and inter-microphone replicas due to its corner position.

Therefore, for the recording of future databases aimed at speaker localization purposes, we here suggest the use of linear arrays characterized by a reasonbly small microphone inter-distance (i.e. 20 mm) in order to avoid aliasing at high frequencies as well as a reasonably large (i.e. > 500 mm) effective length in order to get thin main lobe patterns. The longer the number of elements in the array, the lower the side lobes thus helping to reject signal from undesired directions, but also, the higher the computational time spent to take into account the quadratically growing number of mic pair combinations. Hence, we propose not to neccessarily using a equispaced microphone scheme to fulfil the effective lenght but rather a harmonic distribution of them in the mood of the 11 harmonic array of the Simulated HIFI database depicted in Figure 3.5 in page 68. Moreover, this harmonic array configuration could also be used to implement a Constant Directive Beamformer (CDB) as described in [BW01] pp.3-19, [AG07] pp. 35-38 and Section 2.3.2.3 in page 22. It basically consists on capturing each frequency band with a certain linear sub-array within the harmonic array whose equispaced elements have an inter-distance designed to fit with the frequency band that it aims to capture. The result is an approximately constant directivity pattern along all the frequency ranges as depicted in the instance in Figure 5.1.

In addition to the previous suggestions, and given the fact that XY lined linear arrays cannot make clear distinctions along different locations lying in the z-direction, it would be interesting to form not a linear, 2D array but a T-shaped, 3D array. Some suggestions in this sense can be found at [MSBS95] and [MSBS97]. In its simplest configuration it could be formed by a usual, XY lined, linear array plus the addition of at least one extra microphone centered in a different z-coordinate with respect to the linear array. This way, all the mic pair contributions including this extra microphone would give us precious distinct information about the z-axis localization that would help us reduce the typically large errors committed in this Master Thesis in the z coordinate estimation.

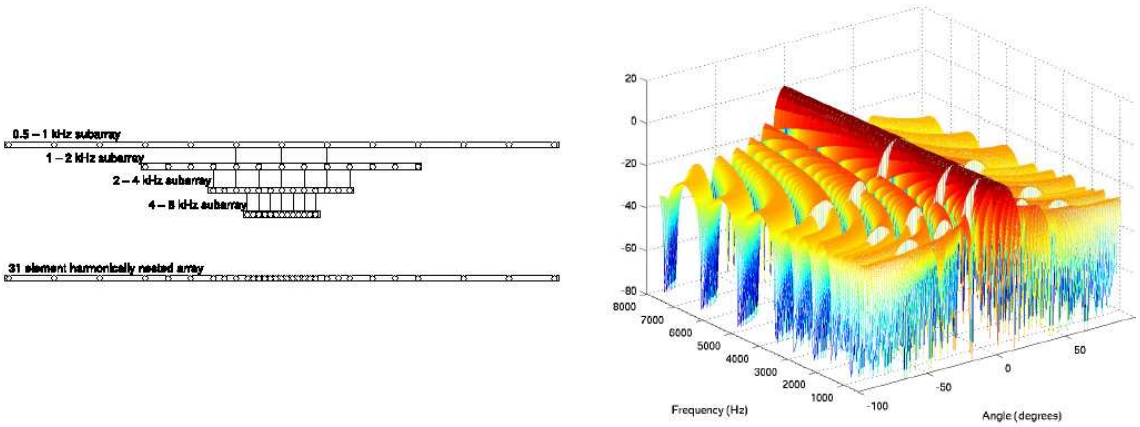

Figure 5.1: Instance of a Constant Directivity Beamformer array (left) and its correponding directivity pattern along all the frequency bands (right). An almost frequency-constant directivity pattern is obtained.

5.6 Algorithm optimization

The final objective of the localization system, once its implementation and testing have been finished, is to perform in real time. In order to do so, an optimization process of the algorithms implied it is crucial to reduce the execution time of each position estimate.

During this optimization process, it is important to find out which parts of the program are the most demanding in computational time and concentrate on their implementation and instructions to try to lower their load. In this sense, the *gprof* tool, [FS], can help us analyze our code giving us precise information about the heavier parts in our code. This application is an useful way to profile programs that are too big or complex as to be directly analyzed from their source code.

gprof harvest information about our program while analyzing it during a real execution of the system. It shows how much time the program spent in each function as well as how many times every function was called and who was the one calling them every time. This turns out to be a valuable knowledge about how to rewrite our code, we should concentrate on optimizing the heavier rutines as well as minimizing the excesive call of computationally demanding functions.

5.7 Tracking algorithms implementation

In order to properly localize moving acoustic targets it is appropiate to develop algorithms able to automatically track the speaker. These tracking techniques can improve

the robustness of our estimates since they integrate several temporally differenciated estimates: In fact, they basically consist on filtering the instantaneous location estimates provided by the localization system. The *Kalman filter* [Kal60] has been proposed in [DESS97] as a way to spatially smooth these estimations. Then, geometrical properties are used in order to infer the source position. As an alternative, Sequencial Monte-Carlo (SMC) methods, also known as *Particle Filtering (PF)*, implement a Bayesian filter that tries to predict the optimal candidate configurations by measuring their likelihood, given the localization estimates [DBWW03].

These techniques are here proposed for further implementation in future research works. The work developed in [CSH06] can be taken as reference for a simple tracking implementation based on a SRP-PHAT method that bases its estimates on the summation of the CPS (Cross-Power Spectrum) functions of the actual and previous frames, weighted by an adaptive smoothing factor given by the Kalman estimator.

5.8 Multiple speaker localization

This current Master Thesis has concentrated on the localization of a single speaker. Further improvements of this work could include the addition of multiple speaker localization techniques. An exhaustive introduction to these techniques can be found in [AM01] pp. 180-203. In this sense, the Steered Response Power (SRP) scheme is appropiate for multi-source detection purposes as ti performs a comprehensive inspection of the search space allowing to encounter different power maxima.

Apart from this possibility, there is a new open field of research to locate multiple speakers through the design and implementation of the so-called High Resolution Sub-Space methods such as the MUSIC (Multiple Signal Classification) algorithm, which makes use of eigenanalysis-based techniques to break up the signal into spatio-spectrally differentiated sub-spaces, see [WK85] and [Sch86].

Bibliography

[AG07] Alberto Abad Gareta. *A Multi-Microphone Approach to Speech Processing in a Smart Room Environment*. phd thesis, Technical University of Catalonia, 2007.

[AM01] Iain A. McCowan. *Robust Speech Recognition Using Microphone Arrays*. phd thesis, Queensland University of Technology, Australia, 2001.

[BS97] Michael S. Brandstein and Harvey F. Silverman. A robust method for speech signal time-delay estimation in reverberant rooms. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1:375–378, April 1997.

[BW01] M. Brandstein and D. Ward. *Microphone Arrays*. Digital Signal Processing. Springer, 2001.

[Com90] D. Van Compernolle. Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2:833–836, April 1990.

[CSH06] C. Nadeu C. Segura, A. Abad and J. Hernando. Multimodal person tracking in a smart-room environment. *IV Conference on Speech Technology, Zaragoza*, pages 271–276, November 2006.

[DBWW03] E. A. Lehmann D. B. Ward and R. C. Williamson. Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Transactions on Speech and Audio Processing*, 11:826–836, November 2003.

[DESS97] Michael S. Brandstein Douglas E. Sturim and Harvey F. Silverman. Tracking multiple talkers using microphone-array measurements. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1:371–374, April 1997.

[DG05] T.V. Dvorkind and S. Gannot. Speaker localization using the unscented kalman filter. *Proceedings on HSCMA Workshop*, March 2005.

[Dib00] Joseph Hector Dibiase. *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. phd thesis, Brown University, 2000.

[FJ06] Matteo Frigo and Steven G. Johnson. *FFTW for Version 3.1.2*. Massachusetts Institute of Technology, June 2006.

[FS] Jay Fenlason and Richard Stallman. *The GNU Profiler*. World Wide Web, URL: `http://www.cs.utah.edu/dept/old/texinfo/as/gprof_toc.html`.

[Gar07] Owen Gareth. *Speed of sound - Wikipedia*. World Wide Web, `http://en.wikipedia.org/wiki/Speed\_of\_sound`, 2007.

[GLGP04] Jean-Marc Odobez Guillaume Lathoud and Daniel Gatica-Perez. *AV16.3: an Audio-Visual Corpur for Speaker Localization and Tracking*. IDIAP Research Institute, Switzerland, June 2004.

[Her05] Wolfgang Herbordt. Sound capture for human/machine interfaces - practical aspects of microphone array signal processing. *Springer, Heidelberg, Germany*, 2005.

[HFSF96] W. R. Patterson H. F. Silverman and J. L. Flanagan. *The Huge Microphone Array (HMA)*. Brown University, Providence, RI, 1996.

[Jab94] Marwan Jabri. *MuME an Environment for Multi-Net and Multi-Algorithm Neural Simulation*. Sydney University Electrical Engineering, January 1994.

[JU97] S.J. Julier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Proceedings on Aerospace/Defense Sensing, Simulation and Controls (AeroSense)*, 1997.

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82 (Series D):35–45, March 1960.

[KC76] Charles H. Knapp and G. Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-24(4):320–327, August 1976.

[LAS03] Maurizio Omologo Luca Armani, Marco Matassoni and Piergiorgio Sveizer. Use of a csp-based voice activity detector for distant-talking asr. *Proccedings of the European Conference on Speech Communication and Technology (Eurospeech)*, pages 501–504, 2003.

[Lat06a] Guillaume Lathoud. *AV16.3: an Audio-Visual Corpus for Speaker Localization and Tracking*. IDIAP Research Institute, Switzerland, `http://mmm.idiap.ch/Lathoud/av16.3_v6`, 2006.

[Lat06b] Guillaume Lathoud. *Spatio-Temporal Analysis of Spontaneous Speech with Microphone Arrays*. phd thesis, Ecole Polytechnique Federale de Lausanne, 2006.

[Leh04] E. Lehmann. *Particle Filtering Methods for Acoustic Source Localization and Tracking*. phd thesis, Australian National University, 2004.

[Lew07] Mark Lewis. *Margin of error - Wikipedia*. World Wide Web, `http://en.wikipedia.org/wiki/Margin\_of\_error`, 2007.

[MG02] Nelson Morgan and David Gelbart. Double the trouble: Handling noise and reverberation in far-field automatic speech recognition. *IEEE International Conference on Spoken Language Processing (ICSLP)*, 2002.

[MH06] Eva Munoz Herraiz. *Design, Implementation and Evaluation of Source Localization and Speech Signal Improvement Techniques in Acoustic Reverberant Environments: Application to Automatic Speech Recognition Systems*. phd thesis, Technical University of Madrid, 2006.

[Moo02] Darren C. Moore. *The IDIAP Smart Meeting Room*. IDIAP Research Institute, Switzerland, November 2002.

[MS94] P.C. Meuse and H.F. Silverman. Characterization of talker radiation pattern using a microphone array. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2:257–260, April 1994.

[MSBS95] John E. Adcock Michael S. Brandstein and Harvey F. Silverman. A closed-form method for finding source locations from microphone-array time-delay estimates. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3019–3022, 1995.

[MSBS97] John E. Adcock Michael S. Brandstein and Harvey F. Silverman. A closed-form location estimator for use with room environment microphone arrays. *IEEE Transactions on Speech and Audio Processing*, 5:45–50, January 1997.

[OM06] Maurizio Omologo and Djamel Mostefa. *Clear Evaluation Plan*. CHIL, February 2006.

[OS89] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, second edition, 1989.

[OS94] Maurizio Omologo and Piergiorgino Svaizer. Acoustic event localization using a crosspower spectrum phase based technique. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2:273–276, April 1994.

[OV03] Javier Ordonez Vazquez. *Design, Implementation and Evaluation of a Pre-Process Modules Library for Automatic Speech Recognition Systems*. phd thesis, Technical University of Madrid, 2003.

[Pro06] AMIDA Project. *Localization and Tracking of Multiple Interlocutors with Multiple Sensors*. AMI Consortium, January 2006.

[PSO97] Marco Matassoni Piergiorgino Svaizer and Maurizio Omologo. Acoustic source localization in a three-dimensional space using crosspower spectrum phase. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1:231–234, April 1997.

[qio02] Qualcomm-icsi-ogi aurora advanced front-end proposal. Technical report, ICSI (International Computer Science Institute), OGI (Oregon Graduate Institute of Science and Technology) and QUALCOMM Inc., 2002.

[RDD01] Dmitry Zotkin Ramani Duraiswami and Larry S. Davis. Active speech source localization by a dual coarse-to-fine search. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 5:3309–3312, May 2001.

[RF07] Francisco JosÃ© Royo FernÃ¡ndez. *Design, Implementation and Evaluation of Techniques for Speech Signal Improvement in Reverberant Environments: Application in Automatic Speech Recognition Systems*. phd thesis, Technical University of Madrid, 2007.

[RL06] Vanesa RodrÃguez Lorenzo. *Desing, Implementation and Evaluation of Acoustic Environment Simulation Tools: Application to Automatic Speech Recognition Systems*. phd thesis, Technical University of Madrid (ETSIT, UPM), 2006.

[Sch86] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34:276–280, March 1986.

[Sel03] Michael L. Seltzer. *Microphone Array Processing for Robust Speech Recognition*. Carnegie Mellon University Pittsburgh, 2003.

[Var02] Krishnaraj Varma. *Time-Delay-Estimate Based Direction-of-Arrival Estimation for Speech in Reverberant Environments*. phd thesis, Virginia Polytechnic Institute, 2002.

[VB88] Barry D. Van Veen and Kevin M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE Signal Processing Magazine*, 5:4–24, April 1988.

[vLK07] David A. van Leeuwen and Matej Konecny. Progress in the amida speaker diarization system for meeting data. *Rich Transcription 2007 Meeting Recognition Evaluation Workshop*, May 2007.

[WK83] M. Wax and T. Kailath. Optimum localization of multiple sources by passive arrays. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31:1210–1217, October 1983.

[WK85] H. Wang and M. Kaveh. Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33:823–831, August 1985.

[WK02] Tim Wickstrom and Sergei Kochkin. Headsets, far field and handheld microphones: Their impact on continuous speech recognition. *Technical Report, EMKAY, a Division of Knowles Electronics*, 2002.

[ZD04] Dmitry N. Zotking and Ramani Duraiswami. Accelerated speech source localization via a hierarchical search of steered response power. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 12(5):499–508, September 2004.

[Zio95] Lawrence J. Ziomek. *Fundamentals of Acoustic Field Theory and Space-Time Signal Processing*. CRC Press, 1995.

Appendix A

Speed of sound

The speed of the sound is variable and can be altered depending on a different set of factors as explained in [Gar07]. However, in our case, we will just concentrate on study its variations when propagating through a gas. The speed of sound is affected by the physical properties of a gas such as its temperature, pressure and humidity. For instance, the higher the temperature, the quicker the propagation speed. An approximate measure of the sound speed is given by the following equation:

$$c = 331.5 + 0.606 \cdot \vartheta \ [m/s] \tag{A.1}$$

where ϑ is the temperature in Celsius scale.

If we compute the speed of sound at the standard air temperature (25 C) with this formula we get: $c(T) = 346.65 \ m/s$.

There is an alternative, more precise equation given by the following formula:

$$c = \sqrt{\frac{\lambda \cdot \kappa \cdot T}{m}} \ [m/s] \tag{A.2}$$

where λ is the adiabatic index of the gas (1.4 for air), κ is the Boltzmann constant ($1.38 \cdot 10^{-23} J \cdot K^{-1}$), T the temperature in Kelvin scale and m is the mass of a single molecule in kilograms ($0.0289645 \frac{kg}{mol} \cdot \frac{1 mol}{6.022 \cdot 10^{23} \ molecules} \simeq 4.81 \cdot 10^{-26} kg$). This formula considers that the transmission of sound in the air is made without energy loss, an approximation very close to reality.

Following this equation, the speed of sound at the standard temperature in Kelvin scale (298 K) is $c(T) \simeq 345.98 \ m/s$.

Appendix B

Windowing

The speech signal is sampled and digitalized so that it can be processed by a computer. These speech signals are then tipically grouped into frames in order to be easily analyzed. Finally, with the aim of getting a clearer analysis, these frames are usually applied different windowing functions that confere different spectral characteristics to the signal considered, i.e. smoothing the frames edges and therefore avoiding abrupt responses in the frequency domain and allowing a more efficient way to deal with a continous signal divided into separated frames.

There are different tipically used window functions. Each of them has different spectral characteristics. Here we show the windows used in this Master Thesis:

- Rectangular

$$w[n] = \begin{cases} 1, & si\ 0 \leq n \leq M \\ 0, & resto \end{cases} \tag{B.1}$$

- Bartlett

$$w[n] = \begin{cases} \frac{2n}{M}, & si\ 0 \leq n \leq \frac{M}{2} \\ 2 - \frac{2n}{M}, & si\ \frac{M}{2} \leq n \leq M \\ 0, & resto \end{cases} \tag{B.2}$$

- Hanning

$$w[n] = \begin{cases} 0.5 - 0.5 \cdot cos(\frac{2\pi n}{M}), & si\ 0 \leq n \leq M \\ 0, & resto \end{cases} \tag{B.3}$$

- Hamming

$$w[n] = \begin{cases} 0.54 - 0.46 \cdot cos(\frac{2\pi n}{M}), & si\ 0 \leq n \leq M \\ 0, & resto \end{cases} \tag{B.4}$$

- Blackman

$$w[n] = \begin{cases} 0.42 - 0.5 \cdot cos(\frac{2\pi n}{M}) + 0.08 \cdot cos(\frac{4\pi n}{M}), & si\ 0 \leq n \leq M \\ 0, & resto \end{cases} \tag{B.5}$$

Figure B.1 shows the shape of the windows defined in (B.1), (B.2), (B.3), (B.4) y (B.5).

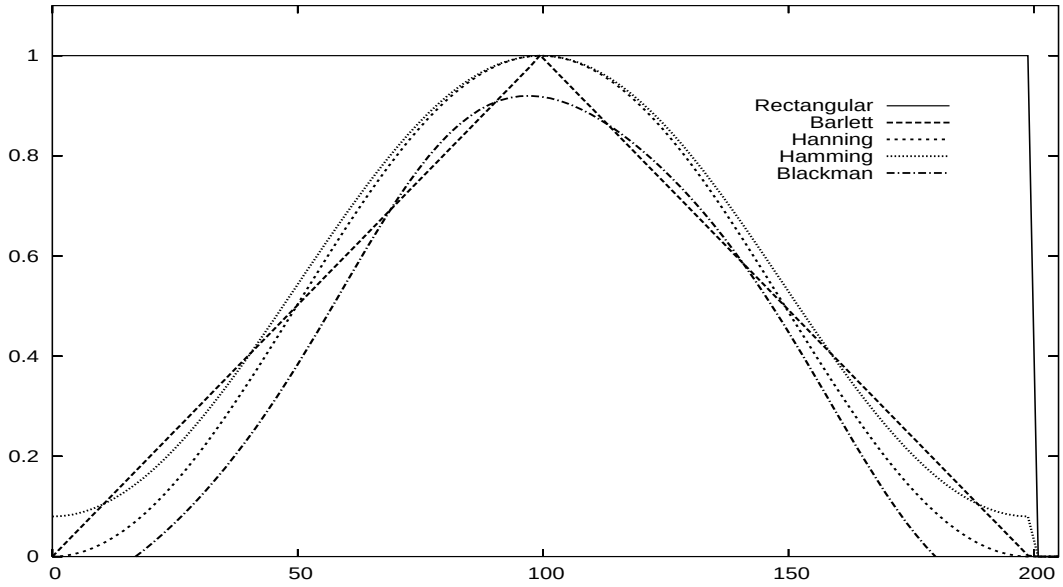

Figure B.1: Commonly used windows (taken from J. Ordonez, [OV03])

Appendix C

User manual

C.1 Introduction

This appendix is intended to show to future users how to use the designed and implemented tools developed during this Master Thesis.

C.2 Steered Response Power (SRP)

C.2.1 Introduction

The program **srp.c** contains the code responsible of performing the Steered Response Power algorithm with speaker localization purposes. In order to obtain an executable file from it we should *'ls'* to its directory and, once there, run the command *make*.

We can see our *srp* program as a black box that will require a set of input informations in order to properly output the speaker position estimates. The general *input-output* structure of the whole process it is depicted in Figure C.1 and exhaustively explained in Section C.2.3.

C.2.2 Command line options

All the main features having any kind of influence in the algorithm behaviour can be totally controlled and tuned thanks to a serie of command line options. A comprehensive enumeration and explanation of them can be found in Section 3.2.4 in the page 56 of this Master Thesis.

For further details about the command line parameters, their default values, their

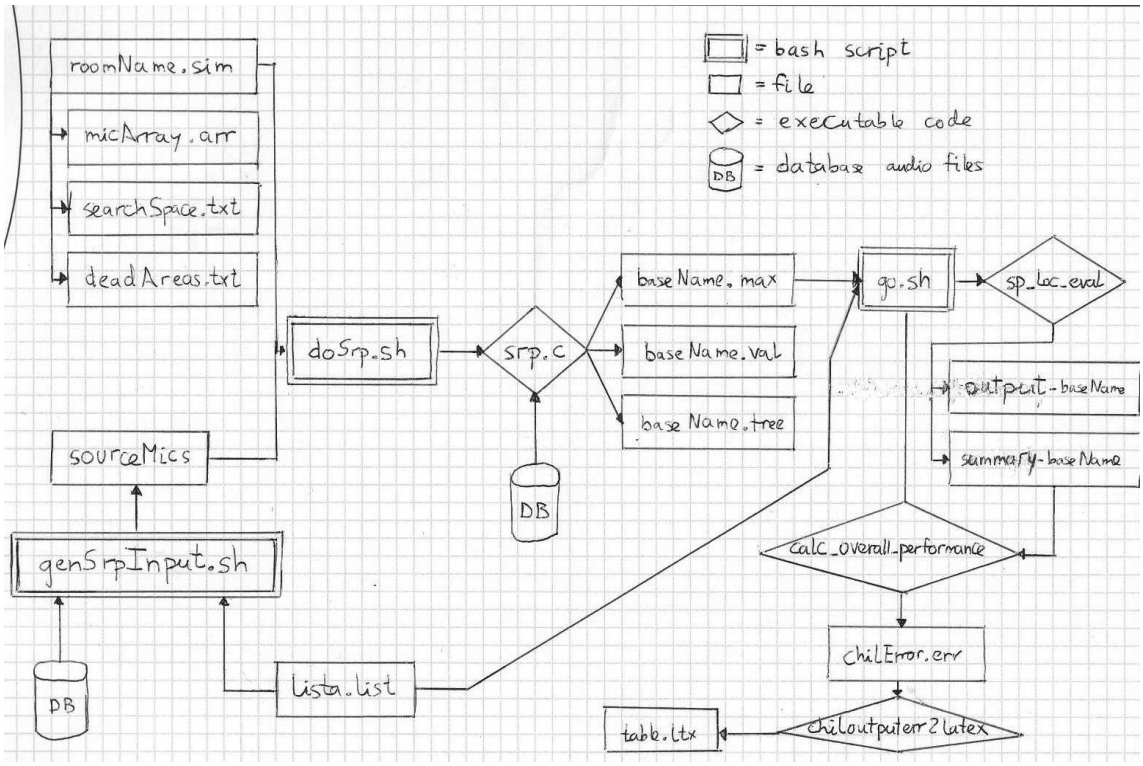

Figure C.1: SRP program block diagram

short and long options, etc. type the command *'./srp –help'*.

C.2.3 Way of operation

As shown in the block diagram in Figure C.1, there are two main types of data taking part in our speaker localization process:

- The **input** data. We have total control to customize it in order to depict any kind of localization scenario: type of speaker's room, microphones geometry, databases, etc. We can specify all this information via several different input files:

  'roomName.sim': This file links all the information relative to the environment geometry. It lists the names of the configuration files, along with their directory paths showing where to find the following specific data:

  - *'micArray.arr'*, this file contains the number of microphones composing the microphone array intended to be used during our simulation. The (x,y,z) coordinates of each one of the microphones involved it is also provided here. The mm is the unit chosen.

  - *'searchSpace.txt'*, this file specifies the spatial limits, in (x,y,z) coordinates,

of the portion of space that our algorithm will inspect in search of a speaker. In addition, it also specifies the isotropic spacing that the program will use to separate consecutive search points in the x,y,z-axis. The mm is the unit chosen once again.

- *'deadAreas.txt'*, this file specifies the number of dead areas (i.e. areas where the speaker is not likely to be such as large tables and wardrobes) along with the coordinates of their (x,y) limits in mm.

'sourceMics-database', the first line of this file provides the program with two figures: The number of microphones involved in the simulation and the number of common characters in the audio files involved, that is to say, the length of the base name after which the simulation output files will be named.

- The **output** data. It contains the localization estimates thrown by our SRP algorithm. Based on this information we will eventually be able to evaluate the performance of our algorithm by means of comparison against the ground truth.

'baseName.max', this file contains several columns. The first one holds the time indexes, in seconds, for which the localization estimates were given. The following columns, grouped in sets of 3 columns, hold, respectively, the (x,y,z) coordinates in mm where our system estimates the speaker to be. There will be as many (x,y,z) estimates as number of maximums were specified to be computed, see command line options in Section 3.2.4 in the page 56. The estimates are sorted from maximum to minimum, from left to rigth.

'baseName.val', this file sorts all the time indexes, in brackets and in second time units, for which the localization estimates were given. For each one of them, first, the frame mean power in dB is given; this information can be useful to check a proper behaviour of the VAD or to ellaborate graphs linking the localization estimate rate with the mean power present in the frames. Secondly, it sorts from maximum to minimon, from top to bottom, as many (x,y,z) coordinates in mm as numbers of maximums were specified in the command line options. Finally, altogether with the (x,y,z) coordinates, the power of each estimate, in watts, is also provided.

'baseName.tree', this output file is only written when the coarse-to-fine version of the SRP algorithm is executed. It contains the same data as the *'.val'* file with the difference that this information it is not only given for the final position estimate of each time index but also for all the prior position estimates, from the coarse to the fine one, through which the coarse to fine algorithm had to go in each case before getting to a final position estimate.

Generating some of the input files listed above, as well as handling the output files containing the position estimates can be sometimes a time-demanding task specially if

there are hundreds or thousands of audio files involved in the simulation, each of them having a high number of time instances when to make a position estimate. For this and some other reasons it was intended to make use of some bash scripts that could help us automatize and reduce such tasks. Bellow, we will describe their function in detail:

- **'genSrpInput-database.sh'**, this script generates the *'sourceMics-database'* input file based on a *'.list'* file which contains all the audio files of an specific database taking part in an specific simulation. This *'.list'* text file can easily be obtained and shaped by means of *'ls'* and *'grep'* bash commands. Then, this script adds to these filenames the proper directory paths where to find them. One such script exists for every different data base in order to consider their specific name formats, etc.

- **'doSrp-database.sh'**, this script allows to easily modify all the different command line options that may have an effect on the simulation we want to perform, see Section 3.2.4 in the page 56. Once all these options have been specified, the script calls the *'srp.c'* program with the appropiate parameters linking it with the appropiate input files *'roomName.sim'* and *'sourceMics-database'*.

- **'go-database.sh'**, once we have got the *'srp.c'* output estimates we want to evaluate a large enough amount of them according to the CHIL standards in order to check the performance of the different variations of our algorithm. This evaluation consists on comparing, audio file by audio file, time index by time index, our algorithm localization estimates against the ground truth position and it is performed by *'sp_loc_eval'*, a program provided by the CHIL consortium. Thanks to this *'go-database.sh'* script we can automate the call to *'sp_loc_eval'* instead of doing it file by file by providing it with the appropiate parameters that it requires along with the appropiate list of output *'.max'* files that we want to evaluate. In this respect, most of the times, this script will make use of the same *'.list'* file which listed the names of all the audio files that were used in that simulation in particular. Once this call is executed, the *'sp_loc_eval'* will create two text files for every of the *'.max'* files listed:

  'summary-baseName.txt', containing the main CHIL evaluation metrics related to that audio filename in particular, see Section 3.2.1 in page 52.

  'output-baseName.txt', containing, time index by time index, the speaker id, error in mm and error classification that resulted from the comparison between our system estimates and the ground truth. Finally, once all the *'summary-baseName.txt* and *'output-baseName.txt'* have been generated, the script *'go-database.sh'* uses them to call the *'calc_overall_performance.pl'* python script, also provided by the CHIL consortium. This python script will output a *'.err'* file containing the average performance of the whole simulation taking into account all the audio files considered. For further details about the CHIL-provided software, *'sp_loc_eval'* and

'calc_overall_performance.pl', try typing *'./sp_loc_eval -h'* and/or reading the *'README.txt'* file accompanying them.

C.2.4 Example

In this Section we will show an example, step by step, on how to entirely set up, run and collect the data thrown by an SRP localization process.

1. **Configure the set of audio files over which the simulation will be performed**

   - Write in a text file the names of those audio files that you want to be involved in the speaker localization. If the audio files names follow an appropiate, systematic format, the use of bash commands such as *grep* toguether with the connection operators ¨|¨ or ¨>¨ can help to easily list a subset of files with an specific feature even when located within an extensive database. Here are some specific instances for the HIFI database. These instances are to be run from the data base directory containing the audio files:

     ls | grep .wav | grep LFD > lista-HIFI.list, selects just those audio files uttered by user LFD.

     ls | grep .wav | grep P1 > lista-HIFI.list, selects just those audio files uttered from position P1. For details about the specific formatting of each database audio files check Section 3.3 in page 61.

   - Run the *genSrpInput.sh* script in order to automatically provide the audio files selected with their corresponding directory path. Basically, this script needs to be specified the name and path of the *lista.list* file toguether with the root database directory where to find the audio files, the number of common characters in the audio files format and the number of microphones involved. As a result, it will output the *sourceMics* text file that will be directly used by the SRP program to get all the information it needs regarding the set of audio files over which it has to perform localization estimates, see Figure C.1.

2. **Configure the environment geometry that you want to be used during the simulation**

   - Go to the directory containing the *roomName.sim* configuration file and set its parameters: the directories where to find the particular configuration files holding the geometry specifications, the number of microphones arrays that are going to be involved toguether with the name of the particular *micArray.arr* files that is going to be used, the name of the particular *searchSpace.txt* and *deadAreas.txt* files that will be taken into account during the simulation, the

particular file format used in this case, etc. Each of these issues can be modified under their corresponding field within the *roomName.sim* configuration file.

- Go to the directory containing the *micArray.arr* file (or files if several microphone arrays are to take part in the task) and fill them with the appropiate info: the number of microphones composing that array and their exact (x,y,z) coordinates in mm within the room. This way, out of the given set of microphones that actually recorded the audio files, any possible kind of new subarray configuration can be tested in the SRP algorithm just by creating its corresponding *micArray.arr* file and filling it in with how many microphones take part on it and which are their spatial coordinates.
- Go to the directory containing the *searchSpace.txt* file and fill it with the appropiate info you want to apply for your particular simulation: the spatial boundaries, top vertex and bottom vertex, to which your search space will be reduced as well as the specific distance in mm separating two neighbouring points in the grid whose SRP (Steered Response Power) will be evaluated. This scheme allows complete flexibility, you could, for instance, select finer grids for the (x,y) coordinates and a grosser one in z. Likewise, you could restrict your speaker search to specific areas or heights in your room instead of performing it along the whole room.
- Go to the directory containing the *deadAreas.txt* file and fill it with the appropiate info you want to apply for your particular simulation, that is to say, the number of dead areas involved and the (x,y) coordinates in mm of their limits.

3. **Launch the experiment with the appropiate parameters** This can be easily done by configuring the appropiate *doSrp-database.sh* script with the desired values for the command line options that you want to use during your test. Plenty of information about these command line options can be read in Section 3.2.4 in the page 56 and by typing *./srp –help* in the SRP program directory. Anyway, there are some peculiarities about some of these parameters that I would like to comment further here:

   - *fs*, the fs selected will only apply in the case of raw audio files. Otherwise, the program will automatically read from the audio file and consider the actual fs at which it was recorded.
   - *FFT size*, must always be greater or equal than the frame size. Additionally and for computational reasons, it is convenient that it is set to be a power of 2.
   - *Dir Input Files*, this command line option specifies the directory where the program can find its required input files, that is to say, the *sourceMics* and *room-*

Name.sim files.

- *First Time Index* and *End Audio File* determine the time boundaries, in seconds, of the piece of audio file to be analyzed by the program. Naturally, these values should be positive and *End Audio File* must always be greater than *First Time Index*. In case *End Audio File* is set beyond the time duration limits of a file it will automatically be set to match that file length. Finally, when we just want our SRP program to analyze the whole duration of our audio files we need to set both parameters, *First Time Index* and *End Audio File*, to 0.
- *Frequency SRP Flag*, when activated the FSRP method is applied and, consecuently, the *Round Flag* stops having any effect on the program performance since this parameter it is only meaningful for the TSRP method. When not active, it means the TSRP alternative will be chosen instead.
- *Round Flag*, can take the following values: 0 (no rounding is applied), 1 (the rounding is done to the closest integer) or > 1 (the rounding is done according to a linear interpolation).
- *Low Frequency*, its value can be used in two possible cases: 1) if the *Filter Flag* is activated, it sets the low cut-off frequency and 2) if the *Coarse to Fine Flag* is activated, it sets the starting cut-off frequency in the first step of the coarse to fine scheme.
- *Maximum Distance* value only applies when the coarse to fine method is activated (this parameter is the one setting the threshold at which the coarse to fine scheme stops).
- *Fixed Threshold Flag*, if active (set to 1) it triggers a fixed noise masking threshold scheme, if not (set to 0) an adaptative threshold will be chosen instead. However, this flag will only have an influence in the case that the *Noise Masking Flag* has been previously activated.
- *Noise Size Secs* parameter is in charge of specifying the length, in seconds, of the frame located at the beginning of the audio file over which the noise estimation will be done. Of course, this parameter will only have an influence in the case that the *Noise Masking Flag* has been previously activated.
- *Noise Threshold* parameter sets the number of dB the audio signal has to overpass the noise level in order not to be discarded. Of course, this parameter will only have an influence in the case that the *Noise Masking Flag* has been previously activated.

Once we have set our desired simulation options the script *doSrp-database.sh* automatically composes and launches the appropiate order to start the localization estimation. Some order examples could be:

- For the **HIFI** database (by order of appearance):

  /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/srp -v edecanRoom-HIFIMM1-srp.sim -M sourceMics-HIFI -l /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/ -f 48000 -n 0.32 -s 0.04 -p 16384 -w m -x 3 -r 3 -j 0 -k 0 -u 0 -z 2 -q 0 -a 0 -b 150 –freq-srp 0 -R 1 -N 0 -X 0 -Z 0.05 -G 24 -F 0 -L 1000 -H 8000 -E 0 -W 0

  -The SRP program is launched from its path, */home/ithil/ccastrogo/reposito/proyecto/far-field/srp/* in this case, making the execution independent of where *doSrp-HIFI.sh* was called.

  -The names of the required configuration input files are *sourceMics-HIFI* and *edecanRoom-HIFIMM1-srp.sim* this time and they can be found at path */home/ithil/ccastrogo/reposito/proyecto/far-field/srp/*.

  -fs = 48 KHz.

  -Frame size = 0.32 secs.

  -Frame shift = 0.04 secs.

  -FFT size = 16384 points.

  -Window = Hamming.

  -Number of maximums = 3.

  -Correlation method = 3 (GCC-PHAT).

  -Starting and end time indexes determined to fit the whole audio file lenght (both are set to 0).

  -Interpolation flag not active (set to 0).

  -Interpolation rate set to x2 (it does not apply since the interpolation flag is not active).

  -Dicard flag not active (set to 0, meaning that the discard of powerless frames will not be active). This option will typically remain unactive since it belongs to early versions of the SRP program. At the moment, this task is performed as a previous step by the VAD.

  -Coarse to fine flag not activated (set to 0).

  -Maximum distance = 150 mm (it does not apply since the coarse to fine method is not active this time).

  -Frequency SRP flag not active (set to 0), meaning that TSRP alternative will be the one performed.

  -Rounding flag = 1, implying that the rounding in the TSRP method will be done to the closest integer (it applies since FSRP is not active).

  -Noise masking flag not active (set to 0).

  -Fixed threshold flag not active (set to 0) meaning that an adaptative noise threshold will be chosen instead (it does not apply anyway since the noise

masking strategy has not been selected).

-Noise size secs = 0.05 secs (it does not apply anyway since the noise masking strategy has not been selected).

-Noise threshold = 12 dB (it does not apply anyway since the noise masking strategy has not been selected).

-Filter flag not active (set to 0).

-Low frequency = 1000 Hz (it does not apply since not the filter flag, neither the coarse to fine flag have been activated).

-High frequency = 8000 Hz (it does not apply since the filter flag has not been activated).

-Discard dead aread flag not active (set to 0).

-Distance weighting flag not active (set to 0).

- For the **AV16.3** database:

  /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/srp -M sourceMics-AV16.3 -v idiapRoom-AV163-srp.sim -l /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/ -f 16000 -n 0.16 -s 0.04 -p 4096 -w r -x 3 -r 3 -j 0 -k 0 -u 1 -z 3 -q 0 -a 0 -b 250 -F 1 -L 4000 -H 8000 –freq-srp 0 -R 1 -N 1 -X 0 -Z 0.05 -G 24 -E 0 -W 1

  sourceMics-AV16.3 and *idiapRoom-AV163-srp.sim* selected as configuration input files, fs = 16000 Hz, frame size = 0.16 secs., frame shift = 0.04 secs., FFT size = 4096 points, Rectangular window, number of maximums = 3, GCC-PHAT correlation method, whole audio file lenght analyzed, interpolation active at 2x, no discard of powerless frames, no coarse to fine, pass-band filtering performed in the band [4000, 8000] Hz, TSRP method selected with rounding to the closest integer, +24 dB adaptative noise masking applied (noise estimation performed over the first 0.05 secs. of audio), no discard of dead areas and microphone distance weighting applied.

- For the **SONY** database:

  /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/srp -M sourceMics-SONY -v edecanRoom-HIFIMM1-srp.sim -l /home/ithil/ccastrogo/reposito/proyecto/far-field/srp/ -f 48000 -n 0.64 -s 0.04 -p 32768 -w m -x 3 -r 3 -j -k 0 -u 0 -z 2 -q 0 -a 0 -b 500 -F 0 -L 1000 -H 4000 –freq-srp 0 -R 1 -N 0 -X 1 -Z 0.05 -G 12 -E 0

  sourceMics-SONY and *idiapRoom-AV163-srp.sim* selected as configuration input files, fs = 48000 Hz, frame size = 0.64 secs., frame shift = 0.04 secs., FFT size = 32768 points, Hamming window, number of maximums = 3, GCC-PHAT correlation method, whole audio file lenght analyzed, no interpolation applied, no discard of powerless frames, coarse to fine applied: starting cut-off frequency = 1000 Hz and 500 mm separation between points in the finer level

as stop condition, no filtering performed, TSRP method selected with rounding to the closest integer, no noise masking applied, no discard of dead areas and no microphone distance weighting applied.

4. **Collect the output data and evaluate it**

   We will now focus on the *baseName.max* output files. For each of the audio files analyzed (those that were listed on *lista.list* on the first place) one of these *.max* files will be created showing for each time index the (x,y,z) coordinates, in mm, where our SRP algorithm estimates the speaker to be. We want to compare these estimates against the golden standard provided by the ground truth. As the number of estimates to compare is often prohibitively large (time indexes times the number of audio files evaluated) we have developed an automatic script, *go.sh*, that will perform this task for us. All that we need to specify to this *go.sh* script is:

   - *Output Basename*, it defines the label after which the *output-basename.txt* files will be named.
   - *Summary Basename*, it defines the label after which the *summary-basename.txt* files will be named.
   - *Threshold Lecturer* and *Threshold Audience* set the distance limit, in mm, according to which the errors will be classified either into Fine or Gross errors in the Lecturer and Audience scenarios respectively. Typically, this thresholds will be set to 500 mm in order to meet CHIL specifications. For more details, check [OM06].
   - *Time Step*, it defines the time step, in seconds, between consecutive location estimates.
   - *Type of Error*, it can be either set to be *ae* (average error) or *rms* (root mean square error).
   - *Lecturer ID*, out of all possible speakers, this label defines over which speaker in particular the evaluation will be performed. When set to *all* it will lead to aggregated evaluation results for all speakers.
   - *Error*, this label can be either *distance*, implying that errors will be shown in mm in the (x,y,z) coordinates or *azimuth*, implying that errors will be shown in angles in the (r,θ,ϕ) coordinates.

   The previous parameters are in fact required in order to make a call to the *sp_loc_eval* program. Provided by the CHIL organization, this software is in charge of comparing CHIL-formatted localization estimates against CHIL-formatted ground truths. Anyway, apart from these parameters, *go.sh* also requires to know:

- *Overall Filename*, it defines the label after which the *.err* file containing the aggregated results for all the audio files involved will be named.
- *Reference Directory*, it defines the path where to find the ground truth files.
- *Input Directory*, it defines the path where to find the input files to *go.sh*, that is to say, the *.max* files.
- *SRP List File*, it defines the path and name of the same *.list* file that was used in the first stage of the process to enumerate the audio files that were taking part on the simulation. This info is also required at this last stage because it will be necessary to evaluate the output results that each of them threw.

Once this information is given the *go.sh* script automatically proceeds as follows:

(a) From the *.list* file tt reads the name of the first audio file involved in the simulation.

(b) It makes a call to *sp_loc_eval* which, with the appropiate parameters listed above, will make a comparison between that specific audio file SRP results and its corresponding ground truth. The evaluation results will be stored in the *output-baseName.txt* and *summary-baseName.txt* files. It is important to note that these files will have a different name each time since the audio file base name changes.

(c) It goes again to the first step and proceeds like this until all the audio files listed in the *.list* file have been evaluated.

(d) After this, *calc_overall_performance* perl script is called having as inputs all the *output-baseName.txt* files generated. This call will create the aggregated final results for all the audio files involved. These results will be stored in the *.err* file.

5. **Present the obtained results**

Finally, the *chiloutputerr2latex* program was designed to read directly the info contained in the *.err* files and convert it to LaTeX tables format (stored in *.ltx* files).