Universidad de Alcalá
COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

ACTA DE EVALUACIÓN DE LA TESIS DOCTORAL

(FOR EVALUATION OF THE ACT DOCTORAL THESIS)

Año académico (*academic year*): 2017/18

DOCTORANDO (*candidate PHD*): **SÁNCHEZ HEVIA, HÉCTOR ADRIÁN**
D.N.I./PASAPORTE (*Id.Passport*): **32877803Q**
PROGRAMA DE DOCTORADO (*Academic Committee of the Programme*): **D445-TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES**
DPTO. COORDINADOR DEL PROGRAMA (*Department*): **TEORÍA DE LA SEÑAL Y COMUNICACIONES**
TITULACIÓN DE DOCTOR EN (*Phd title*): **DOCTOR/A POR LA UNIVERSIDAD DE ALCALÁ**

En el día de hoy 08/06/18, reunido el tribunal de evaluación, constituido por los miembros que suscriben el presente Acta, el aspirante defendió su Tesis Doctoral **con Mención Internacional** (*In today assessment met the court, consisting of the members who signed this Act, the candidate defended his doctoral thesis with mention as International Doctorate)*, elaborada bajo la dirección de (*prepared under the direction of)* ROBERTO GIL PITA.

Sobre el siguiente tema (*Title of the doctoral thesis*): *EFFICIENT MULTICHANNEL ALGORITHMS FOR IMPULSIVE SOUND SOURCE ANALYSIS*

Finalizada la defensa y discusión de la tesis, el tribunal acordó otorgar la CALIFICACIÓN GLOBAL[1] de (**no apto, aprobado, notable** y **sobresaliente)** *(After the defense and defense of the thesis, the court agreed to grant the GLOBAL RATING (fail, pass, good and excellent)*: SOBRESALIENTE

Alcalá de Henares, a 8 de Junio de 2018

Fdo. *(Signed)*: A. GONZALEZ

Fdo. *(Signed)*: D. Ayllón

Fdo. *(Signed)*: Manuel Rosa Zurera

FIRMA DEL ALUMNO (candidate's signature),

Fdo. *(Signed)*: H. Sánchez Hevia

Con fecha 16 de julio de 2018 la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado, a la vista de los votos emitidos de manera anónima por el tribunal que ha juzgado la tesis, resuelve:
☒ Conceder la Mención de "Cum Laude"
□ No conceder la Mención de "Cum Laude"

La Secretaria de la Comisión Delegada

[1] La calificación podrá ser "**no apto**" "**aprobado**" "**notable**" y "**sobresaliente**". El tribunal podrá otorgar la mención de "cum laude" si la calificación global es de sobresaliente y se emite en tal sentido el voto secreto positivo por unanimidad. (*The grade may be "**fail**" "**pass**" "**good**" or "**excellent**". The panel may confer the distinction of "cum laude" if the overall grade is "Excellent" and has been awarded unanimously as such after secret voting.).*

INCIDENCIAS / OBSERVACIONES:
(Incidents / Comments)

Universidad
de Alcalá
**COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO**

En aplicación del art. 14.7 del RD. 99/2011 y el art. 14 del Reglamento de Elaboración, Autorización y Defensa de la Tesis Doctoral, la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado y Doctorado, en sesión pública de fecha 16 de julio, procedió al escrutinio de los votos emitidos por los miembros del tribunal de la tesis defendida por *SÁNCHEZ HEVIA, HÉCTOR ADRIÁN*, el día 08/06/18, titulada *EFFICIENT MULTICHANNEL ALGORITHMS FOR IMPULSIVE SOUND SOURCE ANALYSIS*, para determinar, si a la misma, se le concede la mención "cum laude", arrojando como resultado el voto favorable de todos los miembros del tribunal.

Por lo tanto, la Comisión de Estudios Oficiales de Posgrado **resuelve otorgar** a dicha tesis la

MENCIÓN "CUM LAUDE"

Alcalá de Henares, 18 de julio de 2018
EL VICERRECTOR DE INVESTIGACIÓN Y TRANSFERENCIA

Javier de la Mata

UNIVERSIDAD DE ALCALÁ · VICERRECTORADO DE INVESTIGACIÓN Y TRANSFERENCIA

F. Javier de la Mata de la Mata

Copia por e-mail a:
Doctorando: SÁNCHEZ HEVIA, HÉCTOR ADRIÁN
Secretario del Tribunal: MANUEL ROSA ZURERA. C.U. Dpto. Teoría de la Señal y las Comunicaciones. UAH
Director/a de Tesis: ROBERTO GIL PITA

UNIVERSIDAD DE ALCALÁ. PATRIMONIO DE LA HUMANIDAD

Universidad de Alcalá

ESCUELA DE DOCTORADO
Servicio de Estudios Oficiales de Posgrado

DILIGENCIA DE DEPÓSITO DE TESIS.

Comprobado que el expediente académico de D./Dª Héctor Adrián Sánchez Hevia reúne los requisitos exigidos para la presentación de la Tesis, de acuerdo a la normativa vigente, y habiendo presentado la misma en formato: ☒ soporte electrónico □ impreso en papel, para el depósito de la misma, en el Servicio de Estudios Oficiales de Posgrado, con el nº de páginas: 203 se procede, con fecha de hoy a registrar el depósito de la tesis.

Alcalá de Henares a 26 de abril de 2018

UNIVERSIDAD DE ALCALÁ
SERVICIO DE ESTUDIOS OFICIALES DE POSGRADO

Mª del Val Jiménez Blázquez

Fdo. El Funcionario

Universidad de Alcalá

PhD. Program in Information and Communication Technologies

Efficient Multichannel Algorithms for Impulsive Sound Source Analysis

PhD. Thesis Presented by

Héctor A. Sánchez Hevia

Supervisor

Roberto Gil Pita

Alcalá de Henares, April 26th, 2018

Universidad de Alcalá

DEPARTAMENTO DE TEORÍA DE
LA SEÑAL Y COMUNICACIONES
Escuela Politécnica Superior
Campus Universitario s/n
28805, Alcalá de Henares (Madrid)
Telf: +34 91 885 66 90
Fax: +34 91 885 66 99
dpto.teoriadelasenal@uah.es

UNIVERSIDAD DE ALCALÁ, PATRIMONIO DE LA HUMANIDAD

D. SANCHO SALCEDO SANZ, Catedrático de Universidad del Área de Conocimiento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá, en calidad de Coordinador de la Comisión Académica del programa de Doctorado en Tecnologías de la Información y las Comunicaciones,

CERTIFICA

Que D. Héctor Adrián Sánchez Hevia ha realizado la tesis **"Efficient Multichannel Algorithms for Impulsive Sound Source Analysis"**, en el Departamento de Teoría de la Señal y Comunicaciones bajo la dirección del Dr. Roberto Gil Pita, cumpliéndose todos los requisitos para la tramitación que conduce a su posterior defensa.

Alcalá de Henares, 26 de Abril de 2018.

UNIVERSIDAD DE ALCALÁ * DPTO. DE TEORÍA DE LA SEÑAL Y COMUNICACIONES *

Fdo. Dr. Sancho Salcedo Sanz

Universidad de Alcalá

DEPARTAMENTO DE TEORÍA DE
LA SEÑAL Y COMUNICACIONES
Escuela Politécnica Superior
Campus Universitario s/n
28805, Alcalá de Henares (Madrid)
Telf: +34 91 885 66 90
Fax: +34 91 885 66 99
dpto.teoriadelasenal@uah.es

UNIVERSIDAD DE ALCALÁ, PATRIMONIO DE LA HUMANIDAD

D. ROBERTO GIL PITA, Profesor Titular de Universidad del Área de Conocimiento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá,

CERTIFICA

Que la tesis **“Efficient Multichannel Algorithms for Impulsive Sound Source Analysis”**, presentada por D. Héctor Adrián Sánchez Hevia, realizada en el Departamento de Teoría de la Señal y Comunicaciones bajo mi dirección, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y defensa.

Alcalá de Henares, 26 de Abril de 2018.

UNIVERSIDAD DE ALCALÁ * DPTO. DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

Fdo. Dr. Roberto Gil Pita

Abstract

Acoustic events most commonly associated with security threats such as gunshots, explosions, collisions or glass shattering, among others, are impulsive in nature. This category of sounds is characterized by a large sudden onset, a short time duration (typically a few milliseconds) and wide spectral dispersion, which make them suitable candidates for real-time automated detection. Yet detection alone is not relevant enough for some applications in which the localization and/or identification of the impulsive source are the main concerns, so further analysis is required.

By means of a network of acoustic sensors it is possible to locate the emitting source by exploiting the differences within the signals recorded at each microphone, and having previous knowledge about the time instant and location of each of the captures. Although proper node localization and synchronicity on wireless networks are currently unresolved issues undergoing active research.

On the matter of identification, impulsive sound event recordings usually have more to do with the impulse response of the environment rather than the source itself, which proves as a problem for robust classification. However it is possible to take advantage of the spatial diversity provided by a sensor network to identify the source collaboratively using different data fusion strategies.

This thesis focuses on the research and development of novel multichannel algorithms for impulsive sound source analysis, including four distinct but not necessarily independent problems, namely: acoustic event detection and classification, sound source localization and node self-localization. In general terms, this proposal aims at developing a framework for collaborative processing in which the spatial diversity provided by a group of microphones is exploited to solve each of the presented problems. At the same time, the goal is to improve multichannel acoustic analysis of impulsive events by taking advantage of the information obtained from the resolution of some of the problems to increase the performance of later stages. Additionally, the aim of the thesis is to develop efficient techniques suitable to be implemented on a wireless acoustic sensor network in which using other methods may not be feasible due to a series of constraints, such as limited computing power, or wireless bandwidth. Thus, throughout the whole thesis, there is a constant tradeoff between performance and computational complexity.

The proposals in this thesis come from the necessity for better automated impulsive sounds analysis systems given that they have practical applications in many fields such as forensics, surveillance and security, gun control or military tactics, to name a few.

Resumen

Los eventos acústicos que más comúnmente son asociados con amenazas para la seguridad, como pueden ser disparos, explosiones, golpes o rotura de cristales, entre otros, tienen una naturaleza impulsiva. Esta categoría de sonidos se caracteriza por tener un ataque pronunciado y casi instantáneo, muy corta duración (típicamente pocos milisegundos) y un amplio contenido espectral, lo que los convierte en buenos candidatos para su detección automatizada en tiempo real. Sin embargo, la detección por si misma puede no ser suficientemente relevante para algunas aplicaciones en las que el objetivo principal sea la localización o la identificación de la fuente, haciéndose necesario un análisis más profundo.

Mediante el uso de una red de sensores acústicos es posible localizar la fuente que ha originado el sonido aprovechándose de las diferencias entre las señales captadas por varios micrófonos, siempre y cuando se tenga información sobre los instantes de recepción y la posición de los distintos sensores. Aunque la obtención de sincronismo en la red y la estimación de las posiciones de los nodos son problemas que aún no están completamente resueltos y siguen bajo estudio.

Respecto a la identificación de los sonidos, las grabaciones de eventos impulsivos normalmente están más relacionadas con la respuesta al impulso del entorno que con el sonido generado por la fuente, lo cuál es un gran problema de cara a una clasificación robusta. No obstante, es posible aprovecharse de la diversidad espacial proporcionada por una red de sensores para identificar la fuente colaborativamente mediante la aplicación de varias estrategias de fusión de información.

Esta tesis está enfocada a la investigación y desarrollo de nuevos algoritmos multi canal para el análisis de sonidos impulsivos, lo que incluye cuatro problemas diferenciados pero no necesariamente independientes: detección y clasificación de eventos acústicos, localización de fuentes de sonido y auto localización de nodos. En términos generales, la intención de esta propuesta es el desarrollo de un sistema colaborativo en el cual la diversidad espacial proporcionada por una red de sensores sea explotada para resolver cada uno de los problemas mentados. Al mismo tiempo, el objetivo es la mejora del análisis de eventos impulsivos mediante el aprovechamiento de la información obtenida al resolver algunos de los problemas de cara a mejorar el desempeño en etapas posteriores. Adicionalmente, con esta tesis se pretenden desarrollar técnicas eficientes que permitan su implementación en una red inalámbrica de sensores acústicos en la cual el uso de otros métodos puede que no sea viable debido a restricciones tales como una capacidad de computo limitada o un ancho de banda restringido. De esta forma, a lo largo de toda la tesis se mantiene un constante compromiso entre las cualidades técnicas y la carga computacional de los algoritmos.

Las propuestas de esta tesis vienen dadas por la necesidad de encontrar nuevas soluciones para el análisis de fuentes acústicas impulsivas dado cuenta con aplicaciones prácticas en múltiples campos tales como: vigilancia y seguridad, control de armas, análisis forense o táctica militar, por nombrar unos pocos.

Acknowledgements

Gracias a los que os habéis esforzado más que yo por llegar hasta aquí.
Por no dejarme perderme. Por todos los sinsentidos y todas las rabietas que os ha tocado aguantar. Aunque solo sea un poco espero que os haga felices.

Héctor

I would like to thank all the people that have helped and supported me during the development of this thesis. I had never planned to get a Ph.D, yet, life is but what goes on while you're busy making other plans.

Firstly, I would like to express my gratitude to my supervisor for fooling me into starting a research career. I would also like to thank everyone else from the ASP research group, likewise the rest of my colleagues from the Signal Theory and Communications department. I am also grateful to all the people from Brown University and Tampere University of Technology for welcoming me and giving me valuable advise.

Last but not least, I would like to thank however is taking the time to look at my work. I hope you find something interesting, I tried my best.

This work has been funded by the Spanish Ministry of Economy and Competitiveness/FEDER under projects TEC2012-38142-C04-02 and TEC2015-67387-C4-4-R, and research grant BES-2013-065416.

Contents

List of Figures

List of Tables

List of Acronyms

K-NN	K-Nearest Neighbors.
μC	MicroController.
ADC	Analog to Digital Converter.
AED	Acoustic Event Detection.
ANN	Artificial Neural Network.
ASR	Automatic Speech Recognition.
ATF	Acoustic Transfer Function.
CQT	Constant-Q Transform.
DAC	Digital to Analog Converter.
DCT	Discrete Cosine Transform.
DFT	Discrete Fourier Transform.
DoA	Direction of Arrival.
DSP	Digital Signal Processor.
DWT	Discrete Wavelet Transform.
ER	Error Rate.
FC	Fusion Center.
FCME	Full Covariance Matrix Estimation.
FFT	Fast Fourier Transform.
FIR	Finite Impulse Response.
FN	False Negatives.
FP	False Positives.
GA	Genetic Algorithm.
GCC	Generalized Cross-Correlation.
GMM	Gaussian Mixture Model.
GPS	Global Positioning System.
HMM	Hidden Markov Model.

IDFT	Inverse Discrete Fourier Transform.
IMU	Inertial Measurement Unit.
ISTFT	Inverse Short Time Fourier Transform.
LDA	Linear Discriminant Analysis.
LOOCV	Leave-One-Out Cross-Validation.
LS	Least Squares.
LTI	Linear Time Invariant.
MAC	Multiply ACcumulate.
MDS	MultiDimensional Scaling.
MFCC	Mel Frequency Cepstrum Coefficients.
ML	Maximum Likelihood.
MLP	Multi-Layer Perceptron.
NCME	Naive Covariance Matrix Estimation.
PCA	Principal Component Analysis.
PDF	Probability Density Function.
PHAT	PHAse Transform.
PM	Program Memory.
RAM	Random Access Memory.
RF	Radio Frequency.
RndF	Random Forest.
ROC	Receiver Operating Characteristic.
RSS	Received Signal Strength.
SNR	Signal-to-Noise Ratio.
SPL	Sound Pressure Level.
SRP	Steered Response Power.
STFT	Short Time Fourier Transform.
SVD	Singular Value Decomposition.
SVM	Support Vector Machine.
TDoA	Time Difference of Arrival.
TN	True Negatives.
ToA	Time of Arrival.

ToF	Time of Flight.
TP	True Positives.
TRx	Transceiver.
Tx	Transmitter.
WASN	Wireless Audio Sensor Network.
WMA	Wireless Microphone Array.

List of Symbols

$x(t)$	Discrete-time signal.
$X(k)$	Discrete Fourier Transform of $x(t)$.
$\mathbf{x}$	Vector.
$\mathbf{x}$	Matrix.
x	Scalar.
$\mathcal{F}(x)$	Function of x.
L	Frame length.
f_s	Sampling rate.
c	Speed of sound.
N	Number of sources.
M	Number of microphones.
d	Distance.
$\mathbf{p}$	Cartesian coordinates vector.
τ	Time lag.
δ	Time of Flight of a signal.
τ_{jk}	Time Difference of Arrival between the j^{th} and the k^{th} microphone.
α_{jk}	Direction of Arrival between the j^{th} and the k^{th} microphone.
ϕ	Azimuth angle.
Δ_m	Clock offset of the m^{th} microphone.
$\mathbf{x}_l$	Feature vector of the l^{th} frame.
F	Length of the feature vector.
D	Decision.
y_0	Threshold.
$\mathbf{Q}$	Pattern matrix.
$\mathbf{v}$	Weight vector.
$\mathbf{t}$	Target vector.
$*$	Convolution operator.
$\|\|.\|\|$	Euclidean norm.
$\|.\|$	Absolute value or magnitude.
$(.)^T$	Matrix transpose operator.
$(.)^*$	Complex conjugate operator.
$(.)^{-1}$	Matrix inverse operator.

PART I Preliminary Work

If we had machines that could hear as humans do, we would expect them to be able to easily distinguish speech from music and background noises, to pull out the speech and music parts for special treatment, to know what direction sounds are coming from, to learn which noises are typical and which are noteworthy.

Richard F. Lyon

CHAPTER 1 Introduction and Motivation

1.1 Obtaining Information from Sound

Most living things have the ability to perceive sound in one way or another, from humans to insects and even some kinds of plants[1]. Life forms have evolved to obtain information about their surroundings by different means, one of the most prevalent being the ability to sense vibrations. These vibrations, propagated as pressure waves either thought solids or fluids (see Figure 1.1), are what we commonly refer to as sound. When our inner ear detects small changes in air pressure, i.e., we hear a sound; the information we extract from it is the result of a series of complicated processes that are automatically performed by our brain. Wether the sound we hear is a human voice, music or even a random noise, our brain is capable of perceiving it in background noise, getting an approximate idea of its place of origin and identifying the source that produced it. In the context of signal processing these abilities can be seen as sound source detection, localization and classification.

By the time a sound reaches our ears it conveys information about its source of origin, by means of its frequency content and temporal evolution, and it also contains spatial information in the form of subtler cues, such as the appearance of reflections and reverberation and the timing differences between the signals received at each ear. Our ability to get information from sound is greatly related to our previous experiences (i.e. what we have learned so far). We are capable of recognizing a familiar voice in a crowd and we are 'somewhat' able to locate our phone when it rings. These two examples, while both related to *a priori* knowledge are not exactly alike. Generally speaking, we don't need spatial information to recognize sounds and we don't need to identify the source of the sound in order to locate it, although in any case, we first need to notice the sound in question.

We recognize sounds because we have learned about their particularities through exposure; identical twins' parents may be able to tell their voices apart, unlike the rest of the world, and an experienced car mechanic may be able to find the problem with a car just by hearing it. Usually, the more we hear something, the better our training gets, therefore, the better we are at recognizing it. In a similar way, we easily notice certain sounds because they feel out of place, i.e., they do not match what we have learnt to expect from a certain environment.

Our sound localization mechanisms are also rooted in learning, perhaps on a different basis since they are adapted to our particular physique. Based mostly in the cues derived from the way sound interacts with us, we have unconsciously learnt to interpret them to locate sounds using accurate spatial feedback from other sensorimotor systems. The shape our ears, the distance between them, the shape of our head, and many other factors are what makes us able to locate sound. If we were to

[1] Studies show that plants are capable of responding to certain sounds, e.g., bee buzzing at a particular frequency has been shown to stimulate pollen release, and caterpillar chewing sounds showed an increased production of chemical toxins.

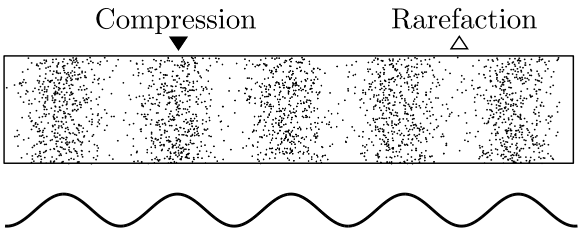


Figure 1.1: Schematic representation of pressure wave propagating through air.

alter some of this factors we would lose some (if not all) of our sound localization ability[2], at least until we relearn how sound interacts with our new "configuration".

In the greater scheme of things we do not even have a good sense of hearing. It is common knowledge that bats and dolphins use echolocation to navigate their environment and to find prey, and any dog/cat owner would know that his pet is much more reliable at detecting sound.

For a long time now, we have been trying to replicate (or even to increase) our cognitive abilities artificially, and hearing is not an exception. However, emulating these seemingly trivial tasks on a machine is not an easy feat, so it should not come as a surprise that they have been the subject of innumerable research efforts over the last decades. Detection, classification and localization are classic signal processing paradigms, found in many fields, not only those based on acoustic signals. Detection and classification are closely related to each other since they are two of the main objectives of machine learning. It is possible to find examples of both of them in the literature for every imaginable application, ranging from self-driving cars to medical diagnosis. Localization is also a widely studied topic, more so in recent years with the increased interest in sensor networks and array-based techniques. Applications of localization can also be counted by the thousands, ranging from GPS navigation to seismology.

Even though many of the mentioned applications are based on a common set of principles, the methodologies followed by every field are not alike, thus this thesis builds upon previous work more closely related to the case in point; acoustic signals and more specifically impulsive sound events.

1.2 Impulsive sound sources

Natural occurring sound is for the most part a random non-stationary signal than can take many different forms. In audio signal processing, sound is basically divided into three categories: speech, music and "noise", the latter including every sound that does not fit into the other labels. The word noise most often than not relates to an unwanted signal, however for some applications a particular type of noise might be the signal of interest, as it is the case.

An impulsive sound source is an acoustic source that generates impulse-like noise[3]; a generic term which includes all forms of high-intensity short-duration sounds, e.g., gunshots, explosions,

[2] Localization of sound elevation can be dramatically degraded by a temporal modification of the outer ear (pinnae) shape, using silicone molds [HVRVO98].

[3] Do not confuse with the interpretation of impulse noise as an almost instantaneous sharp sound (e.g, clicks and pops) appearing on audio equipment, usually caused by electromagnetic interferences or electric/mechanic disturbances.

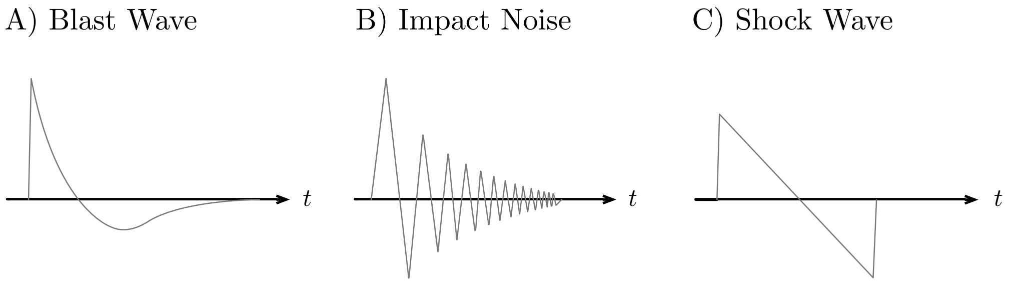


Figure 1.2: Schematic representation of three basic impulse noise pressure-time profiles.

collisions etc. The range of parameters that define an impulse is large. Impulse durations may vary from tens of microseconds to several hundred milliseconds and intensities (Sound Pressure Level SPL) may vary from less than 100dB to in excess of 194dB peak SPL (one atmosphere). The energy of an impulse is most often broadly distributed, but it is possible to find spectral concentrations of energy at various frequencies for some impulses [HH86]. A common characteristic of this sounds is that they are highly dependent upon the environment in which they propagate, i.e., their transmission path.

In more proper terms we should make a distinction between different types of impulse noise transients [HH91], some of the more prevalent being:

- **Blast wave**: a noise transient produced by a sudden change in pressure as the result of a rapid energy release (most often chemical or electrical). The most representatives sources of such waves are explosive discharges. The physical characteristics of these impulses are largely dependent upon the geometry and scale of the source, i.e., the shape and size of the explosive charge, etc.

- **Impact noise**: a noise transient that follows the impact between two objects as the result of a rapid release of energy through primarily mechanical mechanisms, e.g., a hammer strike or a snare drum hit. The physical characteristics of these impulses are largely dependent upon the mechanical properties of the impacting objects.

- **Shock wave**: a noise transient produced by an objet (or disturbance) moving trough a fluid faster than the local speed of sound, e.g., a jet fighter breaking the sound barrier, a whip crack, etc. Shock waves are characterized by a positive overpressure maximum, followed by a corresponding under-pressure minimum. The physical characteristics of these impulses are mostly dependent upon the size of the object and its speed.

Figure 1.2 shows an schematic representation of the pressure-time profile of different types of impulse noise. Please note that for the remaining of this document we will use use the terms impulsive sound source and impulse noise interchangeably for any sound falling within the described categories.

Impulse noise has been a subject of study for a long time [WK74a, Aka78], specially in terms of noise control and audiological effects prevention [HH86]. Large blast waves produced by cannon

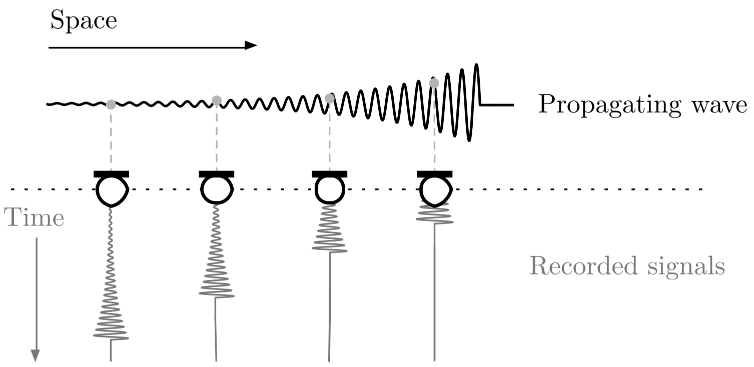


Figure 1.3: Schematic representation of spatial sampling on a microphone array.

fire and explosives have been characterized and modeled [FTCP93], and the same goes for shock waves produced by supersonic ammunition [Whi52] and airplane's sonic booms [ML65].

On a real scenario the recorded waveform may be very different from that described by an ideal model [Emb96] . In close range recordings, ground reflections and turbulences are most likely to be overlapped with the direct signal. While, in long range recordings, the influence of the propagation path has a huge impact on the received sound [MS08]. Additionally, nonidealities on the recording equipment can also produce some artifacts, the most relevant being signal saturation and low pass filtering, either at the microphone or at the analog front-end. Saturation is very likely to occur given the high sound pressure levels created by some impulses that can easily exceed the threshold of pain of the human ear.

1.3 Multichannel Sound Processing

Over the last decade multichannel sound processing has gained widespread usage. Nowadays a group of microphones working in tandem can be found in many devices, e.g. most smartphones and laptop computers are equipped with two (sometimes more) microphones in order to perform speech enhancement. In case an audio processing system has more than one microphone we can then speak of a microphone array.

1.3.1 Microphone Arrays

In general, a sensor array can be considered a sampled version of a continuous aperture [BCH08]. In acoustics, an aperture is a spatial region that behaves as an electroacoustic transducer transforming acoustic waves into electrical signals (microphone), or vice-versa (loudspeaker). The most direct implication of this, is that whenever working with a sensor array, signals are sampled both in time and space simultaneously. Spatial sampling is closely related to discrete-time sampling, since the continuos propagation of a signal though space is sampled at discrete points (the microphone locations). Figure 1.3 shows a schematic representation of a sound wave being sampled by four microphones at distinct locations and their recorded signals.

The basic problems that can be solved using multichannel (or array) processing techniques are:

- *Enhancing the signal to noise ratio SNR of a signal of interest*: e.g., 'beamforming', which consists in using a microphone array to form a spatial filter which can extract a signal impinging from a specific direction while reducing the contamination of signals coming from other directions. This process is analogous to digital 'temporal' filtering, only instead of combining samples of different time instants, a spatial filter combines samples taken at different spatial points.
- *Determining number and locations of emitting sources*: Microphone arrays can be used to extract spatial information from the recorded signals. The most common approach is to calculate the time it takes for a certain sound to travel between microphones and use that information to find the location of the emitting source.
- *Tracking moving sources*: Tracking builds upon the previous point although it requires additional techniques that also take in consideration the temporal evolution of the signals.

Acoustic array processing has already been applied to multitude of real-world applications including: Ultrasonic medical imaging, speech enhancement, advanced noise control, SONAR imaging, virtual 3D sound, sniper detection, etc.

1.3.2 Wireless Acoustic Sensor Networks

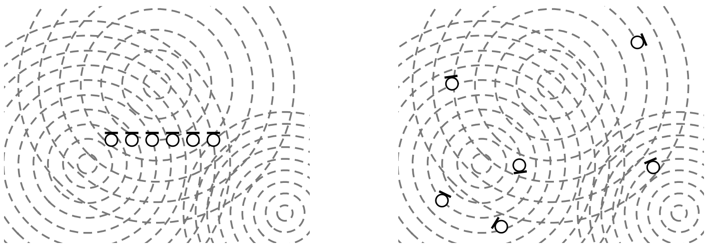
Despite their obvious advantages of multi-microphone systems over a single microphone setup, sometimes a microphone array is not good enough for the task at hand. The next-generation technology for audio applications are Wireless Acoustic Sensor Networks (WASNs) [Ber11], sometimes called Wireless Microphone Arrays (WMAs). The main advantage of WASNs over traditional microphone arrays is their ability to easily cover larger areas which translates into greater spatial diversity.

A traditional 'wired' microphone array is most often intended to sample a sound field[4] locally, mainly due to the inefficiency of a large setup, both in terms of size and number of microphones. This means that sometimes the microphone array is far away from the target source(s) resulting in a low Signal to Noise Ratio (SNR). On the contrary, the dynamic nature of WASNs allows the wireless microphones (nodes) to be placed at almost any position without major concerns, which results in a wider coverage of the sound field, therefore, an increased chance of getting a high SNR. Figure 1.4 illustrates this situation with a schematic representation of two microphone array geometries in a sound field.

Together with their advantages, due the decreasing price of many of the components needed for implementing WASN nodes (in part thanks to the surge of interest in portable electronics), WASNs are believed to be gaining widespread usage over the next decade. Some of the more prevalent applications of WASNs include:

- **Acoustic monitoring**: The most prevalent applications of WASNs in the literature are related to surveillance, e.g., gunshot detection, shooter localization, acoustic vehicle detection, etc. Using microphones to monitor an environment is cheaper than other common solutions such as cameras, and they are not constrained by line of sight. WASNs are well suited for this

[4] Sound field can be defined as the dispersion of sound energy within a region in a material medium in which sound waves are being propagated



ordered distribution random distribution

Figure 1.4: Schematic representation of two different microphone array distributions in a sound field.

task since they allow to quickly and easily cover a large area, e.g., the perimeter of a critical structure or building.

- **Acoustic tracking**: Many of the same techniques used for acoustic monitoring can be applied to source tracking. Notable applications in this field include indoor positioning systems and vehicle tracking. In those situations in which the use of traditional positioning systems such as GPS are not feasible, acoustic tracking is seen as a viable alternative. Vehicle tracking based on WASNs is still in its infancy but shows a great potential for some applications that are not succeeding using traditional methods (Radar or cameras), such as small unmanned aircraft tracking.

- **Speech/Signal enhancement**: Improving the quality of human speech is a critical part of hands-free telephony, teleconferencing, hearing aids, smart assistants, and many other applications. In most cases the number of microphones (and the distance between them) is constrained by the size of the device in charge of the processing. This problem can be bypassed resorting to a WASN by making use of external microphones. Since wireless devices loaded with microphones and wireless connectivity are getting smaller and cheaper, this is expected to evolve towards heterogeneous WASNs with many nodes of different kinds, e.g., worn by the user or strategically placed around a room.

In brief, WASNs can easily cover a larger area, can be quickly deployed, and more often that not, they are less expensive than a centralized solution of comparable size. However WASNs do have a long list of challenges to overcome [RM04] when compared to traditional microphone arrays. Some of the most important considerations when working with WASNs are:

- **Array geometry**: For most array-based algorithms, precise knowledge about the location of the microphones is a basic requirement, thus locating the nodes becomes one of the bigger concerns. Since some algorithms (e.g. beamforming or cross-correlation-based localization), rely on sample-to-sample differences, any error in the microphone location estimates effectually transforms into a temporal bias that has a negative impact on the performance. On such applications, common positioning solutions such as GPS are not precise enough and WASNs have to rely on 'self-localization', or the ability of the nodes to locate one another collaboratively. In some cases it is possible (and usually preferred) to use "blind" algorithms that not require *a priori* knowledge of the node locations.

- **Synchronization**: Each node in the network is an independent device with its own clock source. This has two major implications: the nodes need to be synchronized in order to establish a common timebase within the network (clock offset correction), and each node has a slightly different sampling frequency (clock skew). Failure to do so will result in decreased performance for all algorithms based on temporal differences between microphones (e.g. source localization), and it can also affect communication protocols. Synchronization is specially complicated in networks composed of heterogenous nodes with hardware made by different manufacturers.

- **Bandwidth**: To perform traditional array-based processing on a WASN, the signals captured by every microphone must be shared with the network. This posses a main challenge, specially for large networks, since the available bandwidth is limited and transmitting multiple audio signals is data-intensive. Usually, data compression is employed to minimize data transmissions at the cost of higher computing power. Another problem that arises from wireless communications are packet losses and reception errors, that affect network throughput and should be taken into account by the algorithms.

- **Topology**: Topology selection and efficient routing are critical for data-intensive WASNs. For some applications it is advisable to have a master node, or Fusion Center (FC) to take care of the most compute-intensive tasks. In this situation, 'star' and 'tree' topologies are recommended depending on the physical size of the network. When most of the processing can be done locally at each node, 'mesh' topologies are favored, either fully connected or partially connected depending again on the distance between nodes. It is important to notice that long distance transmission requires more power, specially at high data-rates. Figure 1.5 shows three of the most common WASN topologies.

- **Processing**: Traditional microphone arrays rely on centralized processing. This is not mandatory in WASNs since each node has its own processor. Performing centralized processing when a large number of signals need to be processed/transmitted simultaneously could prove problematic, therefore, in case the algorithms and the network topology allows it, distributed processing is highly encouraged to lessen the work load. Another factor that must be taken into account is the minimization of the input-output delay, critical for real-time applications.

- **Scalability**: Since the number of nodes of a WASN is not fixed, the algorithms should be able to adapt to different number of microphones. On a scalable network adding a new node should not have a significative impact on the computational load or data traffic at the nodes that are not directly connected to the new node.

- **Power Consumption**: Many array-based algorithms require lengthy correlations, matrix inversions and/or other compute-intensive calculations. Computing power is directly related to power consumption which in turn reflects in size, price and battery life (whenever relevant). Another major factor for power consumption is transmission power, ideally it should be optimized based on the network size. Strategies such as the implementation of or low-activity mode are also encouraged for battery powered nodes.

- **Subset selection**: In some instances, such as when working with very large networks or in the presence of faulty nodes or reception errors, there should be some mechanisms to select a certain subset of microphones to perform the task at hand. In large networks where the nodes

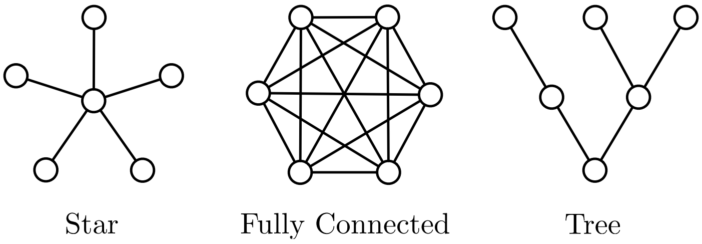


Figure 1.5: Schematic representation of three common wireless network topologies.

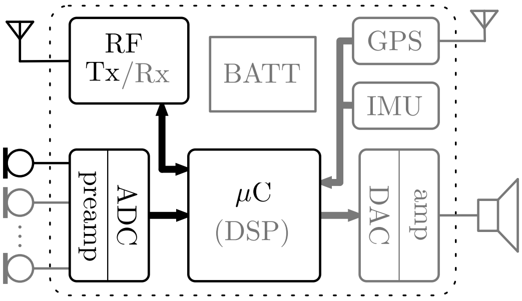


Figure 1.6: Simplified diagram of a typical wireless acoustic node.

are far apart, some of them might have a low SNR resulting in a decrement in terms of both performance and computing efficiency. The same goes for any kind of spurious information. If some data is known to be unreliable with a high probability, it should be discarded.

The most basic configuration of WASN node is composed of: at least one microphone and its pertinent analog front-end, an Analog to Digital Converter (ADC), a microcontroller(μC) and a Radio Frequency (RF) transmitter (Tx). Although, the hardware configuration of the nodes is ultimately subjected to the application. A more typical node configuration improves upon the basic setup by having a Digital Signal Processor (DSP) that allows for more complex algorithms to be implemented, an RF transceiver (TRx), capable of both transmitting and receiving data for inter-node communications, and some elements to perform self-localization; either a GPS module for long distance low-precision localization, or a Digital to Analog Converter (DAC) and a speaker for acoustic-based short-distance high precision localization. Other than the mentioned components, it is common to find general purpose nodes equipped with multiple RF-TRx on different frequency bands that allow for independent data and control channels, a temperature sensor for estimating the speed of sound, or an Inertial Measurement Unit (IMU) for obtaining the node orientation, among others. Figure 1.6 shows a simplified diagram of a typical wireless acoustic node, where the elements in black are basic components, while those in grey are optional components. It is worth noticing that current smartphones include most of the above mentioned components, thus there have been numerous efforts to implement smartphone-based WASNs. However due to the lack of optimization for real-time applications of such devices (specially in terms of low-level hardware control) and the heterogeneous nature of such a network (different hardware specifications at each node) there is a clear preference for *ad hoc* hardware in the literature.

1.4 State of the Art

The following state of the art review divides previous work relevant to this thesis into two discrete, although not necessarily independent categories:

- **Localization**: Algorithms that deal with source localization and node self-localization, either as separate problems or as one self-contained problem. Typically algorithms based on inter-microphone time delay estimation using cross-correlation both for traditional microphone arrays and WASNs.
- **Detection/Classification**: Algorithms most often based on machine learning for tackling acoustic event detection, classification or both simultaneously. Most of the examples in the literature are intended for single-channel systems, nevertheless multi-channel solutions based on data fusion can also be found in the literature to a lesser extent.

1.4.1 Localization

The are multitude of examples of acoustic source localization in the literature; ranging from those that focus on locating a single source to those that simultaneously find the location of a number of sources and microphones. All these algorithms have in common their use of data-intensive calculations that may not be suitable for their implementation on a resource restricted WASN. Localization algorithms are typically based on relative spatiotemporal measurements between microphone/node pairs that are combined to infer the desired spatial information [PAK+05]. Such measurements are taken either from acoustic, infrared, RF or magnetic signals. On wireless networks these measurements need first to be exchanged among the nodes or transmitted to the FC. The most commonly used measurements are Time of Arrival (ToA), angle or Direction of Arrival (DoA) and received energy or Received Signal Strength (RSS). However, their use is not straightforward because of the random component introduced by time-varying errors (e.g., additive noise and interferences) and environment-dependent errors (wall reflections, furniture obstructions, etc.). Wireless and acoustic RSS are seen as poor estimators of range due to these problems and methods relying on it usually lack the precision required for audio localization applications.

Acoustic source localization using the Time Difference of Arrival (TDoA) of an acoustic event between a group of microphones is an old topic [SR87]. This approach involves the use of cross-correlations for the estimation of the time delays [AH84], usually with Phase transform (PHAT) weighting due to its robustness to reverberation[BS97]. A technique known as Steered Response Power (SRP), typically in the form of SRP-PHAT, is very popular for source localization on indoor environments using traditional microphone arrays [DSY07] [CML11]. However, PHAT weighting does not perform equally well with all kinds of sound sources. On [MSK+09] the overall localization scheme is based on SRP-PHAT, although they claim that its performance decreases when used to locate impulsive sources due to their short time duration. They use the activation duration of an energy-based event detector to discern between impulsive and non-impulsive sources. For impulsive events they use a modified localization algorithm that pre-align the signals and uses a smaller window. Some sources take a different approach to locate impulsive sources. In [SCK14] gunshot localization with a linear array in an anechoic chamber is done via time-domain beamforming by finding the angle in which the sum of the signals is maximized according to two different metrics. A different strategy is proposed In [Par08], the use of time reversal for impulsive source localization with a low number of receivers by analyzing the reflections. Using a ray propagation model they are able to locate a source in a known geometry simulated space containing rigid bodies that blocked the direct path to some of the microphones.

Most examples in the literature tackle indoor localization where the main problem is the reverberation, however, outdoor scenarios have a different set of complications. The effect of wind and temperature variations is explored in [FCL02] where the localization of far-field impulsive sound sources is done by triangulation using data recorded during field experiments for a variety of impulsive sound sources: artillery guns, mortars, and grenades. Outdoor localization usually involves larger distances between microphones in order to cover larger areas, thus it is convenient to use a WASN.

Those algorithms intended to work on WASNs have to overcome the particular challenges associated with them, such as unknown node locations, clock offset and skew between nodes, limited bandwidth and scallability [JW00]. Synchronization is a big concern for WASN processing, some techniques assume that thigh clock synchronization between the nodes is achieved via networking messages [EGE02] while on other techniques the estimation of the clock offset between nodes is part of the processing. In [CAST13] source localization on an asynchronous network is proposed using nodes equipped with at least two microphones. The proposed method is based on the minimization of a global cost function that combines individual hyperbolic constraints associated to the TDoA measurements on microphone sub-arrays at local nodes. Since cross-correlation and beamforming require the transmission of sound signals between nodes, the available bandwidth can pose a problem on WASN. Data transmission constraints have also been explored. In [SH05] an energy-based Maximum Likelihood (ML) 2D localization scheme for multiple sources is proposed. The model based on acoustic RSS assumes omni-directional sources and does not account for the potential effects of the propagation path, thus its performance in real scenarios may be affected.

A specialized type of outdoor sound localization system mod often based on WASNs are sniper detection systems. These systems commonly work by detecting the shockwave created by supersonic projectiles [SPS98]. The characteristic time-pressure profile of the shockwaves is related to the physical properties of the projectile [Dan06] allowing bullet trajectory, speed and caliber estimation, while the shooter localization is typically obtained using the muzzle blast. The work on [SHV$^+$11] uses soldier wearable nodes equipped with 4 microphones. They perform a simple detection via a finite state machine and temporal measurements of the shock wave. In addition to shooter localization and trajectory estimation they perform weapon classification by matching the speed profile of the projectile with a series of known templates. Other similar approaches include: [MP10] based on two synchronized subarrays with 4 microphones and [DKW10], that exploits the time difference between the shock wave and the muzzle blast to locate the source on a WASN without synchronicity.

The most active area of research related to WASNs is the localization of the nodes themselves known as self-localization. This is not a new problem either [NS96], a popular method to calibrate traditional microphone arrays is to use a number of acoustic sources at known positions [SSP02].

The initial proposition of such techniques did not have to deal with the lack of synchronicity between microphones. However, more recent approaches adapted to the constraints of WASNs can be found, such as [HMdC+16] that proposes a method using controlled sources surrounding the nodes, where the emitted signals and source positions are known a priori. The proposed technique also tackles typical practical issues such as reverberation, unknown speed of sound, line-of-sight obstruction, clock skew.

In general, self-localization methods can be categorized as active methods when they require sound emission, and passive when they rely on environmental sounds. Passive methods are generally more complex and slower (they need to collect enough measurements), although the sounds emitted by active methods can be an annoyance to the end user. Self-localization in WASN has been solved in many different ways, in [PJHUF16] they differentiate between methods based on single microphone nodes and methods using sub-arrays and in [CAA+17] categorization is made depending on the type of measurement employed.

When more than one microphone is available at each node, self-localization can be performed by finding the location and orientation of each sub-array using Direction of Arrival (DoA) measurements. However, DoA-based algorithms can only find the relative geometry, and additional information is required to scale it. Node orientations can be obtained using an electronic compass, a device composed of a magnetometer and an accelerometer. Unfortunately, magnetometer measurements are sensitive to disturbances from electric equipment (and even large metallic objects) and must be frequently calibrated to avoid large errors [LLW09], thus, most algorithms tackle node orientation as an additional unknown for improved accuracy. Some examples include: [JSHU12] where 4 arrays with 2 microphones are located assuming that the nodes are located along the walls of a room of known dimensions, [PF14] where 3 synchronized sub-arrays with 5 microphones embedded on a table are calibrated, and [AHM+14], where a network of nodes equipped with 3 microphones are located using RF RSS measurements to solve the scaling.

Single microphone methods based on ToA are more prevalent. Early passive methods assume internode synchronization. Notably in [Thr06], a method known as affine structure from sound proposes a solution that approximates self-localization under a far field approximation defined in the calculus of affine geometry, and that relies on Singular Value Decomposition (SVD) to recover the affine structure of the problem. It estimates the locations of both nodes and sources in 2D space and the emission time of the sources. [CDBBM12] breaks the far-field approximation and proposes a closed-form solution by finding an approximate solution of a ML problem by transforming the original nonlinear Least Squares (LS) cost function minimization into a two step procedure Closed-form source localization techniques are attracting due to computational efficiency, but involve linearization of quadratic equations negatively affecting the precision. In [PPH14] a group of smartphones lying on a table are located using the voice from the speakers seated at the table and relying on data-association for relating the ToA at different sensors. Here the closed-form expression is used to initialize an iterative refinement. These type of passive techniques can be adapted to an asynchronous network by using an active approach. In [CPSB+16] syncronization is estimated via pairwise time differences using cross-correlations called “BeepBeep”, a strategy proposed in [PSZ12].

There are also a number of methods that take self-localization, source localization and synchronization as a single minimization problem, although, they are harder to solve and more sensible to measurement errors due to the increased number of unknowns. The work on [GKH13] proposes a method based around low-rank-approximation using SVD for the minimization, that requires a minimum of five microphones and 13 sound source events. Similarly, [WHRC16] proposes a method based on a Gauss-Newton low-rank approximation algorithm aided by a multi-initialization scheme to avoid local minima. Also, [PP17] proposes a self-localization method for user-worn nodes. A user here acts as a sound source and the node positions are recovered using Multidimensional Scaling (MDS). Their method requires node tracking and prediction of spatial information during periods of node silence.

Generally speaking, the implementation of acoustic localization methods on WASNs require high bandwidth for the transmission of the signals, and significative computational power for the estimation process. These requirements are more relevant when the processing has to be performed in real-time or close to real-time (e.g., continuos source localization), but can be relaxed when throughput is not a priority (e.g., 'one-off' self-localization when the nodes are first deployed).

1.4.2 Detection and Classification

Numerous examples of pattern recognition techniques applied to the detection of gunshots and other impulsive events can be found in the literature over the last decade, however, little work has been done on the classification of similar impulsive events. Classification poses a harder problem, specially when applied within a same "family" of sounds, since the captured signals are strongly related to the recording environment and the positioning of the sources and receivers. Moreover, multi-channel classification has not yet been widely studied with only a few precedents available. Unlike localization algorithms, machine-learning based detection and classification are not mandatorily constrained by a lack of knowledge about node locations or the need for tight synchronization. These methods usually work with blocks of data or frames, the duration of which can go from a few milliseconds to around one second and the information carried by the sound waves is condensed into a series of parameters, thus, the requirements to implement such algorithms on a WASN are relaxed.

Most reported work on audio-based machine learning is related to speech and music, however there is a much more broad set of problems that are of interest. Recently, the term 'machine hearing' [Lyo10] has been used to refer to all the problems that can be solved using computers and sound. The most represented problem that deals with 'noise' signals is Acoustic Event Detection (AED) that aims at processing a continuous acoustic signal to attach symbolic descriptions to the sound events present on it. An early example that compares various methods developed for Automatic Speech Recognition (ASR) and musical instrument recognition for their application to AED can be found in [CS03]. Over the last decade, AED starts appearing in the literature as a trend parting from application specific detection/classification, aiming at more generalist goal by introducing a (relatively) high number of classes describing different types of acoustic events. Some examples based on different sets of standard features and various classifiers include [TN09] and [ZZHJH10]. A comparison of different features for event detection can be found on [CK14, KLP+13].

The most recent AED systems are based on deep neural networks inherited from computer vision. These systems, such as [CHHV15] usually implement one ore more convolutive layers that intended to extract high level features from an input spectrogram, followed by a recurrent neural network intended to learn the temporal evolution of the signals. These kind of systems are commonly used for polyphonic detection, that is, when more than one class is present at the same time. Some work is also being done in multichannel systems, such as [APP+17] where a stereo signals is feed to the network together with a series of spatial features extracted from the inter-microphone differences. Unfortunately, the type of deep neural networks that are being applied to AED commonly require hundreds of thousands of operstions per input sample and are executed on massively parallel processing systems.

Interest on audio surveillance systems predates the modern interpretation of AED. Audio surveillance involves the detection of hazardous acoustic events such as gunshots and explosions and other indicators such as the detection of distress in the human voice, most often in the form of screams. A compilation of audio surveillance contributions can be found in [CCTM16]. These kind of acoustic events (specially the impulsive ones) are sometimes called “rare” events which are both abnormal and random in nature. From a detection point of view, abnormal implies that a disproportionate amount of time is spent monitoring noise, while random implies that continuous sensing is needed since predicting event arrival times is not feasible. These factors are accounted for in [CRJC+11] that evaluates various low-power gunshot detection schemes intended for continuous use preceding a more complex classification stage.

Creating an acoustic event database is a laborious task, furthermore, recording certain types of impulsive events, such as gunshots and explosions, is dangerous, tedious and sometimes technically difficult. Therefore, most reported work rely on examples obtained from sound libraries (usually processed samples that do not reflect “real” recordings) or a small number of recordings taken in a safe environment (i.e., gun range) that typically do not reflect the diversity created by different propagation paths. An early example of impulsive event detection is presented in [DBAP99] where the database is composed of recordings of glass breaking,door slams and explosions extracted from a sound library. Most proposals use Mel Frequency Cepstral Coefficients (MFCC) as part of their feature vector, and classic classification algorithms such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) or Support Vector Machines (SVM). Some examples include:[GVT+07] where gunshot and scream detection is performed on audio recordings taken from movies soundtracks and internet repositories mixed with ambient noise. The proposed method is based on MFCC and other standard features with a GMM classifier. In [FAJ10] correlation against templates is proposed as features together with MFCCs. The system uses a HMM classifier for gunshot detection on real recordings taken at an army training session against speech balloon explosions and handclaps recorded in a laboratory environment. Also using real recordings, in [AUM13] gunshots are detected using a linear kernel SVM overMFCCs and other standard features in the presence of other impulsive noises.

A common trend in acoustic surveillance is the implementation of a multi-stage classifier. The work on [NPF09] proposes a two stage approach: an abnormal event detector followed by a classification into gunshot or scream using HMM and MFCC. A similar two staged approach for the detection of screams and cries in an urban environment using standalone smartphones (without networking) can be found in [SK16]. In [MSVG11] another two step process is presented, first distinguishing between impulsive a non impulsive sounds with a Bayesian classifier and some proposed energy-based features together with MFCCs. The same impulsive non-impulsive pre-classification can be found in [CRHK12], using multiple standard features and a GMM. It is also possible to find some acoustic surveillance examples that depart from the use of MFCC such as [GH15], where Gabor features are reported to outperform MFCCs using GMM classification on gunshots, explosion, glass breaking and screams. Similarly, [VA12] proposes the use of gamma tone filters with a SVM for the classification of various events including gunshots and screams extracted from sound libraries.

From these examples it is clear that gunshot detection is a priority for audio surveillance systems not only due to its threatening nature but also for its usability as part of gunshot localization systems. The natural next step in the evolution of gunshot detection is to recognize the type of weapon that was fired. An early proposal can be found in [CER05], where a hierarchical approach to weapon classification, based on MFCC and other standard features with a GMM classifier is used to distinguish between pistol, rifle, submachine gun, grenade and cannon fire sounds obtained from a sound database and mixed with background audio. More recent proposals include [PR12] where a set of specialized features that are compared to MFCCs in the classification of gunshots recordings of 10 weapons obtained from a sound library, and [DT13] that presents a a pre-emphasis filter preceding the extraction of standard features and followed by GMM Hierarchical Classification intended for distinguishing between two types of rifle using signals recorded at a gun range. Note that none of these works provide information about the distance between the microphone and the guns.

Most of the literature on audio-based machine learning is based on a single microphone setup, however the spatial diversity of a multi-channel system can be exploited to increase the accuracy of the system. In order to do so, the information captured by each microphone has to be combined either as signals (usually via beamforming), features or decisions. The most popular choice is to combine the decisions taken by individual classifiers, using a technique known as decision fusion, specially on WASNs since it requires less data transmission than beamforming and feature fusion and does not require tight synchronization. The most basic form of decision fusion is majority voting [LS97] where the winner class is the most represented among available decisions, although there are many alternatives that assign a weight to each decision. A distance-based decision fusion scheme is proposed in [DH04] for vehicle classification with a distributed array. The system uses spectral energy band features and ML classification. Node-source distance is estimated from the received SNR thus the distance fusion scheme does not provide a significative benefit when compared to other fusion methods. In [ZML10], a WASN performs multi-class classification using SVMs and decision fusion with a nearest neighbor weighting scheme, where the decision of the node closest to the source is selected as the final decision and is shown to perform better than majority vote. The work in [GPKM14] compares signal fusion (delay-and-sum beamformer) and decision fusion using standard features and GMM and concludes that weighted average fusion works the best among the tested techniques. Also, [PMH$^+$15] presents the positive impact on accuracy of decision fusion for multi-class classification in a system with just 4 microphones. Examples of other fusion techniques can also be found: in [AEQGC06] a traditional microphone array is used to perform fusion via beamforming on a simulated indoor environment as part of system that classifies an audio segment first into speech or non-speech and then into various classes (including impulsive noises). Regarding feature fusion: [SNM13] presents a graphical model-based feature fusion framework to identify the launch/impact of rockets and mortars using a traditional array. Also in [GNM15], a WASN is used to classify sound as speech music and noise, using feature fusion with modified MFCCs and Linear Discriminant Analysis (LDA) on simulated rooms with various reverberation coefficients. The feasibility of implementing a classification system on a WASNs is studied in [SH15] where the accuracy and execution time of different sound classification tasks using WASNs is compared.

The potential of WASNs for combined detection, location and classification of targets [LWHS02] is of great interest for many security related applications. An early approach to WASN acoustic surveillance is presented in [DGA$^+$05] where a network of ad-hoc multimodal (acoustic, Infrared and magnetic sensors) nodes detects and classifies civilians, soldiers and vehicles using statistical analysis with a focus on scalability (up to thousands of nodes) and long battery life. [SSSS13] tackles acoustic detection, tracking and classification of low flying aircrafts using a centralized WASN where each node is equipped with five microphone clusters (8 microphone subarrays) adding up to 64 microphones per node. Detection and classification are based on the spectral signature of the aircrafts, 3D localization is possible by triangulation with 2 or more nodes. In [MNH08] a network of sensors located along a road to classify vehicles based on their spectral signature using a distributed implementation of K-Nearest Neighbors (K-NN). The first node that detects a vehicle (adaptive threshold) triggers a process that involves the selection of a subset of microphones and the distributed computation of the classification using simulated signals. Precedents of WASN surveillance based on impulsive events can also be found in the literature: [ASJS07] proposes joint source localization and signal estimation based on ML estimation for gunshots, RPG, and artillery fire in a simulated environment formed by 9 sub-arrays equipped with 8 microphones . It also proposes the use of statistics obtained from wavelets showing a high degree of separability in a 2D feature space. The work in [KLC14] also tackles joint detection and localization of selected acoustic events in an acoustic field for smart surveillance applications.

1.5 Scope of the Thesis

The main objective of this thesis is the research and development of novel multichannel algorithms for impulsive sound source analysis, including detection, classification, localization and self-localization methods. The aim is to develop efficient techniques that can be implemented on a WASN in which other methods cannot be used due to a series of constraints, such as restricted computational loads, or limited wireless bandwidth. In general terms, this proposal aims at developing a framework for collaborative processing in which the spatial diversity provided by a group of microphones is exploited to solve each of the presented problems simultaneously. At the same time, the focus is to improve multichannel acoustic analysis of impulsive events by taking advantage of the information obtained from the resolution of some of the problems to increase the performance of later stages. This philosophy is in clear contrast with off-the-shelf approaches where each problem is addressed independently on a centralized structure.

The particular problems arising from the main objective addressed in this thesis are:

- To develop efficient self-localization algorithms for the simultaneous localization and synchronization of the nodes that allow their implementation in a resource restricted WASN.
- To improve impulsive acoustic event detection by merging the information obtained from a network of distributed sensors.

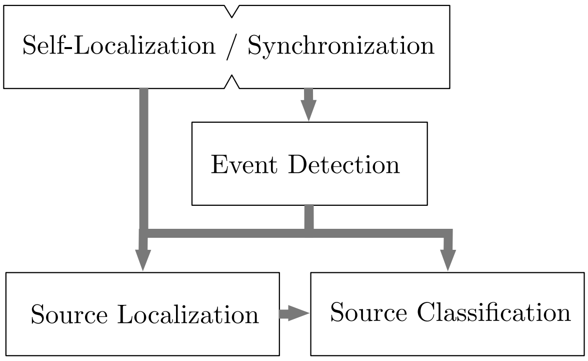


Figure 1.7: Simplified flow diagram of the proposed algorithms.

- To formulate new solutions to the problem of acoustic source localization by exploiting the information obtained from event detection.
- To minimize the cost of the localization process both in terms of computational load and data transmission.
- To design an efficient classification system that exploits the spatial diversity provided by the network.
- To improve impulsive source classification by taking advantage of the spatial information obtained from source localization.

The hierarchical relationship between the presented problem in shown in Figure 1.7 as a flow diagram.

1.6 Structure of the Thesis

This thesis is divided in two main blocks organized as follows:

- The first block contains a preliminary study of the studied problem. It is divided into two chapters:

  • Chapter One (the current one), contains an introduction to the problem in the form of a brief description of the elements that play a role on it, followed by a comprehensive review of the state of the art of the different areas of interest addressed in this thesis, and finally a description of the main objectives of this work.

  • Chapter Two briefly introduces the theoretical basis needed for easily understanding the problems tackled in this thesis, and offers a brief description of some well-known algorithms that are to be used for the solution of said problems.

- The second block, which is the main block of the document, contains a detailed description of the work done to accomplish the objectives of this thesis, and a comprehensive description of the experimental work and obtained results. The chapters of this block correspond to each of the problems tackled in this thesis as follows:

- Chapter Three addressed the self-localization problem on a WASN. A novel closed-form self localization algorithm based on pairwise DoA and range estimates is proposed in this chapter. Additionally, network synchronicity can be achieved as a result of the method employed to obtain the range between a pair of nodes using acoustic signals.

- Chapter Four tackles the problem of impulsive sound event detection on a spatially diverse scenario. A novel classification algorithm that combines clustering and supervised training is proposed. The algorithm is shown to obtain a comparable performance to more complex algorithms, making it a suitable candidate for implementation in resource restricted devices.

- Chapter Five deals with sound source localization. Two novel schemes for performing source localization on a WASN are proposed. The first one takes advantage of a network with acoustic event detection capabilities to reduce the length of the signals used in classic localization algorithms. The second approach bypasses the need for cross-correlations by working with ToA estimates and is capable of obtaining better results while minimizing the computational load.

- Chapter Six tackles impulsive sound source classification on a spatially diverse scenario. First the problem of gunshot classification on a WASN with technical restrictions is studied. Second, a more generalist impulsive source classification system is proposed in a WASN tacking advantage of the capabilities described in the previous chapters. In both cases spatial information is used to improve the results of the classification via weighted decision fusion.

- Chapter Seven summarizes the most relevant results obtained by this research and lists the main contributions of the thesis. This chapter also contains a description of possible future research lines that have been opened or broaden by the work done in this thesis. Finally, a list of publications derived from this thesis is also included.

CHAPTER 2 Background

2.1 Introduction

This chapter intends to serve as an introduction to the theoretical basis needed to better understand the problems that will be explored later. First the main properties of sound propagation and the concept of spatial mixing model are introduced. Audio signal processing is commonly tackled in the time-frequency domain. Hence, the motivation for the use of such domain is presented, and the mathematical basis for the transformation of the sound signals into the time-frequency domain is described. Since most of the proposed techniques are based around some kind of mathematical optimization, the concept is briefly discussed. Once the most fundamental notions have been introduced, the discussion shifts towards the use of acoustic signals in spatial applications. The problem of time lag estimation and its use for sound source localization and self-localization is introduced. Finally, the last section of this chapter offers a small overview of machine learning for audio-based applications. Some concepts such as features, classification and performance metrics are described.

2.2 Sound Mixtures and Propagation

A common approach to model spatial audio is defining the signal recorded by a microphone as a mixture of sounds emitted by different sources that travel trough different propagation paths towards the receiver. This implies that sounds produced by different sources are (for the most part) independent to each other until they get combined into a single acoustic signal (i.e., mixed) by the receiver, thus a receiver at a different location will perceive a different mixture.

2.2.1 Mixing Model

Prior to the description of the mixing model, it is worth clarifying the notation used in this thesis. In signal processing, the most common convention assumes that variable (t) represents continuos time, while variable $[n]$ represents discrete time (i.e. $[n] = (n/f_s)$, where f_s is the sampling frequency). However, in this document, all the signals are defined in discrete time, thus (t) is adopted to represent discrete time signals.

Let us consider a set of M microphones that receive the signals emitted by N sources, $s_n(t)$, $n \in \{1, ..., N\}$, generating M mixtures, $x_m(t)$, $m \in \{1, ..., M\}$. The general expression for the additive mixing model is given by:

$$x_m(t) = \sum_{n=1}^{N} s_n(t) * h_{mn}(t) \ , \tag{2.1}$$

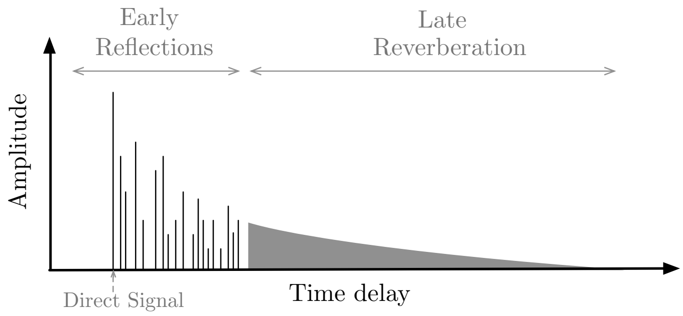


Figure 2.1: Schematic representation of an Impulse Response.

where $h_{mn}(t)$ is the impulse response of a Linear Time-Invariant (LTI) filter that describes the acoustic channel (i.e, the propagation path) between the n^{th} source and the m^{th} microphone, and operator $*$ represents linear convolution. The filter $h_{mn}(t)$ is commonly called acoustic impulse response, and its frequency domain transform is known as acoustic transfer function (ATF). The signals received by the microphones are then the result of the convolution between the original signal emitted by source $s_n(t)$ and the filter $h_mn(t)$ that depends on the environment, as well as the location of the source and the microphone inside said environment.

Mixing models can be expressed in matrix notation to simplify their formulation. Let us define $\mathbf{x} = [x_1(t), ..., x_M(t)]^T$ as an $M \times 1$ vector of mixtures and $\mathbf{s} = [s_1(t), ..., s_N(t)]^T$ as an $N \times 1$ vector of sources, where operator $(.)^T$ denotes matrix transposition.The mixing matrix can then be defined as:

$$\mathbf{A} = \begin{pmatrix} h_{11}(t) & \cdots & h_{1N}(t) \\ \vdots & \ddots & \vdots \\ h_{M1}(t) & \cdots & h_{MN}(t) \end{pmatrix}. \tag{2.2}$$

Accordingly, the general mixing model is given by $\mathbf{x} = \mathbf{A}\mathbf{s}$. Please notice that we are following the linear algebra convention by which vectors are represented in bold lower case and matrices in bold uppercase.

2.2.2 Acoustic Impulse Response

Impulse responses used to represent acoustic environments are composed of three parts: direct sound, early reflections, and late reverberations (See Figure 2.1 for a schematic representation). Direct sound represents the earliest arriving (typically strongest) sound wave. Early reflections describe the sound waves that arrive within the first tenths of milliseconds, when the density of reflected waves is low enough that the human ear is able to distinguish individual paths. These early reflections (and possibly diffractions) are what provide human listeners with most of the spatial information about an environment, because of their relatively high strengths, recognizable directionalities, and distinct arrival times. The late reverberation phase is composed of reflections with such a high density that the ear is no longer able to distinguish them independently. They typically convey information about the size of the environment and the materials of the reflective surfaces. During this phase the impulse response resembles an exponentially decaying noise function with overall low power.

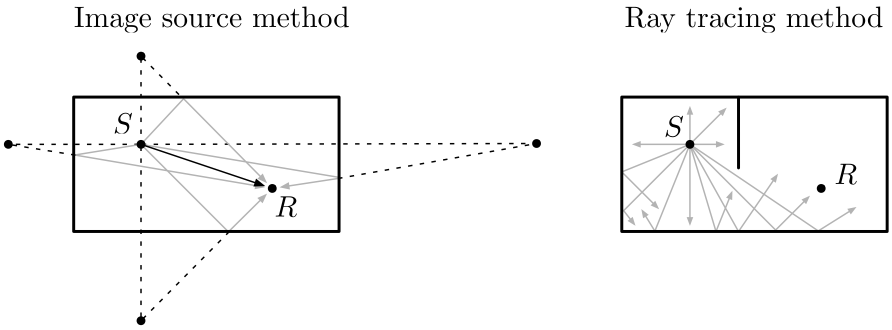


Figure 2.2: Schematic representation of two common methods for room impulse response modeling.

An impulse response can either be measured or modeled. Measurement techniques work based on the deconvolution of a known, and perfectly reproducible excitation signal [SEA02]. The first challenge of acoustic impulse response modeling is to find the significant propagation paths along which sound waves travel from source to receiver. Assuming that sound waves travel as rays that follow the shortest path when the propagation medium is homogeneous, the problem is reduced to finding piecewise-linear paths from source to receiver with vertices on the surfaces of obstacles and then obtaining the time lag and attenuation for said path [Rin00]. Two of the most common approaches to this problem (shown in Figure 2.2) are:

- *Image source method*: reflection paths are computed by considering virtual sources generated by mirroring the location of the audio source (S) over each surface of a polygonal environment. Reflection paths are computed up to any order by recursive generation of virtual sources. This method is particularly efficient in modeling rectangular shaped environments, due to rectilinear symmetries, thus complex geometries are commonly approximated by rectangular sections.
- *Ray tracing method*: reverberation paths are found by generating rays that emanate from the source (usually in random directions) and following them through the environment for a fixed amount of reflections or until they reach the receiver. A primary advantage of this method is the simplicity of the ray-surface intersection calculations, however important reverberation paths may be missed and since the results are dependent on particular positions, this method is not directly applicable when either the source or receiver is moving.

2.2.3 Sound Propagation

Generally speaking, generating accurate sound propagation models is a very challenging task [BGBBJ03] since acoustic waves are subjected to a wide variety of effects such as: attenuation, reflection, absorption, refraction, diffraction, diffusion, interference with other sound waves, etc. However, we have seen that a simple model for an acoustic impulse response can be obtained as a series of time lags with an associated amplitude. Consequently, we are just going to briefly describe the two main mechanisms required for this approximation: attenuation, and propagation speed.

Assuming that a particular sound source can be modeled as an acoustic point source, we can define a basic attenuation model based on the principle that, in a homogeneous medium, wave propagation from a point source is purely spherical. This is formally known as the standard *inverse square law* for point sources and can be expressed as:

$$\text{SPL} = \text{SPL}_0 - 10\log(4\pi r^2), \tag{2.3}$$

where r is the radius of the sphere (i.e., the distance to the source), and SPL_0 represents the original sound pressure of the sound source (in dB). However this model is too simplistic since it fails to capture the fact that sound energy absorption is frequency-dependent and it changes (most notably) with air humidity. A more advanced model will then consider absorption as a function of distance, frequency and humidity. In addition to this, it is common to include the directivity of the source as a parameter that scales the result as a function of the emission angle.

Another crucial factor, specially relevant for multichannel sound processing, is the speed at which sound propagates, that is directly related to the time it takes for the sound waves to travel from its source to a receiver. The speed of sound in air is highly dependent on temperature, higher temperatures producing higher sound speeds. The relationship between the temperature of air and the speed of sound can be expressed as:

$$c = c_0\sqrt{1 + \frac{\mathcal{T}}{273.15}}\ , \tag{2.4}$$

where $\mathcal{T}$ is the temperature in degrees Celsius (°C) and c_0 represents the sound speed at 0°C (331.5 m/s). Generally speaking, for each degree increase in temperature, the speed of sound increases by 0.61 m/sec.

2.2.4 Outdoor Sound Propagation

Sound propagation can be assumed to be time invariant in uniform mediums such as the interior of a room. However, propagation in outdoor environments is anything but uniform. Changing meteorological conditions can easily cause large fluctuations in sound levels over time periods of minutes. In general, the longer the transmission path, the larger the fluctuations. Outdoor sound propagation is affected by many mechanisms, including:

- Source geometry and type (e.g., point, directional, coherent, incoherent).
- Meteorological conditions (e.g., wind and temperature gradients, atmospheric turbulence).
- Atmospheric absorption of sound (e.g., air temperature and humidity).
- Terrain type and contour (e.g., ground absorption, reflections).
- Obstructions (e.g., buildings, vegetation, etc).

Some of these mechanisms, such as the effect of source geometry or surface absorption are also found in indoor propagation, thus, they are contemplated by some of the more complex impulse response modeling techniques. On the contrary, dynamic and/or non-linear effects more common on outdoor environments, such as air temperature gradients or wind speed gradients, are harder to model and most often ignored. Figure 2.3 shows a schematic representation of refraction caused

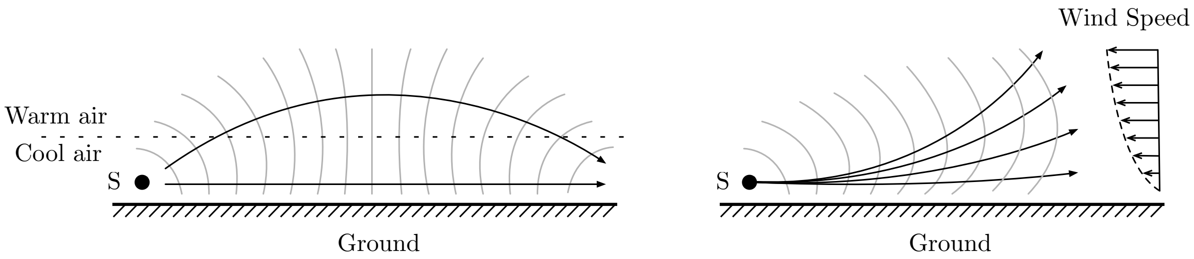


Figure 2.3: Schematic representation of refraction caused by air temperature gradients (left) and wind speed gradients (right).

by a change in the speed of sound due to air temperature and wind gradients. Generally speaking, these mechanisms have not been widely studied yet in terms of signal processing, although, they are very relevant for noise control. In the same way as impulsive sound source modeling, outdoor sound propagation has been studied for decades in order to better predict sound pressure levels for noise control applications. The existing models go from simple tables generated by empirical measurements to very complex numerical models based on fluid dynamics [Ras86].

2.3 Time-frequency representation of sound

The non-stationary nature of most sound sources (specially impulsive noises), motivates the analysis of sound in both the time domain and the frequency domain simultaneously. The frequency content of sound signals changes trough time, so it can only be considered stationary in short-time segments (typically tenths of ms). The classical Fourier series of a 'whole' signal obtains an accurate representation of the frequency content of said signal, but at the same time it fails to provide information about quick changes of frequency content or sudden changes in energy. This is commonly known as the *time-frequency resolution tradeoff*. The most commonly used time-frequency representations of sound is the Short-Time Fourier Transform (STFT) [AR77] which divides the input signal into segments, or frames, and performs the Fourier analysis of each frame independently. There are other widely used methods for time-frequency representation such as the Discrete Wavelet Transform (DWT) [AH01] or the Constant-Q Transform (CQT) [BP92] . However, all the algorithms described in this thesis are based on the STFT, which is described bellow.

2.3.1 Discrete Short-Time Fourier Transform

The discrete STFT is a time-localized spectral transformation based on the discrete Fourier transform (DFT). The DFT is an orthogonal transformation which turns a finite sequence of samples equally-spaced in time into a same-length sequence of complex sinusoids equally-spaced in frequency. The DFT of a discrete time signal $x(t)$ of L samples in length is calculated according to:

$$X(k) = \sum_{t=1}^{L} x(t) e^{-i\frac{2\pi}{L}kt},\ k = 1, ..., L\ , \tag{2.5}$$

where $X(k)$ are the DFT coefficients, k is the frequency index and i is the unit imaginary number. Note that coefficients $X(k)$ are complex values that represent the amplitude and the phase offset of each frequency component.

In the same way as the frequency components are obtained from the time signal, $x(t)$ can be obtained from $X(k)$ using the Inverse Discrete Fourier transform (IDFT) given by:

$$x(t) = \frac{1}{L}\sum_{k=1}^{L} X(k)e^{i\frac{2\pi}{L}kt},\ t = 1, ..., L\ , \tag{2.6}$$

The DFT is a frequency localized transformation, where the analog equivalent to the normalized frequency of the basis functions is dependent on the sampling frequency f_s and fixed according to $f_k = \frac{f_s(k-1)}{L}$. Audio signals are composed of real valued samples ($x(t) \in \mathbb{R}$), which causes the DFT to be symmetric. Due to this fact, only the first $K = \frac{L}{2} + 1$ frequency bands are considered since the upper components are a just mirrored and conjugated version of the lower components.

In order to calculate the STFT the input signal is split into equal-length segments by using a sliding window and then the DFT of each of the segments is computed and stored as a column in a matrix. This way the STFT can be viewed as a two-dimensional transformation which axes represent frequency and time. The segments (frames) are usually overlapped to avoid artifacts at the boundaries.

$$x_l(t') = w(t')x(t + lD),\ t' = 0, ..., L\ , \tag{2.7}$$

where $x_l(t')$ is the l^{th} frame of signal $x(t)$ windowed by function (wt'), (t') is the local time index, L is the window length, and D is the hop size: the number of samples that the sliding window advances between two consecutive frames. The STFT can then be expressed as:

$$X(k,l) = \sum_{k=1}^{L} w(t')x(t + lD)e^{-i\frac{2\pi}{L}kt'},\ k = 1, ..., L\ , \tag{2.8}$$

where $X(k,l)$ is a point of the STFT corresponding to the k^{th} frequency bin on the l^{th} frame.

The STFT is invertible, which implies that the original signal can be reconstructed by applying the inverse STFT (ISTFT). The process involves applying the IDFT to each frame and reversing the windowing process, commonly with a method called overlap-add. It is important to highlight that the choice of the window function is important to obtain a good reconstruction. Some widely used windows that take into account frequency resolution and sidelobe level (spectral leakage) are the Hanning and Hamming windows [Har78].

The time resolution and frequency resolution provided by the STFT are linked and both depend on the window length (L). A wide window gives better frequency resolution but poor time resolution, while a narrower window gives good time resolution but poor frequency resolution. The window length should then be matched to the particular characteristics of the sound signals being processed. Additionally, window length is restricted in real-time systems by the maximum allowed input-output delay. Figure 2.4 shows the well known frequency-time tradeoff of the STFT with the spectrogram of a logarithmic swept sine from $f = 0$ to $f = f_s/2$ obtained with three different windows lengths; from left to right it is possible to see the effect of an increasing window length. A spectrogram is a common method to visualize sound signals which represents the variation in en-

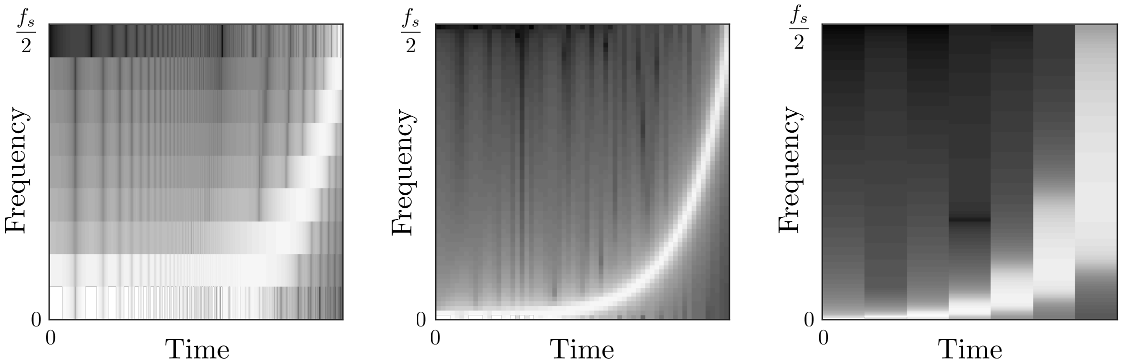


Figure 2.4: Spectrogram of a sine sweep from $f = 0$ to $f = f_s/2$. From left to right,increasing window length.

ergy of the spectral components with time. It can be computed as the squared magnitude of the STFT, usually in dBs with: SPG $= 20\log_{10}(|X(k,l)|^2)$, where operator $|.|$ represents absolute value, or magnitude.

2.4 Optimization

Optimization, or in more proper terms *mathematical optimization*, is a field dedicated to finding the best solution to a problem from some set of feasible alternatives. Optimization has applications in almost every engineering and scientific field. In this thesis it is required for both sound source localization and machine learning, which will be later discussed.

In the simplest case, an optimization problem consists in finding the value that minimizes a given function by choosing different values from within an allowed set and evaluating them. A general optimization problem can be represented in the following way:

Given a function $\mathcal{F} : \mathbb{A} \rightarrow \mathbb{R}$, for some subset $\mathbb{A}$ to the real numbers, find and an element x_0 in $\mathbb{A}$ such that $\mathcal{F}(x_0) \leq \mathcal{F}(x)$ for all x in $\mathbb{A}$.

Typically, $\mathbb{A}$ is a subset of the Euclidean space $\mathbb{R}^n$ specified by a set of constraints (i.e., equalities and/or inequalities that the members of $\mathbb{A}$ have to satisfy). $\mathbb{A}$ is often called the search space, while its elements are called candidate solutions or feasible solutions. The function $\mathcal{F}(x)$ is called by various names depending on the field, e.g., objective function, loss function, cost function or fitness function, among others. A feasible solution that finds the global minimum of the objective function is called an optimal solution. By convention, the standard form defines optimization as minimization problem, considering that a maximization problem can be treated as a minimization problem just by negating the objective function. Most often there is a single objective function which output is an scalar, however it is not uncommon to find problems involving more than one objective function to be optimized simultaneously (multi-objective optimization).

Depending on the optimization problem, it may be possible to use optimization algorithms that find the solution in a finite number of steps (sometimes called closed-form solutions), iterative methods that converge to a solution, or heuristics that provide an approximate solution. An exhaustive search (i.e., evaluating every element contained within the search space) is not usually feasible, therefore different iterative strategies are based on different methods to ensure that some sequence of actions, or iterations, converges to an optimal (or close to optimal) solution. One major criterion for choosing an optimization method is the complexity of the evaluation and the required number of function evaluations to find the solution, as these are often related to a large computational cost.

Optimization problems are commonly multi-modal, i.e, they have more than one optimal (or close to optimal) solution. To some problems, there could be various global minima (same value of the evaluation function) or there could be a mix of global and local minima. As a direct implication of this, iterative and heuristic algorithms tend to be sensitive to the starting point. Most iterative algorithms are deterministic, i.e, they always reach the same result under the same conditions, however, small changes to the starting point can yield different results even when following the exact same rules. In case multiple solutions are wanted, or in order to check the convergence, it is common practice to make multiple runs of the algorithm with different starting points.

Optimization is itself a vast field of research that is divided into a series of subfields based on the particularities of the optimization problems that they are intended to solve (e.g. whether the variables are continuous or discrete).

2.4.1 Gradient Methods

Gradient methods can be described as iterative methods in which the search direction is defined by the gradient[1] of the objective function at the current point, where the term point is used here instead of feasible solution .

Most algorithms for unconstrained gradient-based iterative optimization can be described as follows:

1. Initialize the algorithm by selecting a random initial point $\mathbf{x}$.
2. Compute a search direction. Compute the vector $\mathbf{g}$ that defines the descent direction in the n-dimensional search space.
3. Compute the step length. Find a positive scalar, μ such that $\mathcal{F}(\mathbf{x} + \mu\mathbf{g}) < \mathcal{F}(\mathbf{x})$.
4. Update the point position $\mathbf{x} \leftarrow \mathbf{x} + \mu\mathbf{g}$.
5. Test for convergence. If the stopping criteria satisfied, the current point is the solution, if not go back to step 2.

The difference between the various types of gradient-based algorithms is the method that is used for computing the search direction: gradients, Hessians[2], or only function values. The general idea behind gradient-based optimization is shown in Figure 2.5, which also illustrates the influence of starting points in the minimization process.

The computational cost of each iteration depends on what is to be evaluated. Evaluating gradients and Hessians improves the rate of convergence for functions in which these quantities exist and

[1] The gradient is a multivariable generalization of the derivative.

[2] A Hessian is a square matrix of second-order partial derivatives of a scalar-valued multivariable function, that describes the local curvature at the evaluated point.

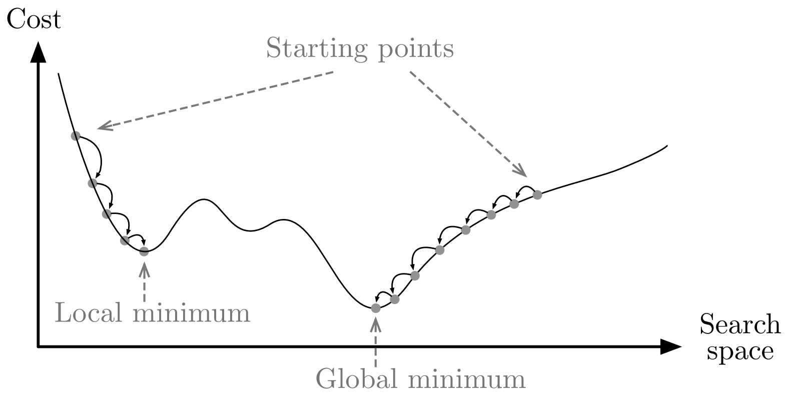


Figure 2.5: Schematic representation of gradient-based iterative optimization.

are sufficiently smooth, however, such evaluations are more computationally expensive, thus there is a trade-off between the number of iterations and the complexity of the evaluation.

2.4.2 Evolutionary Algorithms

Evolutionary algorithms are inspired by the process of natural selection. They use mechanisms such as reproduction, recombination and mutation, with candidate solutions playing the role of individuals in a population. They are part of a larger group of algorithms knows as heuristics. A heuristic is any algorithm which has not mathematical guarantee of finding the optimal solution to a problem. Nevertheless heuristics are useful for getting an approximate solution when classic methods fail to find any exact solution or take too much time to find it.

A typical implementation of an evolutionary algorithm follows the next outline:

1. Initialize the algorithm by creating a population of random individuals.
2. Evaluate the fitness of each individual in the population.
3. Sort the individuals according to the fit criterion, select the best performing individuals for reproduction and dispose of the remaining individuals.
4. Breed new individuals (offspring) from the selected best- performing individuals (parents).
5. Renew the population with the newly breed individuals (new generation).
6. Repeat the process starting from step 2 until the stop criterion is met (fitness level, number of generations, time limit, etc.).

There are many types of evolutionary algorithms, of which Genetic Algorithms (GA) are the most popular. In GAs a candidate solution to the optimization problem is often presented in the form of a string of numbers (commonly binary) referred to as the ‘genome’ where each of said numbers is called a ‘gene’. The best solution is found by evolving an initial set of genomes using the basic genetic operators:

- Selection, the process of allowing the best solutions (genomes) to pass on their ‘genes’ to the next generation of the algorithm.

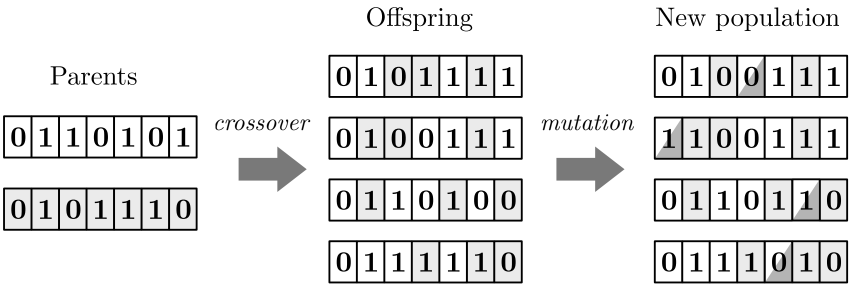


Figure 2.6: Schematic representation of population evolution on an genetic algorithm.

- Crossover, the process of taking two (sometimes more) of the best performing genomes and recombining portions of them to produce a new genome. By recombining portions of good solutions, the genetic algorithm is intended to create a better solution

- Mutation, the process of introducing random changes to the genome of an individual. Mutation attempts to prevent the genetic algorithm converging to a local minimum by stopping the genomes becoming too similar to one another.

These operators work in conjunction with each other to make the algorithm successful in finding a good solution. Also, the specific method in which the operators work should match the chromosome's representation of the solution; e.g., performing the crossover in predefined blocks of genes when some variables are known to be grouped together. Figure 2.6 shows a schematic representation of population evolution on a genetic algorithm.

GAs are a valid option for certain optimization problems but they suffer from a series of limitations, the most relevant being:

- Repeated fitness function evaluation. Every individual of every generation has to be evaluated. This specially becomes a problem when the evaluation function has a high computational cost.

- Poor scaling. The search space grows exponentially with the number of variables (genes). Problems with many variables need very large populations and a large number of generations to converge, which relates to the previous point.

- Tendency to converge towards local optima. The population can quickly become too similar. Although this can be alleviated by forcing a heterogeneous population by different methods such as increasing the mutation rate.

- The best solution is found only by comparison to other solutions. The stop criterion is not often clear.

In general the suitability of GAs is dependent on the amount of knowledge of the problem; well known problems often have specialized approaches that may be more efficient in terms of speed of convergence.

2.4.3 Least Squares

The Least Squares (LS) method is an important technique used to approximate the solution of a system with more equations than unknowns, i.e, an overdetermined system. LS is widely used for obtaining estimates of the parameters in a statistical model based on observed data. The unknown values of the parameters in the model function are estimated by finding the numerical values that minimize the sum of the squared residuals between the observed responses and those predicted by the model. The problem can be described as follows: Assuming a physical phenomenon is modeled by:

$$y = \mathcal{F}(x; \lambda_1, ..., \lambda_N), \tag{2.9}$$

where y is a dependent variable, x a independent variable, and λ_n, $n = 1, ..., N$ is an adjustable parameter. In order to find the best parameters we need a dataset composed of M data pairs (x_j, y_j), $j = 1, ..., M$. The goal is to find optimal parameter values by minimizing S, the sum of squared residuals:

$$S = \sum_{j=1}^{M} r_j^2, \text{ where: } r_j = y_j - \mathcal{F}(x_j; \lambda_1, ..., \lambda_N). \tag{2.10}$$

The parameters (λ_n) are treated as variables for the optimization and the predictor variable values (x_i) are treated as coefficients. It is important to note that the estimates of the parameter values are not the same as the true values of the parameters, and are usually denoted by $\hat{\lambda}_n$ to emphasize this fact.

Depending on whether or not the residuals are linear in all unknowns; LS problems can be divided into linear and nonlinear models. For linear models, the LS minimization is usually done analytically using calculus and it commonly has a closed-form solution. On the other hand, for nonlinear models, the minimization must almost always be done using iterative numerical algorithms.

Let us consider a linear model given by:

$$y_j = \lambda_1 x_{j1} +, ..., +\lambda_N x_{jN} + \epsilon_j, \ \ j = 1, ..., M, \tag{2.11}$$

or in matrix form: $\mathbf{y} = \mathbf{X}\lambda + \epsilon$, where λ is the parameter vector of length N, $\mathbf{X}$, is a $M \times N$ matrix denoting the values of all the independent variables associated with a particular value of the dependent variable vector $\mathbf{y}$ of length M and ϵ is the error vector also of length M. From here, the parameters can be estimated as: $\hat{\lambda} = \mathbf{X}^{-1}\mathbf{y}$, where $\mathbf{X}^{-1}$ is the inverse of $\mathbf{X}$. However, a vector that solves the system may not exist, or if one does exist, it may not be unique ($\mathbf{X}$ could be non-invertible). In these situations, the system can be still be solved by using different numerical methods such as the Moore-Penrose pseudoinverse matrix, given by: $\mathbf{X}^{-1} \approx (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. The solution can then be found with:

$$\hat{\lambda} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.12}$$

Although this equation can work in many applications, it is not computationally efficient so it is preferable to use more specialized solutions based on known properties of $\mathbf{X}$.

In case the model is nonlinear, a direct solution is likely nonexistent or at least much more difficult to find, so iterative solutions are commonplace. The basis of nonlinear LS is to approximate the model by a linear one and to refine the parameters by successive iterations. Although, there are many examples in the literature where different methods are used for a great variety of non-linear data-fitting problems. Some of the most widely employed solutions for nonlinear LS are the Gauss-Newton method [Har61] and its variations such as Levenberg-Marquardt algorithm [Mor78]. As any other iterative algorithm the most immediate implication is the need for initial parameters, thus convergence is not guaranteed.

A variation of the LS method is the Weighted Least Squares (WLS), useful for estimating the parameters of models with differing degrees of variability over the combinations of the predictor values. Each term will then include an additional weight, w_j, that determines how much each observation in the data set influences the final parameter estimates. The criterion that is minimized then becomes:

$$S = \sum_{j=1}^{M} w_j {r_j}^2. \tag{2.13}$$

Weighted LS is usually solved analytically for linear models and numerically for nonlinear models.

2.5 Maximum Likelihood

Maximum Likelihood (ML) estimation is a technique to find the most likely parameters of a function that explains a set of observed data. Assumming that the distribution of the observed random variable(s) is governed by a Probability Density Function (PDF): $\mathcal{F}\left(x; \lambda_1, ..., \lambda_N\right)$, where λ_n, $n = 1, .., N$ are the hidden parameters, the likelihood function associated to a set of observations x_j, $j = 1, ..., M$ is:

$$\mathcal{L} = \prod_{j=1}^{M} \mathcal{F}\left(x_j; \lambda_1, ..., \lambda_N\right). \tag{2.14}$$

In most cases it is convenient to work with the natural logarithm of the likelihood function, called the log-likelihood:

$$\log(\mathcal{L}) = (\sum_{j=1}^{M} \log\Big(\mathcal{F}\left(x_j; \lambda_1, ..., \lambda_N\right)\Big). \tag{2.15}$$

The logarithm is a strictly increasing function, thus, the logarithm of a function and the function itself achieve their maximum value at the same point. The general technique for solving ML problems involves setting partial derivatives of log($\mathcal{L}$) (taken with respect to the unknown parameters) equal to zero and solving the resulting equations (usually non-linear) to find the maximum. Log-likelihood is favored since the logarithm of a product of terms is a sum of individual logarithms, and the derivative of the latter often easier to compute. The ML estimator is then found by solving:

$$\hat{\lambda} = \arg\max\big(\log(\mathcal{L})\big). \tag{2.16}$$

It is worthwhile to note that, the least squares estimator we saw on the previous section is equivalent to the maximum likelihood estimate when the experimental errors follow a normal distribution [Dij83].

2.6 Sound Source Localization

Source Location Estimation relates to how the sound signals received by a group of microphones can be used to estimate the location of an emitting sound source. This can be achieved by measuring the individual time instants at which a particular signal of interest appears on the recorded signals and combining those measurements to pinpoint the location of origin of the signal. The basic types of measures used for sound source localization are:

- **Time of Flight (ToF)**: the travel time of an acoustic signal from a single source to a single receiver.
- **Time of Arrival (ToA)**: the time at which an acoustic signal reaches a single receiver without knowledge about its emission time.
- **Time Difference of Arrival (TDoA)**: the difference in ToA between a pair of receivers.
- **Direction of Arrival (DoA)**: the angle formed by an acoustic wavefront and a pair of receivers.

All of these measures are related to the time lag caused by the propagation of acoustic signals between two points in space. Therefore, they can be obtained by estimating the displacement between two audio signals which is commonly done using the cross-correlation function.

2.6.1 Cross-Correlation

Cross-correlation is a measure of the (linear) similarity of two signals as a function of the displacement of one relative to the other. The cross-correlation of two discrete real valued signals $x(t)$ and $y(t)$ is defined as:

$$r_{xy}(\tau) = \sum_{t=-\infty}^{\infty} x(t)y(t+\tau),\ \tau = 0, ..., \pm L\ , \tag{2.17}$$

where τ represents the time lag, or displacement, between the signals. Cross-correlation is similar to convolution, for discrete real valued signals, they differ only in a time reversal in one of the signals so cross-correlation can also be defined as:

$$r_{xy}(\tau) = x(-\tau) * y(\tau). \tag{2.18}$$

The most common way of computing the cross-correlations is to do so in the time-frequency domain since the convolution becomes an element-wise multiplication in the frequency domain and the time reversal becomes complex conjugation. The DFT of the cross-correlation function can be calculated with:

$$R_{xy}(k) = X(k)Y^*(k), \tag{2.19}$$

where operand $(.)^*$ indicates the complex conjugate. By performing the calculation in the frequency domain and then taking the IDFT it becomes:

$$r_{xy}(\tau) = \frac{1}{L}\sum_{k=0}^{L-1} X(k)Y^*(k)e^{i\frac{2\pi}{L}k\tau},\ \tau = 0, ..., L-1\ . \tag{2.20}$$

Note that expressions (2.19) and (2.20) are actually defining the circular cross-correlation. In order to obtain the full length r_{xy}, $x(t)$ and $y(t)$ have to be padded with a string of L zeros before computing their DFT.

Once the cross-correlation between the two signals is calculated, the maximum (or minimum when the signals are negatively correlated) of the resulting function indicates the point in time where the signals are best aligned; i.e., the time lag between the signals is found as the argument of the maximum (arg max), of the cross-correlation, given by:

$$\hat{\tau}_d = \arg\max_{\tau}(r_{xy}(\tau)) \tag{2.21}$$

In some situations cross-correlation fails to correctly estimate the time lag between the signals, mainly due to the appearance of strong reflections and coherence loss between the signals caused by different propagation paths. In those situations is common to use the Generalized Cross-Correlation (GCC) [KC76] which introduces a weighting function into equation 2.20 in order to alleviate undesired effects. The GCC function can be defined as:

$$g_{xy}(\tau) = \sum_{k=0}^{L-1} \psi(k) X(k) Y^*(k) e^{j\frac{2\pi}{L}k\tau}. \tag{2.22}$$

Notice that the scaling factor $\frac{1}{L}$ of the IDFT has been ignored since it has no effect on the estimation of the time lag between the signals.

Various algorithms can be defined by using different weighting functions, some of which depend on the specific observations while others are based on statistical properties or estimates. Some weighting functions have been shown to work well empirically in many situations, such as the PHAse Transform (PHAT), in which the the magnitudes of the signals are canceled (sometimes called whitening). The PHAT weights are defined by:

$$\psi_{\text{PHAT}}(k) = \frac{1}{|X(k)||Y^*(k)|} \ , \tag{2.23}$$

so that the GCC function becomes:

$$g_{xy}(\tau) = \sum_{k=0}^{L-1} \psi(k) \frac{X(k)}{|X(k)|} \frac{Y^*(k)}{|Y^*(k)|} e^{i\frac{2\pi}{L}k\tau}. \tag{2.24}$$

This is commonly known as GCC-PHAT and it is a widely used method for time lag estimation with acoustic signals. It is worth noting that the whitening effect works well for broadband signals, but when the target signal is narrowband it amplifies background noise.

Since we are working with discrete signals, the cross-correlation can only be evaluated at non-integral values of τ dictated by f_s. In certain applications its is good practice to increase the resolution of the cross-correlations by using some method such as signal up-sampling or quadratic peak interpolation [CHOS95].

It is important to notice that the cross-correlation is affected by coherence loss in the received signals between different propagation paths. A direct implication of this fact is that signals recorded at distant locations or in an environment with low SNR are prone to produce false time lag estimates.

2.6.2 Time Measures

The time lag estimate obtained from the cross-correlation can have different meanings depending on which signals have been used to obtain it. Let us have a group of M microphones, where $x_j(t)$ and $\mathbf{p}_j = [\mathrm{x}_j, \mathrm{y}_j, \mathrm{z}_j]^T$ (coordinates in the 3D euclidean space) denote the received signal and location of the j^{th} microphone respectively. Let us also have a sound source at $\mathbf{p}_s$ emitting signal $s(t)$. Since we are working on the euclidean space, the distance between points $\mathbf{p}_j$ and $\mathbf{p}_k$ is given by:

$$d_{jk} = ||\mathbf{p}_j - \mathbf{p}_k|| = \sqrt{(\mathrm{x}_j - \mathrm{x}_k)^2 + (\mathrm{y}_j - \mathrm{y}_k)^2 + (\mathrm{z}_j - \mathrm{z}_k)^2}, \tag{2.25}$$

where operator $||.||$ is the euclidean distance, also known as the L^2 norm.

In those situations in which the source signal is known *a priori*; assuming a point source and a direct propagation path, the cross-correlation between $s(t)$ and $x_j(t)$ yields an estimate $\hat{\tau}_j$ of the ToA at the j^{th} microphone. Furthermore, if the emission time of the source (t_s) is known in relation to the time base of the microphone it is possible to obtain the ToF as $\delta_j = \tau_j - t_s$.

On the other hand, the most usual situation is to not have any knowledge of $s(t)$. If a direct path from the source to both the j^{th} and k^{th} microphones exists, the TDoA can be measured as the time lag in between the two microphone signals ($\hat{\tau}_{jk}$). In case the ToAs of both microphones are known the TDoA can also be calculated as: $\hat{\tau}_{jk} = \hat{\tau}_j - \hat{\tau}_k$. The theoretical TDoA can be calculated from the distance difference of the microphone pair to the source, the speed of sound (c) and the sampling frequency (f_s), as:

$$\tau(\mathbf{p}, \mathbf{p}_j, \mathbf{p}_k) = \frac{||\mathbf{p}_s - \mathbf{p}_j|| - ||\mathbf{p}_s - \mathbf{p}_k||}{c} f_s \ . \tag{2.26}$$

The DoA is closely related to the TDoA between the two microphones, it can be obtained from the former by applying basic trigonometry. Usually the DoA is calculated under the assumption that the source is in the far field of the array, so that the impinging wavefront becomes planar. A sound source may be considered to be in the far-field of the microphone pair if: $||d_{jk}|| \leq \frac{2r^2}{\lambda_{\text{MAX}}}$, where r is the range, or the minimum distance between the source and the microphones, and λ_{MAX} is the longest wavelength emitted by the source [McC01]. Under this condition the DoA is given by:

$$\alpha_{jk} = \cos^{-1}\left(\frac{c}{f_s}\frac{\tau_{jk}}{d_{jk}}\right). \tag{2.27}$$

Figure 2.7 shows a schematic representation of how DoA can be estimated using the described relations. Unfortunately, a linear array (1D) in a 2D scenario can only discern DoAs between $-\pi/2$ and $\pi/2$ radians, which leads to a problem known as 'DoA ambiguity'. Since $\cos(\alpha_{jk}) = \cos(-\alpha_{jk})$, for every τ_{jk} , there are two equally feasible DoAs. It is possible to use additional DoA estimations between different microphone pairs to solve the ambiguity, although, an n-dimensional microphone array on a higher dimensionality space will still suffer from DoA ambiguity. It is also important to notice that the accuracy of the DoA estimate depends on τ_{jk}, being maximum when $\tau_{jk} = 0$ and minimum when $\tau_{jk} = \pm\frac{d_{jk}}{c} f_s$.

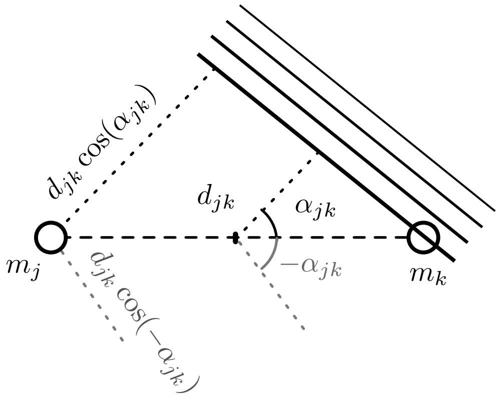


Figure 2.7: Schematic of a planar sound front impinging a microphone pair.

2.6.3 Efficient Cross-Correlation Computation

Time estimates are obtained from the time lag between an emitted signal and a received signal. Most often the emitted signal is unknown, hence, time lag has to be obtained by cross-correlating the signal received by two microphones. However, time lags obtained with direct cross-correlation tend to be related to the strongest signal, which is not necessarily the signal of interest. In some instances it is possible to control the signal being emitted so that the time lag can found from the cross-correlation of the received signal with a known reference signal. This is commonly used in active methods where the nodes are equipped with a speaker so that they become sources. Since long reference signals allow for accurate time lag estimates even on low SNRs conditions, typically, the time duration of such signals ranges from hundreds of milliseconds to a few seconds. Unfortunately, long cross-correlations are not well suited for small embedded systems in which the computing power and memory space are limited.

As we discussed on section 2.6.1, the cross-correlation is often performed in the frequency domain using the DFT for improved efficiency. Furthermore, almost every implementation is done using the Fast Fourier Transform (FFT) [CLW69] for the same reason. The FFT is a very well-known algorithm that re-expresses the DFT of a signal of an arbitrary composite length $L = L_1 L_2$ in terms of L_1 DFTs of size L_2, and does so recursively to reduce the computation time. In Big O notation[3] [Knu76], the direct implementation of the DFT using equation (2.5) has complexity $\mathcal{O}(n^2)$, whereas using the FFT the complexity is reduced to $\mathcal{O}(n \log n)$, where n represents the length of the signal (L). Due to the way the FFT works, the algorithm is more efficient when the length of the input signal is a power of two ($L = 2^n$, $n \in \mathbb{Z}$), although it is possible to use it for any L by means of various strategies. Additionally, specific FFT algorithms for real signals (e.g., audio) take advantage of the symmetry properties of the FFT and have a speed advantage over complex algorithms of the same length. Either way, increasing the length of the input signal has a direct impact on the computational complexity of the cross-correlation. Please note that for the remainder of this document we will use the terms DFT and FFT interchangeably, yet, experimental results are obtained using the FFT.

[3] Big O notation is a mathematical notation that provides a theoretical measure of the execution of an algorithm. It can be used to describe how the execution time or the space used (e.g., in memory or on disk) scales in relation to a given problem size (n).

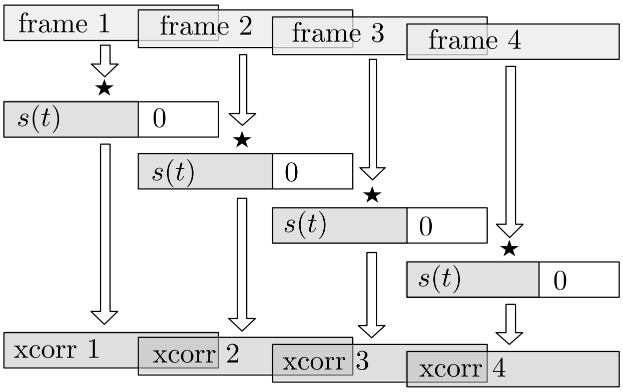


Figure 2.8: Schematic representation of the cross-correlation of a known signal and an audio stream.

Let us assume the worst case scenario in which the emission time of a reference signal of length L_R is unknown. In this case we need to continuously compute the cross-correlation of the received signal with the reference signal until the latter is found. This can be done computing a framed cross-correlation. In order to obtain the best performance, we need to correlate a signal segment long enough to encompass the whole reference signal, so that, frame length $L \geq L_R$. To keep things simple, let us have $L = 2L_R$ and a 50% overlap (L_R), this way we can ensure that the whole reference signal is contained in one of the frames. The cross-correlation of each frame is then obtained by computing the DFT of the input signal, multiplying it element-wise with the precomputed DFT of the reference signal (note that it has to be padded with zeros to make it L samples long), and then applying the IDFT to the product. It is important to keep in mind that for the task at hand, the objective is not the computation of the cross-correlation itself, but accurate estimation of the time lag between the signals. Therefore, obtaining a continuous output is not a priority. This way each frame can be evaluated independently, and the global time lag can be estimated as the time lag at the frame that contains the whole reference signal plus the starting time of said frame. Please note that if the frames are seen as independent to each other it is first necessary to detect which frame contains the received signal. An alternative method is to produce a continuous cross-correlation for a fixed amount of frames using some 'overlapping' technique, such as overlap add, and then finding the global maximum. Figure 2.8 shows a schematic representation of the cross-correlation of a known signal and an audio stream using overlapped frames.

2.6.4 Source Location Estimation

Different localization algorithms employ one or more of the time measures described in the previous section and various mathematical techniques to determine the position(s) of the source(s). Any such algorithm is inherently affected by the measurement error, therefore, finding an exact solution is not feasible. A solution to the problem can still be found by defining an objective function and minimizing it using some optimization method.

Let us assume that the locations of the receiving microphones are fixed and known. It is then possible to estimate the source location by defining an objective function as a sum of errors between the time lag estimates and the theoretical time lags calculated for a candidate position $\mathbf{p}$. The

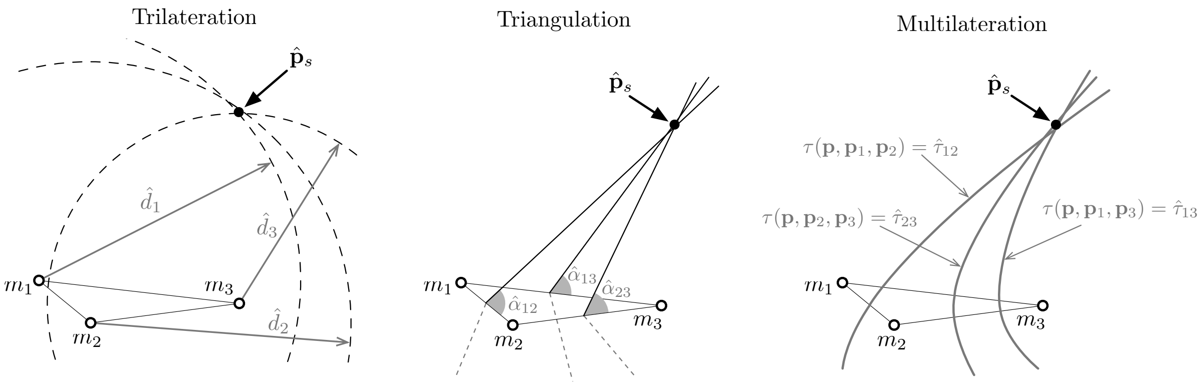


Figure 2.9: Schematic representation of three sound source localization methods in 2D space.

minimization is typically done by casting the problem as a least squares problem. Depending on what measurements are available, different methods can be used to locate the source. Some of the most common methods are: trilateration, triangulation and multilateration. Figure 2.9 shows an schematic representation of these methods with 3 microphones in 2D space.

Trilateration is the process of determining the absolute or relative locations of points using distance measurements, based on the geometry of circles or spheres. The basic concept is that given the distance to an anchor (microphone), the source must be located along the circumference of a circle with a radius equal to the source-anchor distance centered at the anchor. In two-dimensional space, at least three non-collinear microphones are needed. The source position can be estimated from the ToF by solving:

$$\hat{\mathbf{p}}_s = \arg\min_{\mathbf{p}} \left(\sum_{j=1}^{M} \left(\hat{\delta}_j - \frac{||\mathbf{p} - \mathbf{p}_j||}{c} f_s \right)^2 \right). \tag{2.28}$$

Triangulation involves angle (bearing) measurements. As the name implies, it uses the geometric properties of triangles to estimate the source location. Given a number of bearing lines emanating from anchor points, the source is located at the point in space where the lines intersect (a minimum of two bearing lines are needed in two-dimensional space). The anchor points in triangulation are the center of the array formed by a microphone pair. The total number of microphone pairs with M microphones is: $P = \frac{M(M-1)}{2}$. The source position can be estimated from the DoA estimates by solving:

$$\hat{\mathbf{p}}_s = \arg\min_{\mathbf{p}} \left(\sum_{j,k}^{P} \frac{\left(\hat{\alpha}_{jk} - \alpha_{jk}(\mathbf{p})\right)^2}{{\sigma_{jk}}^2} \right), \tag{2.29}$$

where $\alpha_{jk}(\mathbf{p})$ is the theoretical DoA for the j^{th} and k^{th} microphones and σ_{jk} is the standard deviation of the DoA estimates that can be obtained from τ_{jk}, d_{jk} and f_s.

Multilateration works based on TDoAs between microphone pairs. Measuring the distance/time difference at a microphone pair results in an infinite number of possible source locations along a hyperbolic curve. To find the exact source location along that curve, multilateration relies on multiple measurements: a second measurement taken to a different pair of receivers will produce a second curve, which (ideally) intersects with the first. Unlike triangulation, multilateration does not make a far-field assumption. The source location can be found solving:

$$\hat{\mathbf{p}}_s = \arg\min_{\mathbf{p}} \left(\sum_{j,k}^{P} \left(\hat{\tau}_{jk} - \frac{||\mathbf{p} - \mathbf{p}_j|| - ||\mathbf{p} - \mathbf{p}_k||}{c} f_s \right)^2 \right), \tag{2.30}$$

Another popular method is the Steered Response Power (SRP), in which the objective function is expressed as a sum of GCCs (commonly used with PHAT weighting) for al the possible microphone pairs at the time-lag corresponding to a particular point in space $\mathbf{p}$. The SRP algorithm consists in a grid-search that evaluates the objective function on a grid of candidate source locations $\mathcal{G}$. The location of the sound source is estimated as the point that maximizes the SRP:

$$\hat{\mathbf{p}}_s = \arg\max_{\mathbf{p} \in \mathcal{G}} \Big(\sum_{j,k}^{P} g_{jk}(\tau_{jk}(\mathbf{p})) \Big), \quad \text{with: } \tau_{jk}(\mathbf{p}) = \frac{||\mathbf{p} - \mathbf{p}_j|| - ||\mathbf{p} - \mathbf{p}_k||}{c} f_s, \tag{2.31}$$

Notice that τ is only defined for integer values, so $\tau_{jk}(\mathbf{p})$ has to be either rounded to the closest integer or interpolated from the closest existing values. This method can be interpreted as finding the candidate location that maximizes the output of a delay-and-sum beamformer.

A common problem that affects every source localization method is the presence of multiple sound sources. Most methods do not have the ability to select a particular sound source and tend to locate the higher energy source. The problem arises from the way the delay is obtained from the correlation and it is very difficult to solve without a priori knowledge of the signal emitted by the source. In order to locate multiple sources, some methods assume a known number of sources while others apply some kind of threshold to the correlation.

Acoustic source localization is evaluated in terms of the localization error which is given by the euclidean distance between the true source location and its estimate.

2.6.5 Self-Localization

Self-Localization is closely related to source localization with the added constraint of the microphone locations being unknown. The objective is then not only to locate sound sources but also the microphones themselves. There are three main scenarios to be considered depending on the microphone configuration:

- **Array shape calibration**: microphones forming a compact array and sharing a common time base.
- **Microphone configuration**: individual microphones distributed in space and (commonly) lacking synchronization.
- **Microphone array configuration** a group of independent (no synchronicity) microphone arrays whose internal array shape is already known.

Self-Localization methods are based on relative differences on the measurements taken at each microphone, which means that only relative positions can be estimated. In order to obtain the actual positions, it is possible to use some a priori knowledge to translate the relative estimates into positions within a global reference frame. Note that lack of synchronicity impedes the direct estimation of time differences between microphones. In those cases in which the microphones do not share a common time base, the clock offset at each microphone also has to be estimated. Generally speaking, acoustic self-localization is an open ended problem, the solution of which is greatly dependant on a particular set of conditions. That is, what kind of measurements are available and how many unknowns have to be accounted for.

The most restrictive scenario is that in which every source and microphone are independent to one another. Let us have M microphones and N sources located in a p-dimensional space at the points defined by a vector $\mathbf{p}_m \in \mathbb{R}^p$ for the m^{th} microphone and $\mathbf{p}_n \in \mathbb{R}^p$ for the n^{th} source. We consider that we have K time measurements, from each source to each receiver, so that in a fully connected scenario $K = MN$, where each measurement is the reception time τ_{nm} (ToA). Concerning the lack of synchronization, we consider t_n as the transmission time of the n^{th} source and Δ_m as the clock offset of the m^{th} microphone. Defining the ToF of a source-microphone pair as $\delta_{nm} = \tau_{nm} - t_n - \Delta_m$, we can then set an error function given by:

$$E = \sum_{n=1}^{N} \sum_{m=1}^{M} e_{nm}^2, \quad \text{with: } e_{nm} = \tau_{nm} - \delta_{nm} - t_n - \Delta_m. \tag{2.32}$$

Error function E can be minimized by different means, provided that the number of equations is higher than the number of unknowns: $(p+1)N + M < K$. The computational load associated with such minimization is usually significative, being convenient to relax the problem constraints whenever possible.

In contrast to this, the simplest scenario is found when $N = M$ and $\mathbf{p}_n \approx \mathbf{p}_m$ for $n = m$. That is, when each microphone has a source located close enough to it so that, the microphone-source pair can be considered to be located at the same point. A typical example of this configuration is a Wireless Audio Sensor Network (WASN) in which the nodes are equipped with a microphone and a speaker. In such scenario it is possible to use Multidimensional Scaling (MDS) [Kru64], a family of algorithms designed to arrive at an optimal low-dimensional configuration. Given a dissimilarity (distance) matrix $\mathbf{D} = (d_{jk})$, MDS seeks to find $\mathbf{P} = [\hat{\mathbf{p}}_1, ..., \hat{\mathbf{p}}_N] \in \mathbb{R}^p$ so that $||\hat{\mathbf{p}}_j - \hat{\mathbf{p}_k}||$ approximates d_{jk} as close as possible. The estimation of $\mathbf{P}$ can be solved by different means depending on the type of MDS algorithm and the number of elements available on the dissimilarity matrix $\mathbf{D}$ (it is possible to solve the problem when some of the pairwise measurements are missing).

A common problem shared by most self-localization methods is their sensitivity to outliers. Without a proper strategy to asses the quality of the measurements, the minimization can fail to converge due to large time lag estimate errors. This problem is sometimes tackled using outlier detection techniques from simple methods such as limiting the maximum distance between the nodes to iterative solutions like Random sample consensus (RANSAC) [FB87]. Different self-localization algorithms are capable of dealing with different amounts of error; although, in general terms the solution can only be as precise as the measurements themselves.

Acoustic self-localization is often evaluated in terms of the mean localization error which is given by the average distance between the true microphone locations and the obtained estimates.

2.7 Machine Learning

Machine learning is a very broad field that focuses on the design of automated systems that can make decisions and predictions based on data. Such a system "learns" to look for patterns in data based in observation without explicit programming.

Machine learning algorithms are often categorized as supervised or unsupervised depending on whether there is a learning "feedback" available to the system. Supervised systems are those that are "trained" on a pre-defined dataset formed by labeled examples, while unsupervised systems can automatically find patterns and relationships in a given dataset. Another categorization of machine learning algorithms can be made according to the type of output of the system. The most basic distinction for supervised systems is: Classification systems when the predictions are of a discrete nature, Regression systems when the output falls somewhere on a continuous spectrum. In this thesis the main focus is going to be on supervised Classification/Detection algorithms.

In recent years, machine learning has lead to many technological advancements such as computer vision and speech recognition systems. Some of the more relevant tasks of audio-based machine learning are:

- Source identification
- Automatic speech recognition
- Sentiment/emotion recognition
- Speaker recognition
- Source separation
- Automatic music transcription
- Audio stream labeling/tagging
- Audio fingerprinting

2.7.1 Feature Representation

Most machine learning techniques are based around feature space representation. This can be viewed as dimensionality-reduction problem, where an observation in the form of data vector $\mathbf{z}$ of length L: $\mathbf{z} \in \mathbb{R}^L$, is mapped into a lower dimensionality feature vector $\mathbf{x}$, with $\mathbf{x} \in \mathbb{R}^M, M < L$, such that uninformative variance in the data is discarded. The typical approach involves the generation of the feature vector by using a certain set of functions, where each function transforms the input data into a single value or small number of values. The output of all these functions in then concatenated to form the feature vector.

There are many features (i.e., descriptors) that can be used to characterize audio signals [Ler12]. Generally speaking they can be separated into: time-domain and frequency-domain features according to wether they represent the temporal evolution of the signal or its spectral content. Frequency-domain descriptors can also be divided intro spectral and perceptual features. Spectral descriptors,

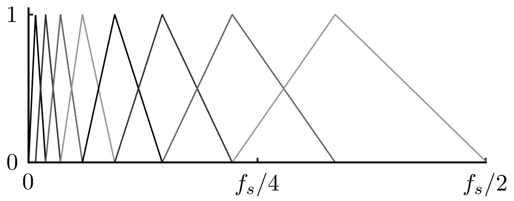


Figure 2.10: Example of a Mel filter-bank with 8 coefficients.

as their name implies, are obtained from the spectrum of the signals, while perceptual descriptors are derived from psycho-acoustic models; their objective is to adapt acoustic measurements to the way humans perceive sound. According to the extent of the description, a given feature could be seen as global or instantaneous. Global descriptors are those obtained from the 'whole' signal (usually requiring segmented events or time localization), while instantaneous descriptors are those obtained from a single time frame.

Perhaps the most widely-used acoustic features are the Mel-Frequency Cepstral Coefficients (MFCCs) [HLM80] which are a compressed perceptual representation of the short-term power spectrum of the signals. MFCCs take into account the logarithmic response of the human auditory system, both in terms of frequency resolution and magnitude. As their name implies, MFCCs are based on the Mel scale, a non-linear frequency scale of pitches judged by listeners to be equal in distance from one another. In order to compute the MFCCs, the magnitude of the DFT of a signal frame is mapped onto the mel scale, using N triangular overlapping windows, formally known as a mel filter-bank (shown in Figure 2.10 with 8 coefficients). The resulting values are known as mel-band energies. The next step is to take the logarithm of each value and then calculating the Discrete Cosine Transform (DCT) [ANR74] of the list of mel log-energies. The output of the DCT are the MFCCs.

The objective of the DCT is to compress the information carried by the MFCCs to the lower coefficients, making it possible to discard some of the higher frequency components further reducing the dimensionality. In many applications this step is ignored in order to reduce computational load and the mel-band log-energies are directly used as features. There are similar approaches to the MFCCs that achieve comparable results, although far less popular such as gammatone filter-banks [PAG95], or the CQT.

Some other of the most commonly used audio descriptors are:

- **Temporal descriptors**: Obtained from the signal waveform, e.g., zero-crossing-rate, auto-correlation coefficients, amplitude, etc.
- **Energy descriptors**: Obtained either in the time-domain or the frequency domain and referring to various energy contents, e.g., global energy, harmonic energy ratio, noise energy, rise time, etc.
- **Spectral descriptors**: Typically obtained from the STFT of the signal and intended to describe the shape of the spectrum, e.g, centroid, skewness, kurtosis, slope, roll-off, etc.
- **Perceptual descriptors**: Derived from a perceptual auditory model, e.g., loudness, brightness, pitch, tonality, etc.

An ideal feature set should be robust (i.e., can be reliably estimated from the available audio), relevant to the classification task, and as invariant as possible to the changes within its natural range. To avoid bias towards features with a wider range, it is good practice to normalize all the features to have zero mean and unit variance, so that a normalized observation becomes:

$$\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma}, \tag{2.33}$$

where μ and σ are the mean and standard deviation of the training set.

On certain applications it is commonplace to start from a large set of features and reduce it to a subset of the most relevant features [LD11] . This process is known as feature selection and has some important implications such as; dimensionality reduction. increased computational efficiency and enhanced generalization (by reducing overfitting). There are many methods that can be used for feature selection, including Principal Component Analysis (PCA)[VDMPVdH09], and heuristics such as genetic algorithms among others.

2.7.2 Computation of Mel Log-Band Energies

Let us consider an audio stream in the time-frequency domain, $S(k, l)$, $k = 1, \ldots, L$, where k is the frequency bin index, and l is the frame index. Considering the DFT of a given time frame as a vector $\mathbf{s}_l = [s_{1l}, \ldots, s_{Ll}]^T$, we can define the periodogram-based power spectral estimate of the t^{th} frame as $\mathbf{s}_t$, with $p_{kl} = |s_{kl}|^2$. Audio is a real-valued signal, so the negative frequency bins of the DFT can be discarded since they are just a reflected and conjugated version of the positive frequency bins. For simplicity let us consider $k = 1, \ldots, \frac{L}{2} + 1$, for even values of L. We want to obtain a perceptual representation of the short-term power spectrum of the input signal compressed into F features, so we are going to reduce the length of $\mathbf{s}_l$ using a Mel filter bank. Each of the F filters of the filter bank is defined as a weighted linear combination of a number of frequency bins of $\mathbf{s}_l$. Let the frequency of the k^{th} bin be $f_k = (k-1)\frac{f_s}{L}$, so that we can define a function to convert Hertz to Mel scale:

$$M(f_{Hz}) = 1125 \ \log\Big(1 + \frac{f_{Hz}}{700}\Big), \tag{2.34}$$

and its inverse:

$$M^{-1}(f_{Mel}) = 700\Big(e^{(f_{Mel}/1125)} - 1\Big). \tag{2.35}$$

Defining a vector $\mathbf{v} = [v_1, \ldots, v_{F+2}]$ that contains $F + 2$ Mel-spaced frequencies (in Hertz) from 0 to $\frac{f_s}{2}$, whose elements are obtained with:

$$v_\rho = \begin{cases} 0, \text{ if } j = 1 \\ M^{-1}\big(\frac{\rho-1}{F+1} M(\frac{f_s}{2})\big), \text{ if } 1 < \rho \leq F+2 \end{cases}, \ \rho = 1, \ldots, F+2, \tag{2.36}$$

it is then possible to represent the filter bank as a matrix ($\mathbf{H}$) of size $F \times (\frac{L}{2} + 1)$, whose elements are given by:

$$h_{jk} = \begin{cases} 0, \text{ if } f_k < \rho_j \\ \frac{f_k - v_f}{v_{f+1} - v_f}, \text{ if } v_f \leq f_k \leq v_{f+1} \\ \frac{v_{f+2} - f_k}{v_{f+2} - v_{f+1}}, \text{ if } v_{f+1} \leq f_k \leq v_{f+2} \\ 0, \text{ if } f_k > v_{f+2} \end{cases}, \ j = 1, \ldots, F, \tag{2.37}$$

This way the Mel log-band energies can be computed as the element-wise natural logarithm of a matrix product given by:

$$\mathbf{x}_l = \log(\mathbf{H}\ \mathbf{s}_t). \tag{2.38}$$

Please note that implementing expression 2.38 is a highly inefficient method of computing the feature vector, due to matrix **H** being largely composed of zeros. The maximum number of elements not equal to zero in **H** is $< L$, for any F, therefore it makes sense to compute the output of each filter independently by only considering those frequency bins relevant to the result. It is also worth mentioning that taking the logarithm at the last step also reduces the computational complexity (compared to taking the logarithm of the squared magnitude of the DFT). It is well known that some functions such as divisions, exponentiations or logarithms take longer to compute than simpler operations such as additions or multiplications, on most digital processors. In fact, base n logarithms are commonly approximated using the base 2 logarithm as: $\log_n(x) = \frac{\log_2(x)}{\log_2(n)}$. This has to do with the existence of very efficient techniques to compute the base 2 logarithm that exploit the binary nature of digital computations. In any case, computing the DFT is likely to be the most time expensive step of the process.

2.7.3 Classification and detection

In machine learning, classification can be described as the process to automatically assign an item to one of a number of categories (classes), based on its characteristics. In supervised learning these categories are predefined in the design stage, where a set of examples called “Training Set” is used to define the rules by which the classification system will operate.

Detection can be seen as a particular case of classification, closely tied to binary classification, in which only two classes exists: the target class and noise. In terms of implementation there is no clear distinction between classification and detection, with binary classification often seen as the default case from where multi-class (more than one target class) classification can be derived. However, the term detection is commonly used when the classification process (binary or otherwise) is applied to a continuous stream of data, in which the most represented class is none of the target classes.

In feature space representation, each observation is a point in an F-dimensional space represented by a vector $\mathbf{x} = [x_1, ..., x_F]^T$. The objective of the classification is to set boundaries to said vector space so that the categories can be defined by discrete regions. Any point falling within a certain region of the vector space will then be assigned to the corresponding category. The ‘decision making’ process can be expressed as:

$$D = \mathcal{F}(y), \ \text{ with: } y = \mathcal{G}(\mathbf{x}), \tag{2.39}$$

where y is the output obtained by classifier function$\mathcal{G}$, $\mathcal{F}$ is the decision rule and D is the decision. In the case of binary classification, the decision function is most often a threshold function, defined as:

$$D = \begin{cases} 0, \text{ if } & y \leq y_0 \\ 1, \text{ if } & y > y_0 \end{cases}, \tag{2.40}$$

where y_0 is the threshold value. The decision rule for multiclass systems is typically the argument of

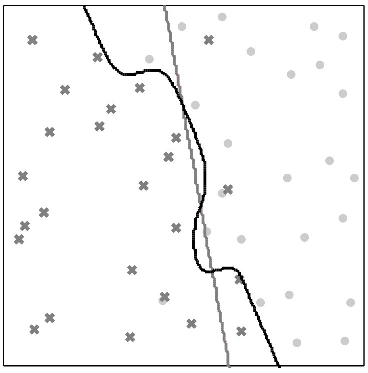


Figure 2.11: Example of linear and nonlinear classification boundaries

the maximum over an output vector $\mathbf{y} = [y_1, ..., y_C]$, where each elements represent the 'score', or predicted probability, of observation $\mathbf{x}$ belonging to one of C classes.

There are multitude of algorithms and many different implementations to perform classification which can be categorized by many attributes; of which we will just mention linear vs nonlinear classification. The decision boundary of a linear classifier is an hyperplane obtained from linear combinations of the feature vector (in two dimensions the decision boundary of a linear classifier is a line). Nonlinear classifiers can be defined as those classifiers that are not linear. The difference between them can easily be seen in figure 2.11 where the boundary between two classes is shown in a two dimensional vector space.

While nonlinear classifiers are capable of achieving better class separation with complex data sets, they are more prone to a phenomenon called *overfitting*. In almost any data set there is always a region on the vector space with overlapping classes i.e. non separable observations. Some nonlinear classifiers are able to pinpoint small regions in the vector space, sometimes containing as little as a single observation. By doing this, some classifiers are often capable of achieving perfect classification of the training set. However, the data set used for the training is just a small set of pre-classified observations that gives an indication of the "true" class distribution in the vector space; thus, specially for small training sets, the boundaries tend to be adjusted in excess resulting in overfitting. An overfitted classifier will sometimes fail to classify new observations due to excessive "learning" leading to what is known as poor 'generalization'.

2.7.4 Least Squares Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a well-known method for dimensionality reduction and classification that plays an important role in the presented work. In LDA, the output is computed as a linear combination of the inputs via a transformation matrix first introduced by Fisher [Fis36]. There is a close connection between multivariate linear regression and LDA, thus, it is possible to cast LDA as a least squares problem [Ye07] in what is known as Least Squares Linear Discriminant Analysis (LS-LDA).

Let us consider a set of O observations, each one containing F input features, $\mathbf{x}_j = [x_{j1}, ..., x_{jF}]^T$,

$j = 1, ..., F$. The output of a linear discriminant is obtained as a linear combination of the L inputs (observation), according to:

$$y = b + \sum_{j=1}^{F} w_j x_j, \tag{2.41}$$

where w_j are the weights applied to each feature and b is the bias term.

For a given system with C possible classes, having O available observations and with L inputs per observation, we can rewrite (2.41) in matrix form using a weight matrix $\mathbf{V}$ of dimensions $C \times (L+1)$ and an input matrix $\mathbf{Q}$ of dimensions $(L+1) \times O$:

$$\mathbf{V} = \begin{bmatrix} w_{11} & \dots & w_{L1} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ w_{1C} & \dots & w_{LC} & b_C \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} x_{11} & \dots & x_{1O} \\ \vdots & \ddots & \vdots \\ x_{L1} & \dots & x_{LO} \\ 1 & \dots & 1 \end{bmatrix}, \tag{2.42}$$

where the output $\mathbf{Y}$ of dimensions $C \times O$ is calculated as $\mathbf{Y} = \mathbf{VQ}$.

Let us also define a binary target matrix $\mathbf{T}$ of dimensions $O \times C$ representing the desired outputs (correct classes), so that the error can be expressed as the difference between the predicted output and the true outputs:

$$\mathbf{E} = \mathbf{Y} - \mathbf{T}, \ \text{ with: } \mathbf{T} = \begin{bmatrix} t_{11} & \dots & t_{1N} \\ \vdots & \dots & \vdots \\ t_{C1} & \dots & t_{CN} \end{bmatrix}, \tag{2.43}$$

As we have seen in section 2.4.3, in order to use the LS criterion to obtain weight matrix $\mathbf{V}$, we need to minimize the sum of squared residuals given by:

$$S = \sum_{o=1}^{O} \sum_{c=1}^{C} {e_{on}}^2 = \|\mathbf{VQ} - \mathbf{T}\|^2, \tag{2.44}$$

where $e_o c$ are the elements of error matrix $\mathbf{E}$. According to [Ye07], the problem is solved by differentiating (2.44) with respect to every weight in $\mathbf{V}$, which yields the following expression:

$$\mathbf{V} = \mathbf{TQ}^T(\mathbf{QQ}^T)^{-1} \tag{2.45}$$

The main advantage of LS-LDA is its simplicity. The computational computational cost of the classification is directly proportional to the number of features, thus, limiting the computational cost of the classifier is equivalent to limiting the number of features. Another advantage comes from its fast training times since the weights are found using a closed expression. LS-LDA is often used as the cost function for heuristic feature selection in some applications where training the intended classifier takes a long time.

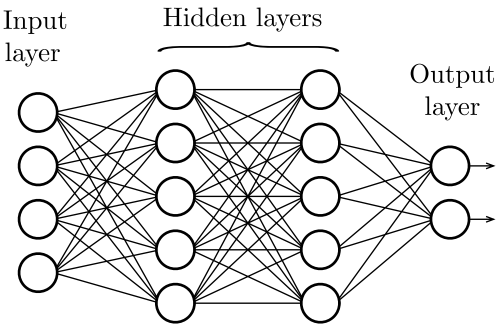


Figure 2.12: Schematic representation of a multilayer perceptron.

2.7.5 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computing systems inspired by biology that are based on a series of interconnected units or nodes called artificial neurons. A typical ANN is composed of a number of layers of units where each artificial neuron computes a value as a non-linear function of the weighted sum of its inputs and then "propagates" it to other neurons in different layers. The learning ability of ANNs resides in the weights that a neuron gives to each one of its inputs that are adjusted during the training stage.

One of the most common types of ANNs (and one of the simplest) is the Multilayer Perceptron (MLP), a class of feedforward network (there is no feedback between layers) composed of a series of fully connected layers (every neuron in one layers is connected to every neuron in the following layer). A typical MLP used for classification is formed by an input layer, one or more hidden layers and and output layer (typically with one output neuron per class). No computation is performed at the input layer, it just passes on the information (i.e., the features) to the hidden neurons. Figure 2.12 shows a schematic representation of an MLP with two hidden layers. For some artificial neurons on any layer, let there be N inputs, so the output of the k^{th} neuron is:

$$y_k = \mathcal{A}\Big(b_k + \sum_{j=1}^{N} w_{kj} x_j\Big), \tag{2.46}$$

where x_j, $i = 1, ..., N$ are the k^{th} neuron inputs, w_{kj}, $i = 1, ..., N$ are the assigned weights, b_k is the bias term, and $\mathcal{A}$ is the activation function or transfer function. The number of inputs for any layer, other than the input layer, is the number of neurons in the previous layer. The activation function serves different roles in ANNs, the most significative being to provide the network with the ability to produce non-linear decision boundaries via linear combinations of weighted inputs. This is achieved by introducing a non-linear activation function, typically a sigmoid function defined as:

$$\mathcal{A}(y_k) = \frac{1}{1 + e^{-y_k}}, \tag{2.47}$$

although, there are many types of activation functions intended for different situations [KO11].

Supervised learning is usually tackled using 'backpropagation' [Wer90], an algorithm based on gradient descent. Given an ANN and an error function, the method calculates the gradient of the error function with respect to each neuron's weights. The "backwards" part of the name refers to the

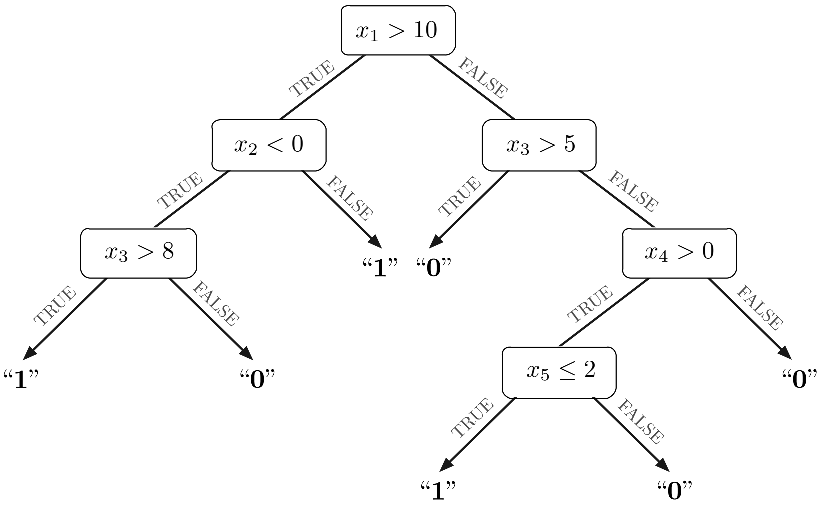


Figure 2.13: Binary decision tree with 6 nodes and 4 levels.

fact that the calculation of the gradient proceeds backwards through the network, from the output layer to the input layer. The partial computations of the gradient of one layer are reused in the computation of the gradient of the previous layer, allowing for an efficient computation.

2.7.6 Decision Trees

A decision tree is a tree-like graph or model that predicts the value of an observation following a simple set of rules inferred from a training dataset. Growing (training) a decision tree is a recursive process that involves deciding which features to choose and what conditions to use for splitting.

Decision trees are formed by a series of decision nodes and leaf nodes. A decision node tests some feature for some condition(s) that determine which 'branch' to follow. The topmost decision node in a tree (which corresponds to the best predictor) is called root node. In the simplest case, any decision has at most two outcomes (i.e., a binary tree). Leaf nodes represent a classification or decision and are reached by following a path of decisions.

The size of a given tree is best described by its 'depth', or the number of levels of decision nodes. A binary decision tree without leaf nodes until the very last level is formed by 2^{K-1} decision nodes, where K is its depth. However, a trained decision tree typically has leaf nodes at multiple levels, making it possible to reach a decision by traversing various paths of various lengths (i.e., levels). Figure 2.13 shows a simple decision tree formed by 6 decision nodes and spanning 4 levels with multiple path lengths.

The size of a decision tree depends on the size and quality of the training set, so that an optimal decision tree is defined as a tree that accounts for most of the data, while having the minimum possible number of nodes, therefore minimum depth. There are a number of techniques to reduce both the size of the tree and overfitting. The most popular is 'pruning' that works by removing sections of the tree that provide little performance improvements based on a posteriori probabilities.

The type of decision tree most relevant for this thesis is called Random Forest (RndF). This classifier is formed by multiple decision trees each of which is trained using different subsets of features and/or examples from the training data. In more formal terms, a RndF is an ensemble of bootstrap-aggregated (or bagged) [Bre96] decision trees, where the output is obtained by consensus.

2.7.7 Other Classification algorithms

Some other popular classifiers include:

- **Support Vector Machine (SVM)**: SVMs define one or more hyperplanes in a high-dimensional space. However, since it is common to have data that is not linearly separable in the original space, SVMs use a technique called the "kernel trick". It consists on using a function known as a 'kernel', that transforms the original low dimensional input space into a higher dimensional space with the goal of making linear separation possible. With a suitable kernel function SVMs can perform non-linear classification.

- **K-Nearest Neighbors (K-NN)**: In K-NN classification, the output is a class membership. The output is obtained by using some functions to measure the "distance" of an input to each element in a database (training set) and then selecting the mode of the class of the k nearest elements. It is one of the simplest machine learning algorithms and has the particularity of deferring all the computational load to the classification stage.

- **Gaussian Mixture Model**: GMMs assume that the observed data is made up of a mixture of several Gaussian distributions with different means and variances, and fits one such a mixture over those samples in the training data belonging to each class. GMMs can be applied to different tasks. When they are used as a classifier the probability of a given observation is obtained by multiplying each mixture component (i.e., feature) by its associated mixture weight and adding them together (the mixture weights must add to one). The observation is then assigned to the class with the highest probability.

- **Hidden Markov Model**: A HMM is defined by a finite set of states, each of which associated with a multidimensional probability distribution, and related by a set of transition probabilities between states. The states are not directly visible, only the output is, hence the term 'hidden'. HMMs can be used to classify sequences of observations. Given a set of HMMs, each trained on data belonging to one class, it is possible to compute the likelihood that a given sequence has been generated by any of the HMMs. Then, the sequence can be classified as belonging to the class with the highest likelihood.

2.7.8 *K*-means Clustering

In contrast to supervised classification, unsupervised learning methods do not require a predefined set of examples. The objective of this family of methods is to infer a function that describes some hidden structure learnt from unlabeled data.

One of the simplest unsupervised learning algorithms K-means clustering [Jai10]. The goal of K-means is to find K distinct groups (clusters) in the input data. The algorithm works iteratively to assign each observation to one of K groups, where observations are treated as data points in a F-dimensional space and are clustered (i.e., grouped) based on feature similarity.

Let us have a set of observations $\mathbf{X} = \{\mathbf{x}_n\}$ in an F-dimensional space, $\mathbf{x}_n \in \mathbb{R}^F$, that are to be assigned to one of K clusters in $\mathbf{Z} = \{\mathbf{z}_k \in \mathbb{R}^F,\ k = 1, \dots, K\}$, where cluster $\mathbf{z}_k$ contains a number of $\mathbf{x}_n$. Let μ_k be the centroid, or the mean of the points in the k^{th} cluster ($\mathbf{z}_k$). The goal of K-means is to minimize the sum of the squared distances between μ_k and the observations belonging to cluster $\mathbf{z}_k$ over all K clusters, given by:

$$S(\mathbf{Z}) = \sum_{k=1}^{K} \sum_{\mathbf{x}_n \in \mathbf{z}_n} \left(||\mathbf{x}_n - \mu_k||\right)^2. \tag{2.48}$$

Typically, K-means obtains the centroids using F-dimensional Euclidean distances, as shown by expression 2.48, although, it is possible to use various distance measures. Therefore, the location of the centroids depends on the selected distance measure since they are computed differently.

Regarding computational complexity, finding the optimal solution of K-means clustering is considered a NP-hard problem[4]. However, it is worth noting that this fact only affects the training time of the algorithm, and that, there is a good number of efficient strategies to find an approximate solution

Assigning an observation to one of the clusters is a very simple process. Each centroid defines one of the clusters, thus, each data point is assigned to its nearest centroid based on a squared distance metric. In more correct terms, if μ_k, $k = 1, \dots, K$ are the centroids found by the algorithm, then each data point $\mathbf{x}_n$ is assigned to a cluster based on:

$$k = \arg\min_k \left(\operatorname{dist}(\mu_k, \mathbf{x}_n)^2 \right), \tag{2.49}$$

where operator dist(x) is the distance metric used to obtain the centroids.

2.7.9 Multiobservation Classification

Multi-observation classification, which can also be referred to as joint classification or classification fusion, has similarities with ensemble methods such as Bootstrap Aggregation (bagging) or boosting [OM99]. Classic ensemble techniques take advantage of an ensemble of “weak” classifiers in order to “boost” the classification using decision fusion. As we previously mentioned, when working within a spatially diverse scenario, the signal received at each of the microphones is subjected to variations that cannot be predicted during the training stage, decreasing the performance of the classifier. However, since there are multiple observations of a single acoustic event available, it is possible to combine the decision taken for each observation to achieve higher accuracy [LS97] in the same way as ensemble methods.

Let us have an ensemble of M classifiers providing an output $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ with $\mathbf{y}_m = \mathcal{G}(\mathbf{x}_m)$, $l = 1, ..., M$, where $\mathcal{G}$ is a function shared by every classifier in the ensemble and $\mathbf{x}_m$ is a local observation of $\mathbf{x}$. By combining the individual outputs, we aim at obtaining a higher accuracy than that of

[4] NP is the set of problems solvable in polynomial time, which is defined by a time complexity $\mathcal{O}(n^P)$. For some K and F (dimensions), K-means can be exactly solved in time $\mathcal{O}(n^{FK+1})$, where n is the number of samples to be clustered.

the best classification. The decision taken by the classifier team is then: $D = \mathcal{F}\left(\mathbf{y}_1, ..., \mathbf{y}_M\right)$, where $\mathcal{F}$ is the fusion method. Previous research shows that the major factor for a better accuracy is the diversity in the classifier team and so, the fusion method is of secondary interest [TDVB97]. However, choosing an appropriate fusion method can further improve the performance of the ensemble. For the time being we are going to focus on averaging and majority vote since they are the most commonly used, although it goes without saying that there are more alternatives [CV86, ZHNZ11].

Averaging is self explanatory; the output of the ensemble is averaged so that it can be treated as that of a single classifier. In majority vote, the outputs are first "hardened", turning them into individual decisions, so that the final decision is obtained by selecting the class label most represented among the M outputs. Majority vote and Averaging are closely related, in fact, they are equivalent when the output of the classifier ensemble is already a local decision vector (there is no access to real valued outputs). If we were to perform one-against-all multi-class classification using averaging, where the output of each classifier in the ensemble is a vector $\mathbf{y}_m = [y_{m1}, ..., y_{mC}]^T$, the final decision D would be obtained with the following expression:

$$D = \arg\max_c \left(\frac{1}{M} \sum_{m=1}^{M} y_{mc} \right) \tag{2.50}$$

Since it is very likely for some of the classifiers to perform better than others, it is only logical to assume that their decisions should be given a certain degree of priority or weight [LW94] . Weighted decision fusion adds an additional step to the methods previously discussed. The basic idea is to assess the contribution of each classifier to the ensemble according to some performance metric [YS06], so that, the accuracy of the system gets increased. Fusion function $\mathcal{F}$ remains the same but there is an additional variable to consider, a weight vector $\mathbf{w} = [w_1, ..., w_M]$, so that the final decision becomes: $d = \mathcal{F}(w_1\mathbf{y}_1, ..., w_M\mathbf{y}_M)$. One common solution (e.g., used in Adaboost [HRZZ09]) is to use the accuracy of each classifier to weight its contribution using:

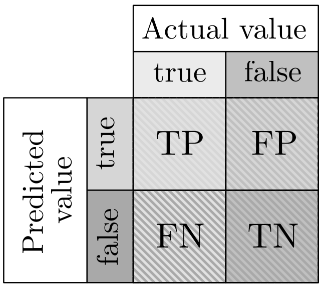
$$D = \arg\max_c \left(\sum_{m=1}^{M} w_m y_{mc} \right), \text{ with: } w_m = 0.5 \log((1 - e_m)/e_m), \tag{2.51}$$

where e_m is the error rate of the m^{th} classifier. It goes without saying that there are many valid methodologies to compute $\mathbf{w}$, ranging from statistical analysis to heuristics.

2.7.10 Performance Metrics

The most common way of measuring the performance of supervised classification is with a confusion matrix (sometimes called error matrix), a table layout that allows visualization of the performance of an algorithm on an specific task. In binary classification, the confusion matrix is is a table with two rows and two columns (shown in Figure 2.14) that assigns every observation to one of four categories according to the correspondence between the predicted class and the true class:

- **True Positives (TP)**: Samples the classifier has correctly classified as positives.
- **True Negatives (TN)**: Samples the classifier has correctly classified as negatives.



		Actual value	
		true	false
Predicted value	true	TP	FP
	false	FN	TN

Figure 2.14: Confusion matrix of binary classification

- **False Positives (FP)**: Samples the classifier has incorrectly classified as positives.
- **False Negatives (FN)**: Samples the classifier has incorrectly classified as negatives.

These metrics are typically expressed as a ratio between the number of 'hits' and the total number of samples. Every observation must be accounted for by one of these four values, so that their sum adds up to 1 (or 100%) of the observations.

- **Accuracy** is the number of correct predictions made by the model over all the predictions made, it can expressed as:

  $$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{2.52}$$

  Accuracy is not a reliable metric for the real performance of some classifiers, because it yields misleading results when the data set is unbalanced (i.e, when the numbers of observations in different classes vary greatly).

- **Error Rate (ER)** is the complementary of accuracy:

  $$\text{ER} = 1 - \text{Accuracy} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{2.53}$$

- **Precision** measures the proportion of correct predictions over those predicted as true:

  $$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.54}$$

- **Recall** or sensitivity, is a measure of the proportion of correct predictions over all those labelled as true:

  $$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.55}$$

- **Specificity** is a measure of the proportion of correct predictions over all those labelled as false:

  $$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{2.56}$$

  Specificity is the exact opposite of Recall.

- **F1-Score** or F1-measure, is a single value that combines precision and recall. It is obtained as the harmonic mean of precision and recall, given by:

  $$F_1 = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.57}$$

> F1-score is often used as an alternative way of measuring the accuracy of the system. However, it is important to note that it does not take the TN into account.

In detection it is common to use the Receiver Operating Characteristic (ROC) curve to illustrate the performance of a binary classifier as a function of its discrimination threshold. The ROC curve is obtained by plotting sensitivity against the probability of false alarm for various threshold values. The probability of false alarm is also known as false-positive rate or fall-out and can be calculated as the complementary of the specificity (1−specificity).

Most of the described metrics can be applied to multiclass classification. Multiclass problems are commonly reduced to C binary classification problems, where C is the number of classes. This is known as “one-vs-all” strategy and it consists on labeling the observations belonging to one class as true and those belonging to the remaining classes as false for each of the C classes. The complete confusion matrix of a multiclass problem is often used to check wether the system is confusing two or more classes (i.e. commonly mislabelling one as another).

PART II Methods and Results

Everything should be as simple as it can be, but no simpler.

Roger Sessions

CHAPTER 3

Self-Localization

3.1 Introduction

Locating the nodes in wireless networks is an essential step for many applications [PAK+05], where the location of the sensors gives meaning to the collected data; thus, self-localization in wireless sensor networks is currently a very active area of research due to an increasing interest on such systems. Regardless of the type(s) of sensor(s), to make networks with a large number of nodes viable, device costs should be as low as possible, nodes need to have a long operating time, and the network needs to auto-manage without significant human intervention. Most traditional node localization techniques are not well suited for these requirements, e.g., having a GPS receiver on each device since obtains limited precisions, only works outdoors and can be cost and energy prohibitive for some applications. The alternative is then to use measurements taken by the nodes to estimate their location collaboratively. The most common measurements are ToA, TDoA, DoA and RSS obtained from acoustic, infrared, or RF signals exchanged between devices (or from the environment) that provide relative timing and/or distance information.

Classic approaches for 'collaborative' node localization rely on beacon nodes (sometimes called anchor nodes), nodes whose coordinates are known a priori to a certain extent and often, they do not have the same restrictions as the "basic" nodes (e.g, a powerful processor and connection to the electrical grid). When a number of beacon nodes is available, the locations of the remaining sensors can be estimated using source localization techniques. However, in ad hoc networks, such as an opportunistic network formed by smartphones, the probability of having beacon nodes is low because of their dynamic nature. Without the beacons, relative locations can be estimated using various methods and establishing an arbitrary coordinate frame of reference, to what we are referring to as self-localization in this thesis.

Wireless networks composed of readily available devices, such as smartphones, have gained particular interest, mainly due to the high availability and low price of such devices. Current generation smartphones pack sufficient hardware so that a group of devices with the correct software can be used for many applications; however, despite their potential, smartphones have several hardware and software limitations that must be considered such as the limited number of available specialized sensors, their limited sampling rate, the lack of optimization of the operative system for real-time applications, and restricted hardware access that makes precise timing of the audio playback and record buffers difficult if not impossible.

In the context of WASNs, acoustic-based self-localization is preferred to RF methods since it is more precise and more cost-effective (especially the passive methods), although the implementation of large distance self-localization is not feasible, since the sources have to be within 'hearing' range of the nodes, typically a few tens of meters. The actual working distance is subjected to many factors such as: the source sound pressure level, the background noise level or the presence of obstructions

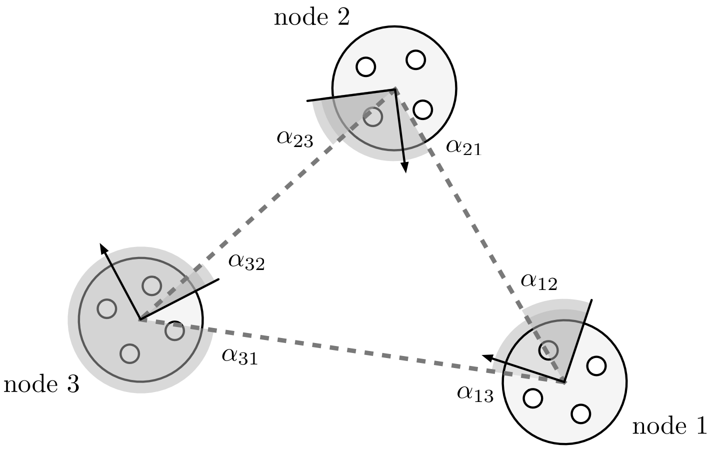


Figure 3.1: DoAs between three wireless nodes.

or strong winds. Generally speaking, the longer the propagation path the lower the SNR and the coherence between signals, thus, the larger the measurement error due to 'poor' cross-correlations.

It is worth noting that self-localization is an ill-posed problem[1], thus, it is often re-formulated before giving it a full numerical analysis, which commonly involves taking some assumptions to fully define the problem and narrow it down.

3.2 Distributed Array Configuration Calibration

The objective of distributed array configuration calibration is to infer the location and orientation of a number of independent small microphone arrays, (i.e., nodes with more than 1 microphone with known local geometry), using DoA measurements. Since each node is equipped with more than one microphone, the location of each of the microphones within a node is given in terms of the node position and orientation combined with a priori knowledge about the sub-array geometry. Figure 3.1 shows a wireless network composed of three nodes and the DoA relations between them.

This family of methods takes advantage of the sub-array local timebase to bypass the need for tight synchronization between nodes, although, DoA-based localization is unable to determine the scale of the network geometry, thus, scale ambiguity needs to be resolved by employing additional measurements. Most methods find first the "constellation" (i.e., scaleless geometry) formed by the nodes by minimizing an error function based solely on DoAs and then apply the information inferred from the additional measurements to properly scale the geometry. Scaling can be resolved either by inter-node measurements such as the TDoA between nodes pairs (requires synchronization) or by local measurements such as RSS.

The proposed solution to self-localization problem is an active method where each node has at least two microphones and a speaker, and works based on DoA and range estimates taken between nodes. DoA estimates are obtained using an efficient cross-correlation technique that assumes the emitted signals are known. Range is estimated from the ToF. Inter node synchronization is not

[1] An ill-posed problem meets one or more of the following criteria: it does not have an exact solution, the solution is not unique, the solution's behavior does not change continuously with the initial conditions.

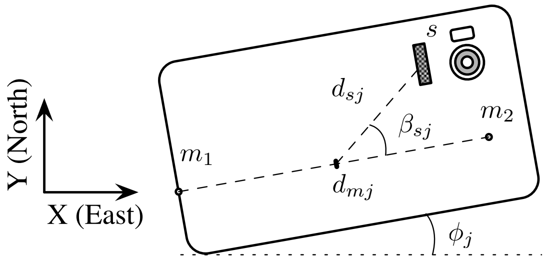


Figure 3.2: Configuration of an smartphone acting as the j^{th} node

necessary, because the clock offset is canceled out by exchanging ToAs. The location and orientation of the nodes in a 2D space are obtained with a closed-form expression, provided that the nodes are with three or more non collinear microphones. The solution is still possible with only two microphones per node, but in that case optimization is required to solve the DoA ambiguity.

3.2.1 Problem Formulation

Let us consider a fully-connected WASN composed of N nodes, where each node contains M microphones forming an array of known geometry. If we have a number of acoustic sources that emit from unknown locations, we can obtain a series of DoA estimates from each node to each source. Now, if we also consider the nodes themselves as sources, we can obtain pairwise DoA estimates between the nodes and since we control the emission, pairwise range estimates also. In the most basic configuration, which is the most common setup found in “off-the-shelf” portable devices such as smartphones, tablets or laptops, each node is equipped with two microphones and a speaker.

Our goal is to find the location and orientation of the N nodes that form the network. We focus on the 2D case, where all nodes lie flat on the XY plane. Note that a 3D generalization will require more than two non collinear microphones per node. We use an active approach by having the nodes be the sound sources, so that we can define a node location as the location of its speaker. Then, the localization problem is reduced to the estimation of $2N$ speaker coordinates $\mathbf{p} = (x_1, \cdots, x_N, y_1, \cdots, y_N)^T$ and N orientations(azimuth) $\phi = (\phi_1, \cdots, \phi_N)$ based on a combination of DoA and range estimates between node pairs. Figure 3.2 represents a typical smartphone configuration that acts as the j^{th} node, where d_{mj} is the distance between the microphone pair (m_{1j} and m_{2j}), d_{sj} and β_{sj} are the distance and angle between the center of the array and the speaker, respectively, and ϕ_i is the orientation of the node.

3.3 Obtention of the estimates

The proposed localization algorithm is based on the combination of $N \times N$ DoA ($\hat{\alpha}_{jk}$) and range ($\hat{r}_{jk}$) estimates, and N orientation ($\hat{\phi}_j$) estimates. Please note that from now onwards we are dropping the hat operator on estimates for simplicity. Each α_{jk}, r_{jk} pair is an estimate of the relative location of the k^{th} speaker with respect to the j^{th} node in polar coordinates, while ϕ_j is an estimate of the orientation of each node inside a common reference frame.

3.3.1 DoA estimation

Let the two microphones of node j be a linear array, so that, if we assume that a source (k^{th} speaker) is in the far field of the array, a plane wavefront impinges it with an angle α_{jk}. The DoA at the microphone pair is reflected as the TDoA between the two sensors, which is given by $\tau_{jk} = f_s d_{mj} \cos(\alpha_{jk})/c$, where d_{mj} is the inter-microphone distance, and c is the speed of sound. Unfortunately, a linear array (1D) in a 2D scenario can only discern DoAs between $-\pi/2$ and $\pi/2$ radians, which leads to DoA ambiguity. For every τ_{jk}, there are two potential DoAs, therefore, the measurement of the angle between node pairs is biased by the node orientation and affected by DoA ambiguity. We can define the angle between node pairs (γ_{jk}) as:

$$\gamma_{jk} = u_{jk}\alpha_{jk} + \phi_j, \tag{3.1}$$

where $u_{jk} = \{-1, 1\}$ is the DoA ambiguity correction variable. Please note that nodes with three or more non collinear microphones are capable of finding DoAs between $-\pi$ and π radians and so DoA ambiguity (in 2D) is not longer a problem [ASMM11]. DoA ambiguity is easily eliminated by the addition of a third microphone which can resolve from which 'side' the sound is impinging in relation to the original microphone pair [KS06]. Without DoA ambiguity the angle between node pairs becomes: $\gamma_{jk} = \alpha_{jk} + \phi_i$.

The classic approach to DoA is to cross-correlate the signal received at each microphone to obtain the time lag between them, or TDoA. Time lags obtained with direct cross-correlation tend to be related to the strongest signal, which is not necessarily the signal of interest. Since we are using an active approach, it is advisable to obtain the time lag between some known reference signal an the signal received at each microphone independently. This method has two main advantages: first, the correlation of the received signals with a known signal removes uncorrelated noise, and second, we ensure that the TDoA estimates originate from the reference signal.

Using this approach, each node emits a reference acoustic signal which is received by every node in the network. Let these reference signals be known and denoted by $s_k(t)$, where k indicates the emitter node. It is then possible to obtain two ToA estimates for every node pair: τ_{1jk} and τ_{2jk} obtained from the respective cross-correlation of microphone signals $m_{1j}(t)$ and $m_{2j}(t)$ and reference signal $s_k(t)$. The TDoA between microphones can be easily computed as the difference between their ToAs: $\tau_{jk} = \tau_{2jk} - \tau_{1jk}$, from which the DoA is directly obtained as: $\alpha_{jk} = \cos^{-1}(\frac{\tau_{jk}c}{f_s d_{mj}})$.

DoA estimates are affected by error. It is clear that failure to obtain the correct TDoA implies a wrong DoA estimate. However, even in the case of finding the best possible TDoA value, the DoA estimate still has some error due to its limited resolution. The range of potential TDoA values is restricted to a finite interval determined by the physical separation between the microphones and the sampling frequency. Furthermore, DoA resolution is not constant over the whole range. Since we are working with discrete signals, the obtained resolution depends on the arc of possible DoAs covered by a discrete TDoA value, which in turn, depends on the sampling frequency (f_s), the distance between microphones (d_{mj}) and the actual impinging angle of the signal. In general, the accuracy of the DoA estimate is better the greater $\frac{d_{mj}c}{f_s}$. Assuming that the DoA of a node pair follows

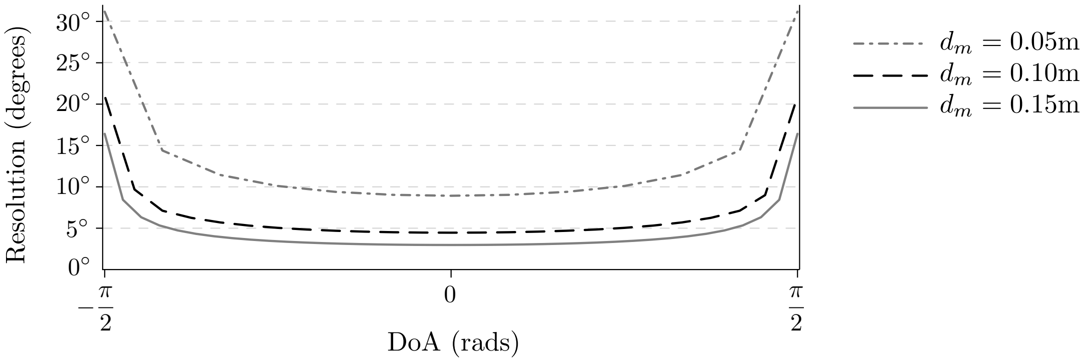


Figure 3.3: Effect of the inter-microphone distance in DoA resolution.

an uniform distribution, the limited resolution introduces and error with a standard deviation (σ_α):

$$\sigma_\alpha = \frac{\left(\alpha(\tau + 0.5) - \alpha(\tau - 0.5)\right)^2}{\sqrt{12}}, \tag{3.2}$$

where $\alpha(\tau)$ represents the DoA obtained for a given TDoA value (in samples). Figure 3.3 shows how the inter-microphone distance affects the accuracy of the estimation in relation to the actual DoA.

3.3.2 Orientation Estimation

Once the DoAs have been obtained, the angle between the j^{th} and k^{th} nodes (γ_{jk}) can be found by adding the j^{th} node orientation (ϕ_j) to α_{jk}. As we will later see, knowing γ_{jk} eases the localization process when compared to its counterpart using α_{jk}. Thus, it is advisable to also obtain the orientation of each of the N nodes within a common reference frame.

The orientation of each node can be easily obtained locally using an accelerometer and a magnetometer (i.e., a digital compass), hardware commonly included in portable devices. However, acceleration and magnetic field measurements are subjected to noise, and so, orientation errors are unavoidable. Noise is especially problematic on the magnetometer; the existence of strong magnetic fields and the presence of electric equipment, and even large metallic objects, affect the magnetic sensor readings making it necessary to calibrate it frequently to avoid large errors [LLW09]. Because the digital compass is uncalibrated more often than not, it introduces a large error (commonly in excess of 15^o) that typically outweighs that of the DoA estimation. Thus, we decided to estimate the orientation of the nodes using the available information instead of relying on an imprecise measurement.

Let us consider that the nodes have their sound source at the center of their microphone array ($d_{sj} = 0$) and that we know the value of the true angle between node pairs φ_{jk} (i.e., the actual value without any error). In this scenario, we know that: $\varphi_{jk} - \varphi_{kj} = \pm\pi$ rad, for $k \neq j$. Now, if we introduce the approximation from (3.1), substitute φ_{jk} with γ_{jk} and substitute the first assumption with $d_{sj} << r_{jk}$ (i.e., the distance between the center of the array and the speaker is much smaller than the distance between the nodes), we arrive to: $\gamma_{jk} - \gamma_{kj} \simeq \pm\pi$, from where the following

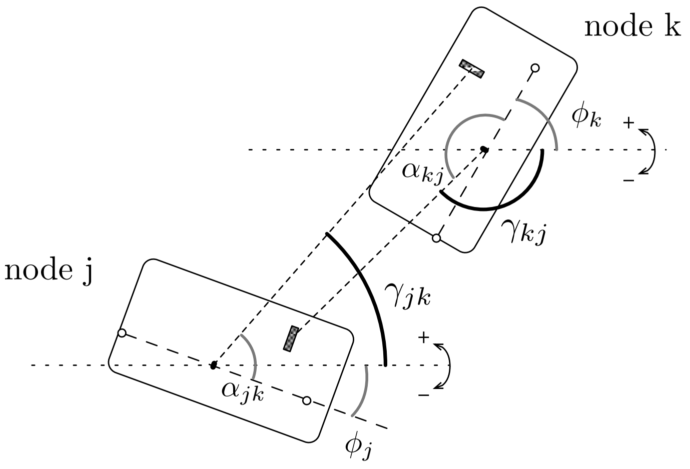


Figure 3.4: Schematic representation of the angular relation between two nodes.

generalization is obtained:

$$u_{jk}\alpha_{jk} + \phi_j - u_{kj}\alpha_{kj} - \phi_k \simeq \begin{cases} \pm\pi, & \text{if} \quad j \neq k \\ 0, & \text{if} \quad j = k \end{cases} \tag{3.3}$$

Figure 3.4 shows the angular relations between node pairs. Notice that when the distance between the nodes is sufficiently large, the error introduced by the speaker not being located at the array center is negligible since it is smaller than the DoA resolution. For the time being, let us drop the DoA ambiguity ($\gamma_{jk} = \alpha_{jk} + \phi_j$), so that taking expression (3.3) into the complex plane, after exponentiation and some operations it becomes:

$$e^{i(\phi_k - \phi_j)} \simeq \begin{cases} e^{i(\alpha_{jk} - \alpha_{kj} - \pi)}, & \text{if} \quad j \neq k \\ 1, & \text{if} \quad k = j \end{cases} \tag{3.4}$$

Now, to estimate the orientations, we can force a relative orientation reference, where $\phi_1 = 0$, arriving to the following expression:

$$e^{-i\phi_j} \simeq \begin{cases} e^{i(\alpha_{j1} - \alpha_{1j} - \pi)}, & \text{if} \quad j \neq 1 \\ 1, & \text{if} \quad j = 1 \end{cases} \tag{3.5}$$

Plugging expression (3.5) into (3.4) we obtain the the final expressions for the orientation estimation:

$$e^{i\phi_k} \simeq \begin{cases} e^{i(\alpha_{jk} - \alpha_{kj} - \alpha_{j1} + \alpha_{1j})}, & \text{if} \quad j \neq k, j \neq 1 \\ e^{i(\alpha_{1k} + \alpha_{k1} - \pi)}, & \text{if} \quad j = 1, k \neq 1 \\ e^{i(\alpha_{1k} + \alpha_{k1} - \pi)}, & \text{if} \quad j = k, k \neq 1 \\ 1, & \text{if} \quad k = 1 \end{cases} \tag{3.6}$$

By combining DoA estimates we have produced $N \times N$ 'partial' orientation estimates, N per node, which are averaged to obtain ϕ_j. This way the quality of ϕ_j is directly related to the error in α_{jk}. With the orientation of the first node fixed at zero, we are establishing a relative coordinate system.

The points in this space are a rotated version of those in a different reference space (e.g, using the magnetic north as a global reference).

When the nodes only have two microphones, each of the DoA estimates is affected by ambiguity (u_{jk}) turning (3.6) into:

$$e^{i\phi_k} \simeq \begin{cases} e^{i(u_{jk}\alpha_{jk}-u_{kj}\alpha_{kj}-u_{j1}\alpha_{j1}+u_{1j}\alpha_{1j})}, & \text{if} \quad j \neq k,\, j \neq 1 \\ e^{i(u_{1k}\alpha_{1k}+u_{k1}\alpha_{k1}-\pi)}, & \text{if} \quad j = 1,\, k \neq 1 \\ e^{i(u_{1k}\alpha_{1k}+u_{k1}\alpha_{k1}-\pi)}, & \text{if} \quad j = k,\, k \neq 1 \\ 1, & \text{if} \quad k = 1 \end{cases} \tag{3.7}$$

In this scenario each ‘partial’ orientation estimate is affected by various ambiguities, and u_{jk} is an unknown that also needs to be estimated. u_{jk} has the particularity of being binary, thus, failure to correctly estimate it, implies a change of sign in α_{jk} which has a significative impact on the orientation estimate. Due to this fact, in the presence of DoA ambiguity, ϕ_k is obtained by taking the 70% trimmed mean of the N available estimates, thus, making the results more robust against outliers created by erroneous u_{jk} values.

3.3.3 Range Estimation and Synchronization

The simplest way of obtaining the range between node pairs is by measuring the ToF (δ_{jk}) of a reference signal. Assuming that the nodes are perfectly synchronized, that is, every node in the network shares a common timebase and has an identical sampling frequency (f_s), the problem of range estimation is reduced to:

$$r_{jk} = \frac{c}{f_s}\Big(\frac{\sum_{m=1}^{M}\tau_{mjk}}{M} - t_k\Big), \tag{3.8}$$

where t_k is the emission time of the k^{th} node reference signal and τ_{mjk} are the ToAs at the M microphones of the j^{th} node. Note that because the nodes have more than one microphone, taking the average of the ToAs gives us the reception time at the center of the array.

Unfortunately, the most likely situation is to lack synchronization between nodes, although it is possible to obtain the ToF by having each node receive its own reference signal, a method sometimes called ‘BeepBeep’ [PSZ12, CPSB+16]. Assuming that clock jitter and drift (differences in f_s between devices) are negligible, the clock offset between nodes can be bypassed by exchanging time references. Let the nodes receive their own reference signal, so that τ_{mj} denotes the ‘self-ToA’ at the m^{th} microphone of the j^{th} node. Defining τ_{jj} as the average τ_{mj}, or the ‘self-ToA’ at the center of the array, the range between a given node pair (r_{jk}), can be obtained by exchanging the ToA of both reference signals at both nodes. The clock offset between the node pair (Δt_{jk}) is cancelled by subtracting the TDoA of both reference signals at both nodes, thus, the range can be estimated as:

$$r_{jk} \approx \frac{c}{2f_s}\big((\bar{\tau}_{jk} - \tau_{jj}) - (\tau_{kk} - \bar{\tau}_{kj})\big), \tag{3.9}$$

where $\bar{\tau}_{jk}$ is average ToA of the k^{th} reference signal at the M microphones of the j^{th} node. Figure 3.5 shows an schematic representation of the time references used to estimate the range between

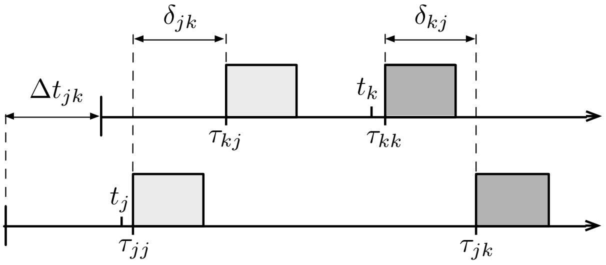


Figure 3.5: Schematic representation of the time references used for range estimation

two nodes.

Note that this method obtains the ToF between the speaker of one node and center of the array of the other node, resulting in two distinct ToFs (see Figure 3.4). In case the speaker is not located at the center of the microphone array, its displacement will introduce an error in the TDoA estimation, which is carried into the r_{jk} estimate. Assuming that the speaker position introduces a random error[2] with a mean value equal to the average of the speaker distances, adding it as an offset r_{jk} becomes:

$$r_{jk} \approx \frac{c}{2f_s}\big((\bar{\tau}_{jk} - \tau_{jj}) - (\tau_{kk} - \bar{\tau}_{kj})\big) + \frac{d_{sj} + d_{sk}}{2}, \tag{3.10}$$

where d_{sj} and d_{sk} are the speaker distance to the center of the j^{th} and k^{th} node respectively. Note that the average of the speaker displacements is added in order to compensate the time it takes for the signal emitted by the nodes to reach their own microphones. On a network formed by N nodes, the standard deviation of the range estimation error (σ_r) can then be approximated by the average of speaker displacements as:

$$\sigma_r \approx \frac{1}{N}\sum_{j=1}^{N} d_{sj} \tag{3.11}$$

Instead of being used to find the ToF between node pairs, this method can be used to establish a common timebase by finding the clock offset between nodes. The main methodology is analogous to the obtention of r_{jk} only instead of pairing the time references to cancel the clock offset, we search to estimate it, bypassing the ToF. This way, the clock offset between a given node pair (Δt_{jk}) can be obtained as:

$$\Delta t_{jk} \approx \frac{(\bar{\tau}_{kj} - \tau_{jj}) - (\tau_{jk} - \bar{\tau}_{kk})}{2}. \tag{3.12}$$

We want to establish a common time base in the network, so we need to find N clock offsets (Δt_j) that cancel the difference in the reception start times among the nodes. This can be done by using the start time of one node as an arbitrary reference for the remaining nodes and combining different Δt_{jk} in relation to it. Let $\Delta t_1 = 0$ so that it becomes the global time reference, defining $\Delta' t_{jk}$ as the clock offset between nodes j and k biased by Δt_1:

$$\Delta' t_{jk} = \begin{cases} \Delta t_{1k} - \Delta t_{jk}\ , & \text{if} \quad j \leq k \\ \Delta t_{1k} + \Delta t_{jk}\ , & \text{if} \quad j > k \end{cases}. \tag{3.13}$$

[2] It should be possible to lessen the impact of the speaker displacement by knowing the angle formed by the nodes, however DoA estimates lack the required resolution.

The clock offset of the j^{th} node can be estimated by averaging its biased pairwise estimates as:

$$\Delta t_j \approx \frac{1}{N-1} \sum_{\substack{k=1 \\ k \neq j}}^{N} \Delta t'_{jk}, \; j > 1. \tag{3.14}$$

Again, the displacement of the speaker from the center of the microphone array introduces an error in the estimation. However since we are averaging $N-1$ estimates, the error gets smaller for larger values of N. The maximum error of the estimation (when the time estimates are correct to the sample) is approximately $\pm\frac{f_s}{cM}\sum_{j=1}^{N} d_{sj}$, which is good enough for many applications but might be to restricting for tasks requiring sub-sample level synchronization such as beamforming.

This method of range and clock offset estimation is robust to a lack of control over the emission time. An implementation on an operative system not optimized for real-time applications will still work as long as the recording starts before the emission. It is important to highlight again, that accurate timing is a must for audio-based spatial applications: a timing error of just 1ms would translate in a range estimation error over 0.3m. In relation to this, the clock drift between nodes also introduces an error that we have not considered. Time drift is inherent to every audio recording device, resulting from the effects of temperature and random errors in oscillators. A typical sound recording system can present time drifts of around 30ms per hour recorded between two identical devices [GLB15]. Clock drift should be taken into account, specially for the validity of the synchronization over long periods of time.

3.4 Maximum Likelihood Estimator of Node Locations

This section describes how the proposed self-localization method combines the angle and range estimates taken by the nodes in order to obtain their locations.

Let us consider that a full set of estimations of the range r_{jk} and incidence angle γ_{jk} from the k^{th} node to the j^{th} node are available, and that each of these estimates has an associated standard deviation error, $\sigma_r(j,k)$ and $\sigma_\gamma(j,k)$ respectively. The objective of the self-localization process is to estimate the node position vector $\mathbf{p} = [\mathbf{p}_x, \mathbf{p}_y]^T$ from the estimate pairs taking into account the standard deviation of each estimate.

Defining each estimate pair (γ_{jk}, r_{jk}) as a vector in polar coordinates that describes the location of the k^{th} node in relation to the j^{th} node, the vector in Cartesian coordinates beacomes: $\mathbf{d}_{jk} = [v_{jk}, w_{jk}]$, where $v_{jk} = r_{jk}\cos(\gamma_{jk})$, and $w_{jk} = r_{jk}\sin(\gamma_{jk})$, with $r_{jk} \geq 0$ and $\gamma_{jk} \in [-\pi, \pi]$ for all j and k from 1 to N. Let us also consider the joint PDF of the estimates in cartesian coordinates as a multivariate normal distribution, given by

$$\mathcal{F}_{jk}(\mathbf{p}) = \frac{1}{2\pi|\mathbf{C}_{jk}|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(\mathbf{p}_j - \mathbf{p}_k - \mathbf{d}_{jk})\mathbf{C}_{jk}^{-1}(\mathbf{p}_j - \mathbf{p}_k - \mathbf{d}_{jk})^T\right)}, \tag{3.15}$$

where $\mathbf{C}_{jk}$ is the covariance matrix of the PDF related to the vector that the j^{th} node estimates to the k^{th} node and $\mathbf{p}_k$ is a column vector containing the two components of the position of the k^{th} node.

In simpler terms, this can be seen as node j providing the relative location of node k by pointing

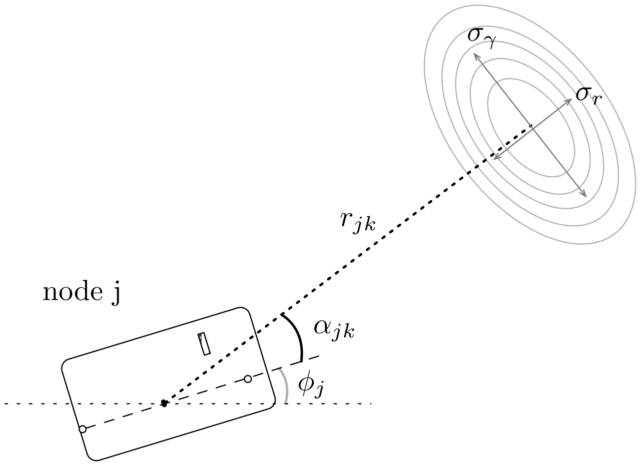


Figure 3.6: Relative location of the k^{th} node obtained by the j^{th} node.

towards a region of probable locations in the cartesian space, that is positioned by $\mathbf{d}_{jk}$ and delimited by $\sigma_r(j,k)$ and $\sigma_\gamma(j,k)$ as shown in Figure 3.6.

If we combine all the pairwise relative locations it is possible to obtain the most probable node locations inside a common reference frame. Using a maximum likelihood estimator, the log-likelihood $\mathcal{L}$ of a given geometry is calculated as:

$$\mathcal{L} = \sum_{j=1}^{N} \sum_{\substack{k=1 \\ k \neq j}}^{N} \log(f_{jk}(\mathbf{p})). \tag{3.16}$$

Replacing equation (3.15) in (3.16) and simplifying, the next expression is obtained:

$$\mathcal{L} = b - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (\mathbf{p}_j - \mathbf{p}_k - \mathbf{d}_{jk}) \mathbf{D}_{jk} (\mathbf{p}_j - \mathbf{p}_k - \mathbf{d}_{jk})^T, \tag{3.17}$$

where $b = -\log(2\pi |\mathbf{C}_{jk}|^{\frac{1}{2}})$, and $\mathbf{D}_{jk} = \mathbf{C}_{jk}^{-1}$ for $j \neq k$, and a zero matrix for $j = k$:

$$\mathbf{D}_{jk} = \begin{pmatrix} \rho_{xx}(j,k) & \rho_{xy}(j,k) \\ \rho_{yx}(j,k) & \rho_{yy}(j,k) \end{pmatrix}, \forall\, j \neq l. \tag{3.18}$$

In order to maximize the log-likelihood function, expression (3.17) has to be differentiated with respect to each term of vector $\mathbf{p} = [x_1, \cdots, x_N, y_1, \cdots, y_N]^T$. After equaling each resulting expression to zero, a system of $2N$ linear equations is obtained. This system of equations can be written in matrix notation according to:

$$\begin{pmatrix} \text{diag}(\mathbf{R}_{xx}\mathbf{n}) - \mathbf{R}_{xx} & \text{diag}(\mathbf{R}_{xy}\mathbf{n}) - \mathbf{R}_{xy} \\ \text{diag}(\mathbf{R}_{yx}\mathbf{n}) - \mathbf{R}_{yx} & \text{diag}(\mathbf{R}_{yy}\mathbf{n}) - \mathbf{R}_{yy} \end{pmatrix} \cdot \mathbf{p} = \mathbf{s}, \tag{3.19}$$

where $\mathbf{n}$ is a column vector of length N of all ones, diag($\mathbf{x}$) is a diagonal matrix with the elements of vector $\mathbf{x}$ on its main diagonal, and $\mathbf{R}_{xy} = \mathbf{H}_{xy} + \mathbf{H}_{xy}^T$. Matrix $\mathbf{H}_{xy}$ is a $N \times N$ square matrix containing the corresponding x-y terms of $\mathbf{D}_{jk}$, given by:

$$\mathbf{H}_{xy} = \begin{pmatrix} 0, & \rho_{xy}(1,2), & \cdots, & \rho_{xy}(1,N) \\ \rho_{xy}(2,1), & 0 & \cdots, & \rho_{xy}(2,N) \\ \vdots & \vdots & & \vdots \\ \rho_{xy}(N,1), & \rho_{xy}(N,2) & \cdots, & 0 \end{pmatrix} \tag{3.20}$$

Finally, the independent term of the system of equations (s) can be determined using the following expression:

$$\mathbf{s} = \begin{pmatrix} \sum_{j=1}^{N} \rho_{xx}(1,j)v_{1j} - \rho_{xx}(j,1)v_{j1} + \cdots + \rho_{xy}(1,j)w_{1j} - \rho_{xy}(j,1)w_{j1} \\ \vdots \\ \sum_{j=1}^{N} \rho_{xx}(N,j)v_{Nj} - \rho_{xx}(j,N)v_{jN} + \cdots + \rho_{xy}(N,j)w_{Nj} - \rho_{xy}(j,N)w_{jN} \\ \sum_{j=1}^{N} \rho_{yx}(1,j)v_{1j} - \rho_{yx}(j,1)v_{j1} + \cdots + \rho_{yy}(1,j)w_{1j} - \rho_{xy}(j,1)w_{j1} \\ \vdots \\ \sum_{j=1}^{N} \rho_{yx}(N,j)v_{Nj} - \rho_{yx}(j,N)v_{jN} + \cdots + \rho_{yy}(N,j)w_{Nj} - \rho_{xy}(j,N)w_{jN} \end{pmatrix} \tag{3.21}$$

Unfortunately, the system of equations described by (3.19) can not be solved directly. There are $2N$ equations and the rank of the coefficient matrix is $2N-2$, making it an underdetermined system of equations. In order to solve it, two additional constrains have to be added. The simpler solution is to to place the center of mass of the nodes in the origin of coordinates, so that $\sum_{j=1}^{N} x_j = 0$ and $\sum_{j=1}^{N} y_j = 0$. These two equations can be added to the system of equations given in (3.19), allowing to solve the self-localization problem by means of the inversion of a $2N \times 2N$ matrix.

The proposed method provides an analytical solution to the self-localization problem, but it requires knowledge about the covariance matrices of the estimates ($\mathbf{C}_{jk}$), and its solution requires in the order of $\mathcal{O}((2N)^3)$ operations. For certain applications, the additional precision obtained by taking into consideration the standard deviation of the individual estimates might not be desired, or perhaps, the computational load associated with the solution is too large. For this reason we propose two solutions: first, a simple version that assumes all the covariance matrixes equal and proportional to the identity matrix, which we call Naive Covariance Matrix Estimation (NCME), and second, a solution in which the covariance matrixes are estimated taking into account the errors of both DoA and range estimates, or Full Covariance Matrix Estimation (FCME).

3.4.1 Naive Covariance Matrix Estimation

On a first approach, we suppose that all the covariance matrices are equal and proportional to the identity matrix (I), so that $\mathbf{D}_{jk} = \rho\mathbf{I}$, with $\rho = 0$ when $j = k$. This is equivalent to assume that the variables of the PDF are independent and their standard deviation is constant. In this scenario $\sigma_r(j,k)$ and $\sigma_\gamma(j,k)$ are not longer considered, thus, every estimation has the same weight.

Under this conditions, $\mathbf{H}_{xx} = \mathbf{H}_{yy} = \rho(\mathbf{N} - \mathbf{I})$, $\mathbf{N}$ being a $N \times N$ matrix of all ones, and

$\mathbf{H}_{xy} = \mathbf{H}_{yx} = 0$. After this simplification, the system of linear equations given in (3.19) can now be expressed as:

$$\begin{pmatrix} N\mathbf{I} - \mathbf{N} & 0 \\ 0 & N\mathbf{I} - \mathbf{N} \end{pmatrix} \cdot \mathbf{p} = \begin{pmatrix} \frac{1}{2}\sum\limits_{j=1}^{N} v_{1j} - v_{j1} \\ \vdots \\ \frac{1}{2}\sum\limits_{j=1}^{N} v_{Nj} - v_{jN} \\ \frac{1}{2}\sum\limits_{j=1}^{N} w_{1j} - w_{j1} \\ \vdots \\ \frac{1}{2}\sum\limits_{j=1}^{N} w_{Nj} - w_{jN} \end{pmatrix}, \tag{3.22}$$

where $\mathbf{I}$ is a $N \times N$ identity matrix. Finally we need two add two constraints to lower the rank. Adding the constraint $\sum_{j=1}^{N} x_j = 0$ to the first N equations (x coordinates), and the constrain $\sum_{j=1}^{N} y_j = 0$ to the last N equations (y coordinates), the system of equations becomes now trivial, its solution being given by:

$$\mathbf{p} = \begin{pmatrix} \frac{1}{2N}\sum\limits_{j=1}^{N} v_{1j} - v_{j1} \\ \vdots \\ \frac{1}{2N}\sum\limits_{j=1}^{N} v_{Nj} - v_{jN} \\ \frac{1}{2N}\sum\limits_{j=1}^{N} w_{1j} - w_{j1} \\ \vdots \\ \frac{1}{2N}\sum\limits_{j=1}^{N} w_{Nj} - w_{jN} \end{pmatrix} \tag{3.23}$$

Since every estimate has the same weight and we have place the origin of coordinates at the center of the constellation, the location of the nodes can be found by simply averaging the relative position estimates obtained by each node. Taking the original estimates into expression (3.23), the location of the j^{th} node, is obtained as:

$$\mathbf{p}_k = \frac{1}{2N}\Big[\sum_{j=1}^{N}\big(r_{jk}\cos(\gamma_{jk}) - r_{kj}\cos(\gamma_{kj})\big), \sum_{j=1}^{N}\big(r_{jk}\sin(\gamma_{jk}) - r_{kj}\sin(\gamma_{kj})\big)\Big]^T. \tag{3.24}$$

This simple method of self-localization is made possible by the existence of a common orientation reference (i.e., known ϕ_j). Without knowledge about the node rotations, the relative location estimations of a given each would reside on a rotated space, making direct averaging averaging not feasible.

The NCME method estimates the position of the nodes very fast, so it is of special interest when the computational cost is relevant. Due to this fact, the NMCE method is of special interest in solving the ambiguity problem, as it will be described later.

3.4.2 Full Covariance Matrix Estimation

It is possible to improve the precision of the self-localization by taking into consideration the error in each of the estimates, although, the resolution of the equation system requires the covariance matrices to be known.

Let us consider that every DoA and range estimate has has the same error, so that subindex jk associated to said errors can be dropped for clarity. In order to determine an analytical expression for the covariance matrix $\mathbf{C}_{jk}$, the polar to cartesian transformation can be carried out using an approximation of the non-linear function at the estimated point. Let us consider the transformation functions $x = (r_{jk} + \Delta r)\cos(\gamma_{jk} + \Delta\gamma)$ and $y = (r_{jk} + \Delta r)\sin(\gamma_{jk} + \Delta\gamma)$, where $E\{\Delta r\} = E\{\Delta\gamma\} = 0$, $E\{f(\Delta r)g(\Delta\gamma)\} = E\{f(\Delta r)\}E\{g(\Delta\gamma)\}$, and $E\{\Delta r^2\} = \sigma_r^2$, and $E\{\Delta\gamma^2\} = \sigma_\gamma^2$. Then, $E\{x\} = r_{jk}\cos(\gamma_{jk})$ and $E\{y\} = r_{jk}\sin(\gamma_{jk})$. Applying trigonometric properties, the following approximations are obtained:

$$x = (r_{jk} + \Delta r) \cdot (\cos(\gamma_{jk})\cos(\Delta\gamma) - \sin(\gamma_{jk})\sin(\Delta\gamma)), \tag{3.25}$$

$$y = (r_{jk} + \Delta r) \cdot (\sin(\gamma_{jk})\cos(\Delta\gamma) + \cos(\gamma_{jk})\sin(\Delta\gamma)), \tag{3.26}$$

where $\Delta\gamma$ and Δr represent the error of a DoA and range estimation respectively. Considering that the value of $\Delta\gamma$ is small (i.e. the DoA is correctly estimated), it is possible to take the approximations: $\sin(\Delta\gamma) \simeq \Delta\gamma$ and $\cos(\Delta\gamma) \simeq 1$. In order to estimate the terms of the covariance matrix, we must determine its values one by one, so that: $\sigma_{xx}^2(j,k) = E\{x^2\} - E\{x\}^2$, $\sigma_{yy}^2(j,k) = E\{y^2\} - E\{y\}^2$, and $\sigma_{xy}^2(j,k) = \sigma_{yx}^2(j,k) = E\{xy\} - E\{x\}E\{y\}$. After evaluating these terms, covariance matrix can be estimated as:

$$\mathbf{C}_{jk} \simeq \mathbf{R}_{jk} \cdot \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_a^2(r_{jk}^2 + \sigma_r^2) \end{pmatrix} \cdot \mathbf{R}_{jk}^T, \tag{3.27}$$

where $\mathbf{R}_{jk}$ is a rotation matrix, given by:

$$\mathbf{R}_{jk} = \begin{pmatrix} \cos(\gamma_{jk}) & -\sin(\gamma_{jk}) \\ \sin(\gamma_{jk}) & \cos(\gamma_{jk}) \end{pmatrix}. \tag{3.28}$$

Taking into account that $\mathbf{R}_{jk}^{-1} = \mathbf{R}_{jk}^T$, the inverse of the covariance matrix ($\mathbf{D}_{jk}$), whose elements where unknown in (3.18), can easily be obtained using the following expression:

$$\mathbf{D}_{jk} = \mathbf{C}_{jk}^{-1} \simeq \mathbf{R}_{jk}^T \cdot \begin{pmatrix} \frac{1}{\sigma_r^2} & 0 \\ 0 & \frac{1}{\sigma_\gamma^2(r_{jk}^2 + \sigma_r^2)} \end{pmatrix} \cdot \mathbf{R}_{jk}. \tag{3.29}$$

This is an interesting way of estimating $\mathbf{D}_{jk}$ as a function of the values of the pairwise estimates, r_{jk} and γ_{jk}, and the variances of the errors associated to these estimations: σ_r^2 and σ_γ^2. Please note that σ_r^2 and σ_γ^2 have different effect on the weight applied to a given estimate, the latter being modulated by the range. This can be explained by seeing the effect of the angle estimate on the relative position estimate as a circumference arc, whose length increases as a function of the radius. This way, the zone of probable locations of the k^{th} node in relation to the j^{th} node will get larger as a function of the distance between them (r_{jk}) for any given value of σ_γ^2. It is also important to

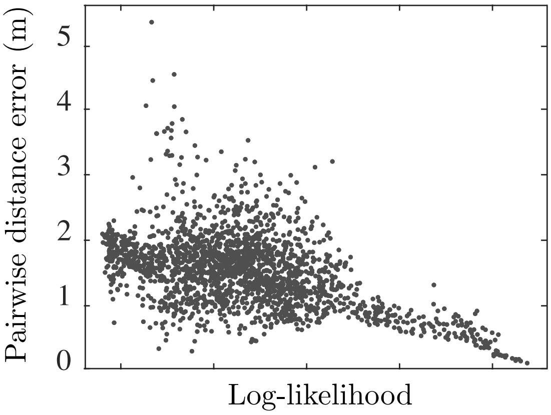


Figure 3.7: Relation between the log-likelihood and the pairwise distance error for all possible values of $\mathbf{U}$ with $N = 4$

highlight that the errors can either be considered constant for all the estimates, or alternatively, they can be modeled or estimated for every pairwise estimation. On previous sections describing how the DoA and range estimates are obtained in this proposal, there were some mention of the standard deviations of said estimates. In this particular case, we decided to keep σ_r^2 constant, while modeling $\sigma_\gamma^2(j,k)$ as a function of the resolution of each individual α_{jk}.

3.5 DoA Ambiguity Solution

Up until this point, we have considered two different scenarios: either the DoA estimates are affected by ambiguity or not. The proposed solution to the self-localization problem requires the angle measurements between nodes to be known. For this to be feasible, we first need to estimate each node orientation (ϕ_j) within a common reference frame. However, expression (3.7) shows that the orientation estimation is affected by ambiguity, which in turn is carried into the subsequent node location estimates. This posses a new problem: when the nodes are equipped with just two microphones, the whole self-localization problem depends on an unknown DoA ambiguity correction matrix $\mathbf{U}$ composed of $N \times N$ u_{jk} values.

Solving the DoA ambiguity of an isolated microphone pair is a very hard problem that requires additional knowledge and specialized techniques[3], however, since we are obtaining pairwise DoA estimates between a number of microphones pairs randomly located within a 2D space, it is possible to jointly solve the pairwise ambiguities by setting some fitness function and minimizing it. We have found a clear relation between the log-likelihood obtained for a certain $\mathbf{U}$ and the self-localization error. Thus, we propose to use expression (3.17) as the fitness function. Figure 3.7 shows the relation between the log-likelihood (using the NCME method) and the pairwise node distance error for all possible values of $\mathbf{U}$ in a network with $N = 4$.

Since the node locations ($\mathbf{p}$) for a given $\mathbf{U}$ are obtained using a closed-form expression, $\mathbf{U}$ can be seen as the variable in an optimization problem, on which, $\mathbf{p}$ is obtained as a byproduct. The ambiguity correction is a binary variable ($u_{jk} = \{-1, 1\}$), thus, there are 2^{N^2} unique $\mathbf{U}$ values.

[3] On an ideal scenario it is possible to solve DoA ambiguity by having directional microphones pointing in opposite directions perpendicular to the line formed by them and comparing the energy of the received signals.

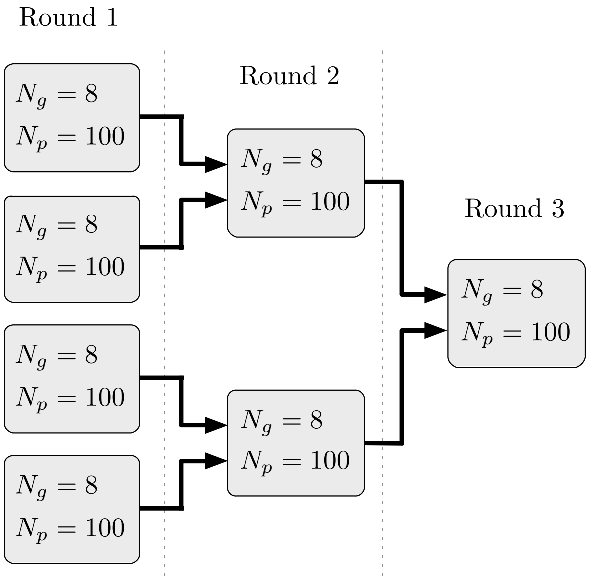


Figure 3.8: Example of elimination tournament with $N_r = 3$ rounds, $N_g = 8$ generations per round, and $N_p = 100$ individuals per stage of the tournament.

Taking into consideration that the main diagonal of $\mathbf{U}$ is of no interest (the case when $j = k$) and does not need to be estimated, the maximum number of combinations is reduced to $2^{N(N-1)}$. The number of unique $\mathbf{U}$ values, grows exponentially with N, making it unfeasible to test every single value. Thus, we decided to use a Genetic Algorithm (GA) to find the solution.

To improve the convergency of the optimization algorithm with respect to the total number of performance evaluations, instead of using a single GA and several runs (i.e., the standard scheme), we use an elimination tournament of small GAs. We start with a set of 64 small GAs (each GA denoted as and stage of the tournament) with a population of $N_p = 10N$ individuals and $N_g = N$ generations each. The best solutions of the first round are then paired, generating a new population for every two winners, which are set to compete in the next round. The process is repeated until a global winner is obtained. For illustrative purposes, Figure 3.8 shows an example of the elimination tournament with a total of $N_r = 3$ rounds. In our case, we used $N_r = 7$ rounds, since it empirically gave us good convergence results.

The GA algorithm used at each each stage is implemented as described in section 2.4.2. Population size is $N_p = 10N$ individuals. Each individual ($\mathbf{U}_p$) contains $N(N-1)$ genes corresponding to u_{jk}. On the first round of the tournament the genes are randomly selected, for every subsequent round, they are created by reproduction and mutation from the previous stage winners. The fitness function is computed with (3.17) using the NCME method. Each generation, the top performing 10% individuals are selected to breed a new population and the remaining 90% of the population is discarded. The full population is mutated by selecting 1% of their genes at random and inverting their value, except for the best performer. Since the probability of a change in u_{jk} involving a change in u_{kj} is very high, 75% of the mutations produce a change of sign in both genes. After N_g generations, the best $\mathbf{U}_p$ is selected as a candidate and is set to compete in the next round of the tournament. After the GA tournament is completed, the best individual becomes $\mathbf{U}$ and is used to estimate the final node positions and orientations.

It is important to highlight that, while the the computational cost of the optimization algorithm is quite high, the parallelization of the elimination tournament is trivial, and the various small GAs can be easily divided by the total number of nodes of the network. In a rough approximation, taking the computation time of the closed-form expressions of the ML estimator and the orientation estimation as a single operation, in Big O notation, the parallelized tournament has a complexity $\mathcal{O}(N)$. The tournament is composed of 127 GAs divided among N nodes. In the worst scenario a node has to take care of $N_{ga} = \lceil 127/N \rceil$ GAs. Each GA performs N iterations with a population size of $10N$, so in total, each node needs to compute $\lceil 127/N \rceil \times N^2$ operations. In average, the computational load of the optimization algorithm (for one node) is around $1300J$ times higher than that of the estimations using the NCME closed-form expression.

Please note that the need for an iterative algorithm is a direct consequence of the DoA ambiguity. Provided that each node was capable of resolving 360^o DoAs (by having 3 or more microphones arranged in a 2D array), the solution to the self-localization problem can be found directly using the analytical expression presented in the previous section.

3.6 Experimental work and Results

This section describes the experiments conducted to evaluate the performance of the self-localization algorithm proposed in this chapter, as well as a discussion of the results obtained. We decided to evaluate the algorithm using just two microphones per node for three main reasons: first, it is the most common configuration of “off-the-shelf” smartphones and other portable devices; second; to prove the feasibility of performing distributed array configuration calibration in the presence of DoA ambiguity; and third; to show that even in the presence of DoA ambiguity it is preferable to estimate the orientations of the nodes rather than using an uncalibrated digital compass.

3.6.1 Description of the Experiments

To evaluate the proposed algorithm, we generated a realistic database of acoustic signals, which contains 300 different scenarios that include both reverberation and background noise. Each scenario contains 10 nodes (equipped with two microphones and a speaker) at random locations and with random orientations and is defined by a random a combination of the next parameters: room dimensions of 6-12 m long/wide and 2-3 m high; absorption coefficient of 0.5-1; SNR of 5-20 dB. Background noise was added as additive white noise (simulated diffuse noise field) with an energy level controlled by the SNR. The positions of the nodes were restricted as if they were lying on a table of dimensions 5x2 m (a large-sized conference table) with a minimum distance between nodes of 15 cm. The internal geometry of the nodes is modeled after a BQ Aquaris 4 smartphone, whose physical characteristics are the following: $d_m = 0.106$m, $d_s = 0.096$m, and $\beta_s = 5.9°$. The acoustic signals received by the microphones of each node were generated using a room impulse response generator, which was computed using the simple image method described in Allen and Berkley [AB79] at a sampling frequency of 44100 Hz. The method considers the dimensions of the room, the absorption coefficient of the walls, the location of the sources and the location, directivity pattern and frequency response of the microphones. The frequency response of the microphones has been experimentally measured in an anechoic chamber for various angles over a whole sphere (360°). The obtained directivity patterns can be approximated by a cardioid pattern with a front-to-back ratio of 10 dB[4]. In order to generate the database we assume that every devices lies flat on a table with its back pointing upwards. This assumption affects the directivity of the microphones in the following way: the directivity of the first microphone depends on the orientation of the device (it points toward ϕ) and the main lobe of the second microphone points upward, so it is considered omnidirectional.

The acoustic reference signal emitted by the nodes is a band-limited white noise signal (500 Hz - 16 kHz). The time duration of the reference signals is a tradeoff between computational complexity and robustness against low SNR, hence, in order to see the effect of the cross-correlation of the self-localization results, we tested 3 different reference signal lengths. The selected lengths are $L_R = \{8192, 4096, 2048\}$ samples, or 18.58, 9.29, and 4.65 ms respectively at $f_s = 44100$ Hz. Each device emits a unique reference signal, which is known by every node in the network. The reference signals were obtained from normally distributed random sequences filtered in the aforementioned bandwidth. The sequences were generated with $L_R = 8192$ and were cropped to obtain the shorter reference signals. The selected frequency bandwidth is related to the frequency response of the integrated microphones as well as the typical frequency response of smartphone speakers. The time lag between the received signals and the reference signals was obtained using framed cross-correlation with a frame length $L = 2L_R$ and PHAT weighting for robustness against reverberation. The conducted simulations show that less than 4% of the DoAs are missed when using the selected parameters with 4096 samples long band-limited white noise reference signals. A chirp signal with the same frequency range needs to be double the length (8192 samples) to achieve comparable results under the same conditions, which implies a higher computational cost. In addition to this, we have conducted an informal listening test where the noise signal was tagged as less disturbing than a chirp signal by most of the participants. Note that using short reference signals also has the benefit of making the localization process less disturbing to users who are exposed to them.

The pairwise estimates between nodes are obtained using the methods described in section 3.3.1 for the DoAs, and section 3.3.3 for the range. Regarding the orientation we have evaluated the performance of the algorithm under two conditions, assuming that the node orientations in relation to the magnetic north are obtained with a digital compass; and estimating the orientations as described in section 3.3.2, thus setting an arbitrary orientation reference. DoA ambiguity is solved as described in section 3.5 using the NCME method as the fitness function. When the orientations are obtained with a digital compass, we have considered the ideal scenario, in which the orientations are measured without any error ($\sigma_\phi = 0°$), and a more realistic scenario, in which the typical error of a digital compass is considered ($\sigma_\phi = 15°$). The final node location estimates are obtained using the FCME method in every case.

The last consideration is the coordinate system. We have previously mentioned that the origin of coordinates was set at the center of mass of the node locations in the localization process; however, without loss of generality, we can assume that the first node is located at the origin of the coordinates. Then, the transposed locations are found by subtracting the coordinates of the first node. Hence,

[4] The microphones are rated as omnidirectional, the cardioid directivity is a result of the 'shadowing' effect produced by the body of the smartphone.

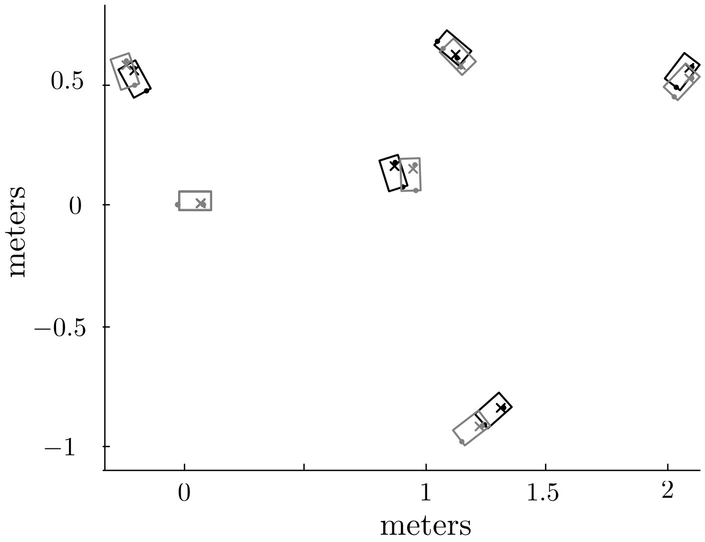


Figure 3.9: Localization results (transposed and rotated) for one example in the database with $N = 6$. The true positions are in black, and estimates are in grey.

together with the condition set for the orientation estimation, the localization results are provided in relation to the first node. With a localization example in Figure 3.9, we observe that when this reference system is used, the estimated and true locations of the first node are identical, thus it is not considered for computing the results. Regardless of where the origin of coordinates is set, the location estimates produced by the algorithm reside on a relative coordinate system. The points in this space are translated (and rotated in case the orientation of the nodes is also estimated); it suffices to know the actual position and orientation of one of the nodes (i.e., having a beacon node) to transform the results to a global coordinate system.

In order to set a comparison with the proposed method we have implemented 2 of the self-localization methods available on the literature, namely: Jacob et al. [JSHU12], that was selected to represent DoA-based distributed array configuration calibration; and Crocco et al. [CDBBM12], that was selected to represent range-based distributed microphone configuration. All the tested methods work with the same set of estimates, and under the same conditions. The differences between their performance are limited to the specific methodologies by which the various estimates are combined in order to find the node locations.

The method presented in Jacob et al. [JSHU12] is based on DoA estimates alone. The method originally uses 4 sub-arrays with 2 microphones, assuming that the sub-arrays are located along the walls of a room of known dimensions (i.e., sound never impinges from the back if the array). However, we have adapted the described method for the use of range estimates, so that, the solution is scaled to minimize the difference with the measured range values as described in Schmalenstroeer et al. [SJHU$^+$11]. It is important to highlight that this method only works without DoA ambiguity and so, in order to obtain the results we assumed that the nodes were capable of measuring 360° DoAs using only 2 microphones which is physically impossible.

The method described in Crocco et al. [CDBBM12] is closely related to MDS, it describes a closed-form solution for the self-localization problem. This method is intended for nodes with a single microphone, so it is based on range measurements alone. Note that we obtain the range

Table 3.1: Localization error (centimeters) for different algorithms and network sizes (N). $L_R = 4096$. DoA estimates with ambiguity have a range of 180° instead of 360°.

		Crocco et al. [CDBBM12]			Jacob et al. [JSHU12]			**Proposed**		
		Known orientations*			Without ambiguity			With ambiguity		
length	N	Mean	Std	Trim	Mean	Std	Trim	Mean	Std	Trim
$2L_R$	4	13.0	10.7	10.8	18.5	35.9	12.3	10.3	9.6	8.5
	5	13.9	12.3	11.2	16.3	23.1	11.7	10.6	12.3	8.5
	6	14.5	14.2	11.4	15.2	22.2	11.4	10.1	11.0	8.4
	7	14.5	13.1	11.6	14.2	19.8	10.6	10.2	10.3	8.3
	8	14.7	13.0	11.6	13.5	14.7	10.7	9.8	8.3	8.2
	9	14.6	12.4	11.7	14.5	21.0	10.7	10.9	11.8	8.3
	10	14.8	12.6	11.8	13.4	13.7	10.5	11.5	11.0	8.8
L_R	4	13.3	11.2	10.8	17.3	25.6	12.4	10.1	8.5	8.5
	5	14.1	12.7	11.2	15.1	16.1	11.5	10.2	9.1	8.5
	6	14.6	14.7	11.4	14.3	14.9	11.2	10.1	9.6	8.4
	7	14.6	13.4	11.6	13.9	15.2	10.8	10.0	8.5	8.3
	8	14.7	13.5	11.7	15.4	30.6	11.0	10.1	9.8	8.2
	9	14.9	13.5	11.7	14.0	18.0	10.7	10.5	10.2	8.3
	10	14.9	13.0	11.8	13.4	13.3	10.6	11.4	10.8	8.7
$\frac{L_R}{2}$	4	28.7	40.4	17.0	27.6	58.7	12.7	14.8	19.3	9.2
	5	33.7	41.7	22.9	23.9	43.2	12.7	14.9	18.1	9.8
	6	41.1	45.4	31.2	27.6	57.7	12.8	15.2	16.2	10.7
	7	42.6	43.7	33.5	27.0	42.1	14.2	14.4	12.4	11.2
	8	43.3	42.4	34.6	28.5	40.1	17.0	13.9	11.8	11.0
	9	43.9	40.1	36.5	31.0	41.2	20.1	14.6	12.4	11.6
	10	46.6	40.0	40.4	35.7	41.4	26.3	15.9	12.9	12.8

$^{*}\sigma_\phi = 0°$

estimates by averaging the ToAs at both microphones, thus, this method is not capable of obtaining orientation estimates. In case the ToAs were obtained at each microphone it should be possible to also estimate the orientations by adding some constraints (known distance between microphones inside a given node), although in [CDBBM12] this is not considered. In order to obtain the presented results we assummend that node orientations were obtained with a digital compass.

It is important to mention that, none of these methods are capable of discerning between reflected solutions, therefore, in order to obtain the results we considered all the possible reflections.

3.6.2 Experimental Results

Table 3.1 shows the Mean, the standard deviation (Std) and the 25% trimmed mean (Trim) of the localization error obtained with the proposed algorithm and those obtained with Jacob et al. [JSHU12] and Crocco et al. [CDBBM12]. Please note that, Crocco et al. [CDBBM12] is using perfectly known orientations ($\sigma_\phi = 0°$), and that Jacob et al. [JSHU12] its using 360° DoA estimates, while the presented method works with 180° DoA estimates. Among the tested self-localization

Table 3.2: Localization error (centimeters) for the proposed algorithm with known and estimated orientations for different network sizes. $L_R = 40196$.

		Proposed ($\sigma_\phi = 0°$) Known orientations			Proposed ($\sigma_\phi = 15°$) Known orientations.			Proposed Estimated orientations.		
length	N	Mean	Std	Trim	Mean	Std	Trim	Mean	Std	Trim
$2L_R$	4	9.1	7.1	8.0	22.1	13.4	19.7	10.3	9.6	8.5
	5	9.1	7.1	7.9	21.9	12.2	20.0	10.6	12.3	8.5
	6	9.2	7.8	7.9	21.9	11.2	20.3	10.1	11.0	8.4
	7	9.2	8.0	7.8	21.3	9.9	19.9	10.2	10.3	8.3
	8	9.1	7.9	7.8	21.0	9.8	19.6	9.8	8.3	8.2
	9	8.9	6.2	7.8	20.6	8.4	19.5	10.9	11.8	8.3
	10	8.7	5.3	7.7	20.2	7.9	19.3	11.5	11.0	8.8
L_R	4	9.0	6.1	8.0	22.1	13.3	19.6	10.1	8.5	8.5
	5	9.0	6.1	7.9	21.8	11.5	20.0	10.2	9.1	8.5
	6	8.9	6.2	7.9	21.8	10.7	20.3	10.1	9.6	8.4
	7	8.9	6.7	7.8	21.2	9.4	19.9	10.0	8.5	8.3
	8	8.8	6.3	7.8	20.8	8.9	19.6	10.1	9.8	8.2
	9	8.6	4.7	7.8	20.5	8.0	19.6	10.5	10.2	8.3
	10	8.6	4.6	7.7	20.0	7.1	19.4	11.4	10.8	8.7
$\frac{L_R}{2}$	4	13.9	19.1	8.7	25.6	19.9	21.2	14.8	19.3	9.2
	5	13.4	15.0	9.1	25.2	16.3	22.2	14.9	18.1	9.8
	6	13.7	12.9	10.1	25.6	14.6	23.4	15.2	16.2	10.7
	7	13.1	10.9	10.5	24.6	12.8	22.9	14.4	12.4	11.2
	8	13.0	10.1	10.5	24.2	11.9	22.4	13.9	11.8	11.0
	9	13.0	9.8	10.8	24.0	11.4	22.4	14.6	12.4	11.6
	10	13.3	9.2	11.3	23.9	10.4	22.7	15.9	12.9	12.8

algorithms, the proposed method consistently obtains the best results, while having to solve DoA ambiguity. Regarding the length of the reference signals, the results obtained for the longer reference signals ($2L_R = 8192$ and $L_R = 4096$) are practically equivalent, however when the shorter reference signals are used ($\frac{L_R}{2} = 2048$) the localization error is notably increased. This is explained by the number of erroneous ToA estimates growing larger when the length of the reference signals is not enough to counter the effects of low SNR and reverberation. The proposed method also shows a better behavior in the presence of estimation errors. In contrast to this, Crocco et al. [CDBBM12] and Jacob et al. [JSHU12] performance is highly affected by failed cross-correlations. The sensibility of these method to estimation errors clearly shows when comparing the results obtained with increasing L_R values, and it is best reflected by their Std values. The influence of larger network sizes is also more noticeable on the Std values that, for the most part follow a descending trend with N, except with the proposed method where larger networks start affecting the convergence of the DoA ambiguity solution.

Table 3.2 shows the Mean, Std and Trim of the localization error obtained with the proposed algorithm working with known orientations (assumed to be obtained with an electronic compass), with and without orientation measurement error, and those obtained when the orientations are

Table 3.3: Orientation estimation error (degrees) for different algorithms and network sizes. DoA estimates with ambiguity have a range of 180° instead of 360°.

		Jacob et al. [JSHU12]			**Proposed**		
		Without ambiguity			With ambiguity		
lenght	N	Mean	Std	Trim	Mean	Std	Trim
$2L$	4	4.6°	4.7°	3.7°	6.6°	15.7°	4.0°
	5	4.3°	4.6°	3.4°	5.9°	12.8°	3.7°
	6	4.0°	4.3°	3.2°	5.1°	9.6°	3.4°
	7	4.2°	5.7°	3.1°	5.6°	12.5°	3.2°
	8	4.4°	7.3°	3.1°	5.0°	10.9°	3.0°
	9	4.4°	8.0°	3.0°	8.6°	19.1°	3.1°
	10	4.3°	6.1°	3.0°	12.3°	21.9°	5.7°
L	4	4.5°	4.9°	3.7°	6.3°	14.3°	3.9°
	5	4.2°	4.2°	3.4°	5.8°	12.0°	3.7°
	6	4.1°	4.3°	3.3°	5.0°	9.0°	3.4°
	7	4.1°	4.7°	3.1°	6.0°	14.1°	3.2°
	8	4.6°	8.5°	3.1°	6.0°	14.5°	3.0°
	9	4.5°	7.6°	3.0°	8.3°	17.3°	3.2°
	10	4.3°	5.8°	3.0°	13.1°	24.3°	5.3°
$\frac{L}{2}$	4	8.9°	23.3°	3.9°	9.2°	22.1°	4.1°
	5	8.5°	20.7°	3.6°	9.5°	23.1°	3.9°
	6	10.2°	20.9°	3.6°	9.3°	22.4°	3.6°
	7	10.5°	19.5°	4.2°	8.2°	18.3°	3.4°
	8	10.9°	18.9°	5.1°	7.2°	17.4°	3.2°
	9	11.5°	17.3°	6.5°	9.3°	19.9°	3.3°
	10	12.7°	16.5°	8.8°	13.1°	23.9°	5.6°

estimated. Comparing this table with the previous one, we observe that the error obtained using orientation estimates is between those obtained using measured orientations: it is larger than that of the ideal case but much smaller than when the typical error of an uncalibrated compass is introduced. Thus we can conclude that, in the presence of an uncalibrated electronic compass, it is more reliable to estimate the node orientations rather than measuring them. Note that when the orientations are known *a priori* there is no need for solving the DoA ambiguity, thus, the STD values show a descending trend when N is increased.

Although not show in the results, the use of NCME algorithm for the final node location estimation increases the localization error by 12% on average, compared to the FCME solution. On the other hand, obtaining the solution with the NCME method requires on average 21% less operations than doing so with the FCME method, thus, favoring one method over another is a tradeoff between computational complexity and accuracy.

Table 3.3 also shows the Mean, Std and Trim, in this case of the orientation estimation error with the proposed method and Jacob et al. [JSHU12]. The proposed method gets larger errors that Jacob et al. [JSHU12], although it is worth recalling that the latter is not dealing with DoA ambiguity. From the table we can observe that, orientation estimation is less affected by estimation errors, but

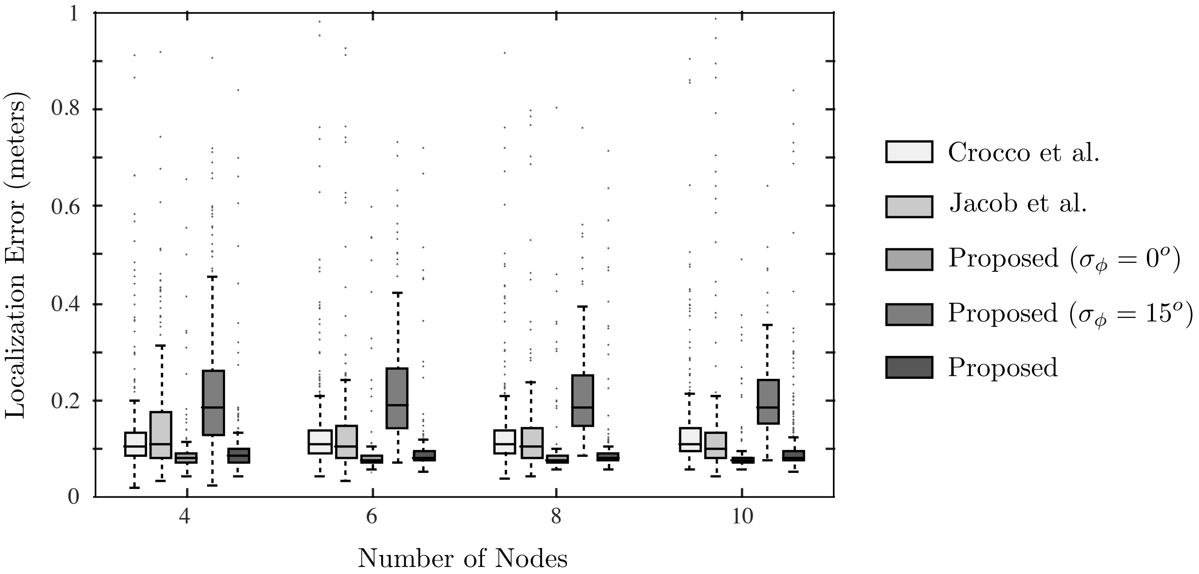


Figure 3.10: Boxplot of the self-localization results for the tested algorithms with different number of nodes and $L_R = 4096$.

Table 3.4: Standard deviation of the clock offset (samples) for different network sizes and reference signal lengths ($L_R = 4096$).

	4	5	6	7	8	9	10
$2L_R$	9.60	8.70	8.59	7.88	7.06	6.75	6.39
L_R	11.08	8.48	8.25	7.38	6.88	6.42	6.32
$\frac{L_R}{2}$	34.60	25.52	26.08	24.87	24.54	26.56	28.65

again the proposes method is more robust. With both methods the orientation estimation error is lower than that of a typical digital compass, rendering them useless for this particular application.

Figure 3.10 shows a boxplot of the results obtained for all the tested algorithms with $L_R = 4096$, from where is easier to compare how the different algorithms perform. In general, large localization errors are associated to large time lag estimation errors, i.e., those instances when the largest peak of the correlation corresponds to a reflection instead of the direct signal. Some proposals in the literature use outlier detection techniques to reduce the effect of spurious measurements. Jacob et al. [JSHU12] used random sample consensus (RANSAC) for the minimization algorithm (not implemented in our version); while in Plinge and Fink [PF14], outliers are detected by applying a threshold to the estimation error. Our current implementation does not contemplate outlier detection; therefore, the obtained localization errors have large variances.

Regarding the number of nodes, localization accuracy usually increases with larger networks. This result is expected because there is more information available; thus, it is easier to compensate for small pairwise estimation errors (either DoA or range) in one or several nodes. However, due to DoA ambiguity, the proposed method has some convergence problems with large networks that need to be addressed.

Finally, Table 3.4 shows the standard deviation of the clock offset (σ_Δ)for the 3 tested reference signals lengths with different array sizes. The obtained results show that when the time lags are successfully estimated, it is possible to obtain a fairly reasonable synchronization. Note that the results are larger than that predicted in 3.3.3 because of the time lag estimation errors. It is worth remembering that this method does not account for clock drift, thus, clock offsets will grow larger in time if not additional action is taken.

3.7 Conclussions

In this chapter, a new active self-localization algorithm for WASNs has been presented. The algorithm assumes that the nodes are equipped with at least two microphones and one speaker. The entire localization process is based on the combination of DoA and range estimates between node pairs obtained from acoustic signals, and it does not require synchronization between the nodes.

The main novelty of the proposed method is the ML estimator of node locations, that obtains a simple closed form solution to the self-localization problem by treating pairwise angle and range estimates as polar coordinates pointing towards the possible relative location of the nodes. Perhaps most importantly, the proposed method is capable of solving the self-localization problem even without prior knowledge about the orientation of the nodes and in the presence of DoA ambiguity. To the best of our knowledge, our proposal is the only method capable of 2D DoA-based self-localization using nodes equipped with only 2 microphones.

A byproduct of the proposed self-localization method is the obtention of synchronization within the network. With a simple modification, the method used for obtaining the range estimates can be used to estimate the clock offset between the nodes. The error of these clock offset estimates is directly related to the distance between the center of the array and the speaker of the nodes, thus, it is better classified as ‘loose’ synchronicity method. Although, as we will see on the next chapter, the precision of this synchronization suffices for the problems that we are to be tackled.

In summary, the performance of the proposed algorithm is on the same scale as other DoA-based algorithms without requiring ad-hoc hardware or synchronization between nodes. The localization error obtained with estimated orientations is lower than that obtained when an uncalibrated electronic compass is used. Furthermore, the computational cost of the algorithm is assumable for current mobile processors, and since it is able to work on unsynchronized networks and it only requires 2 microphones per node, is a good candidate for implementation on a WASN composed of commercial ‘off-the-shelf’ smartphones.

However, the solution to the DoA ambiguity using a GA tournament adds a significative computational load, since it requires an iterative minimization. Furthermore, its convergence is compromised in networks with a large number of nodes. At this point, it is worth mentioning that the GA tournament is not intended as a definitive solution to this problem, but as a proof of concept, showing the feasibility of the DoA ambiguity solution. It is worth exploring more efficient solutions that exploit the relations between the DoA estimates of various nodes.

Another aspect worth of consideration is the ability to assert the quality of the cross-correlations. Most of the work on acoustic-based self-localization and other related fields rely on accurate time lag estimates, and oddly enough, the quality of the correlations is often overlooked. The techniques applied to self-localization are very sensitive to large estimation errors, more often than not, one wrong enough estimate can inhibit the convergence to the correct solution. Some proposals use outlier detection techniques to lessen the problem, such as making an statistical analysis of the estimates, or taking a 'brute force' approach by evaluating the results with multiple subsets of estimates. However if there was a direct method to 'measure' the confidence of individual cross-correlations, it would then be possible to apply a weighted optimization scheme, that would improve the quality of the results without a significative impact to its computational complexity.

As with any other sound-based algorithm, the proposed self-localization method can only be used when the distance between the nodes is relatively small. Although, in a hypothetical scenario in which the nodes where equipped with a RF-TRx with more than one antenna and the ability to measure the angle-of-arrival of the RF signals, the proposed algorithm could also be applied. In that scenario range and DoA estimates would be substituted by RF RSS and angle-of-arrival respectively. Working with RF signals instead of audio, synchronization could not longer be obtained and localization error would likely be much higher, but the localization procedure would works with much larger distances, which can be desirable in many applications.

Future work will address spurious estimates and will study different approaches to the DoA ambiguity estimation because they are the main sources of error.

3.8 Summary

The main contributions described in this chapter of the thesis are the following:

1. A simple closed form solution to the self-localization problem that uses pairwise DoA and range estimates obtained from acoustic signals exchanged between devices, without requiring prior synchronization.

2. Two different solutions to the maximum likelihood estimator that either consider the estimates equally accurate or assigns a different weight to each estimate based on its confidence level.

3. A solution to the DoA ambiguity problem that makes it possible to obtain the node locations and their orientation using only two microphones per node.

4. An efficient set of methods for obtaining DoA, range, orientation and clock offset estimates, that work by combining various time lag estimates between the signal recorded at a microphone and a reference signal.

The contributions described in this chapter have originated publications [P3] and [P1] which are detailed in section: Publications.

CHAPTER 4 Impulsive Sound Detection

4.1 Introduction

Over the last decades, there has been a surge in the presence of surveillance systems in our living environment, so it should not come as a surprise that the most prevalent applications of WASNs in the literature are related to automated acoustic surveillance. Using acoustic signals as a means to autonomously monitor an environment is cheaper than other common solutions, such as video monitoring, and has a series of benefits when compared to them: no need for direct line of sight, the sensors have an unrestricted field of view (i.e., onmidirectional microphones), illumination and temperature are not relevant issues, some audio events relevant to surveillance, such as screams, do not clearly show on other sensors, etc.

WASNs are specially well suited for this task since they can be easily deployed to cover a large area, with a minimal setup. Aditionally since sound signals are less data intensive than other alternatives, the cost and physical size of the nodes can be kept low, so that the size of the network can grow.

Generally speaking, the kind of acoustic events that automated audio surveillance systems are set to detect, can be catalogued as "rare" acoustic events (specially the impulsive ones), which implies that their occurrence can no be predicted, thus, creating the need for continuous monitoring. Moreover, a disproportionate amount of time is spent monitoring background noise, thus every node in a WASN-based detection system needs to process their input audio stream continuously to maximize probability of detection.

Most recent work in the more generalist sub-field known as AED, that deals with many different types of audio events, tends to focus on very large ANNs inherited from ASR and computer vision. Such systems, while capable of achieving the best detection results to this day (for multiple simultaneous detections), are far too complex to be implemented on a small wireless node. On the contrary, impulsive audio events are fairly easy to detect given their particularities, hence the detection system does not need to be overly complex, thus traditional machine learning techniques, often seen as 'outdated', can still serve the purpose.

Rather than relying on a single very complex detector, it is possible to take advantage of an ensemble of simpler, spatially distributed detectors so that their combined performance surpasses that of any of the members. Thus, sound event detection on WASNs can be made to be both reliable and computationally efficient.

4.2 Feature Extraction

The most common approach to acoustic impulsive event detection is to transform an audio stream, frame by frame, into a series of feature vectors that can be individually evaluated with some classification algorithm. The features are computed from a segment of raw audio data of length L, using a set of functions whose outputs are concatenated to form the feature vector of length $F < L$. Feature extraction serves two main purposes: to reduce the dimensionality of the raw input data, and to produce data that is mathematically and computationally convenient to process.

Perhaps the most popular features for all applications rooted in audio-based machine learning are the MFCCs, which are a compressed perceptual representation of the short-term power spectrum of the input signal. Computing F MFCCs requires computation of a F point DCT, intended to further compress the information onto the lower frequency bands. Thus, in those cases in which better computational efficiency is wanted, or if the objective of the feature vector is to approximate a logarithmic frequency scale, the DCT can be omitted. The DCT-less version of the MFCCs is sometimes called Mel log-band energies. These features are also widely used for audio-based detection and classification and they have been shown to obtain results comparable (sometimes better) to those of obtained with the MFCCs [CHHV15, GPAR+15]. In order to keep the feature extraction process as simple as possible, the proposed detector system works with Mel log-band energies (and linear combinations of them) as the only features. This way the basic feature vector for the l^{th} frame ($\mathbf{x}_l$) is computed as described in section 2.7.2.

4.2.1 Efficient Feature Temporal Context

Given that sound is a time evolving signal, it makes sense to feed the evolution of its spectral content to the classification algorithm in order for it to find temporal patterns. Certain types of classifiers are able to work with time evolving feature vectors, such as recurrent neural networks or HMMs, that use their internal state to process a sequence of inputs. However, most classification algorithms work by examining isolated inputs, thus the possible relations between the current time instant and those preceding it are not considered by the classifier.

A common workaround for this issue, often used in conjunction with the MFCCs, are the differential (delta) and acceleration (delta-delta) coefficients. They can be be described as the trajectories of the MFCCs over time, which are the difference between j^{th} coefficient on the current frame and the j^{th} coefficient on the previous frame, for the delta, and the same thing but with that of two frames ago for the delta-delta. These time-relative features are then concatenated with the MFCCs, so that the length of the feature vector increases by $2F$, where F is the number of coefficients.

Instead of using the differences between the coefficients at various time instants, it is also commonplace to just concatenate multiple frames, to provide the classifier with some temporal context. This way the feature vector becomes:

$$\mathbf{x}'_l = \begin{bmatrix} \mathbf{x}_l \\ \vdots \\ \mathbf{x}_{l-\Gamma} \end{bmatrix}, \tag{4.1}$$

where Γ is the number of past frames that get concatenated with the current one. Note that the feature vector grows by F features for every past frame added to it, so both the dimensionality of the problem and the computational complexity of the evaluation are increased by a factor $F \times \Gamma$. Although the size of the feature vector is most likely to be negligible for most target platforms, it should be mentioned that memory space requirements are also increased by the the same factor.

In order to increase the temporal information available to the classifier without excessively increasing the dimensionality of the problem, we propose to use a 'compressed' feature temporal context scheme. The main idea is to reduce the resolution of the past frames that get concatenated to the current feature vector, such that the further away in time a given frame is, the lower its resolution.

Let us consider a past feature vector $\mathbf{x}_{l-\varsigma}$, $\varsigma = 1, \dots, \Gamma$, defining the compression ratio as $\mathbf{c}_R = [1, c_{R,1}, \dots, c_{R,\Gamma}]$, we can also define the compressed version of a given feature vector $\dot{\mathbf{x}}_{l-\varsigma}$ of length $\frac{F}{c_{R,\varsigma}}$, where F is divisible by $c_{R,\varsigma}$. Compressed feature vectors can be computed in a multitude of ways. Regardless of the selected method, the best results are to be obtained when the original vector $\mathbf{x}_{l-\varsigma}$ is used as an input for the compression. However, by doing this, the complexity of the process will increase at least proportionally to the number of past frames included as temporal context. We propose to use a sequential compression scheme, where the length of a given feature vector is reduced progressively by computing the ς-times compressed vector as a linear combination of the elements in the (ς-1)-times compressed vector. This way, we can define a series of matrices $\mathbf{B}_\varsigma$ of size $(\frac{F}{c_{R\varsigma}} \times \frac{F}{c_{R\varsigma-1}})$ such that: $\dot{\mathbf{x}}_{l-\varsigma} = \mathbf{B}_\varsigma \mathbf{x}_{\varsigma-1}$, $\varsigma = 1, \dots, \Gamma$. Using compressed temporal context, the feature vector becomes:

$$\mathbf{x}'_l = \begin{bmatrix} \mathbf{x}_l \\ \vdots \\ \dot{\mathbf{x}}_{l-\Gamma} \end{bmatrix}, \text{ where } F' = F + \sum_{\varsigma=1}^{\Gamma} \frac{F}{c_{R,\varsigma}}. \tag{4.2}$$

The simplest version of this algorithm, which is the one that is being used in the proposed detection system, is to set F and $c_{R,\varsigma}$ so that the relative compression ratio equals 2 between successive compression stages. Then, one feature on the ς-times compressed vector is easily computed as the average of two features of the (ς-1)-times compressed vector. Note that it is not mandatory to compress the past feature vectors each time a new frame arrives. $\mathbf{c}_R$ can share the same compression ratio for various values of ς, and by doing so, the computational complexity of the process is further reduced.

Figure 4.1 shows a schematic representation of two methods for adding feature temporal context, a 2-frame "full" context feature vector, and 4-frame compressed context feature vector with $\mathbf{c}_R = [1, 2, 4, 4]$. The resulting length of the extended vector is the same with both techniques.

4.3 Local Detector

After transforming the raw input audio into a number of descriptive values (i.e., features) we have to define the process by which each observation is automatically assigned to one of a number of categories. The most common approach is to train some classifier on a pre-defined dataset formed by labeled examples, so that it can learn how to recognize the class of an observation based on its

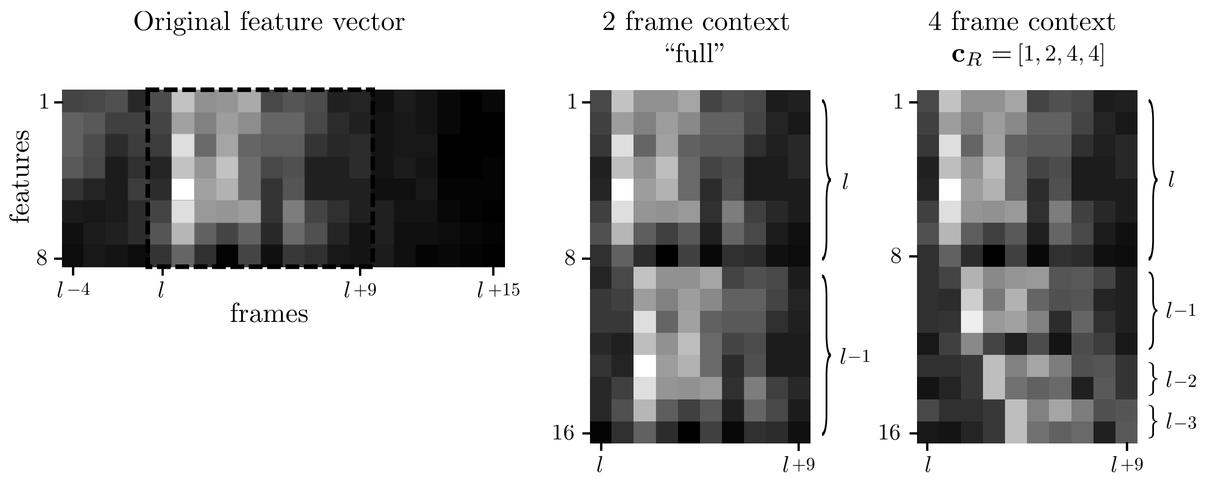


Figure 4.1: Schematic representation of feature time context, showing extended feature vectors with 2 frames full context, and 4 frame compressed context.

features. In this particular case, the classifier, or more properly the detector, has to be able to find observations belonging to the target classes on a continuous stream of data, in which most observations do not belong to any of the classes.

This section describes the algorithm used by each of the nodes forming a WASN to perform impulsive audio event detection locally. At this point, there is no collaboration from the network, thus, every node must process its own input data in isolation. Generally speaking, complex algorithms often obtain better results at the cost of a larger computational load, however, the main objective of this thesis is to develop efficient techniques, thus, a tradeoff solution between performance and efficiency must be found.

4.3.1 *K*-LDA

In order to increase the performance of the detection while keeping computational complexity as low as possible, we propose to use a modified LS-LDA classifier that can use one of various weight matrices, depending on where a given observation resides within the feature space. We are calling this proposal K-piecewise LDA, or K-LDA for short.

We have already mentioned that LDA is a linear classification method, thus, it is not a good tool when the classification boundary can not be described as a linear combination of the feature vector. However, it is possible to assign different classifiers to different regions of the feature space to approximate nonlinear boundaries with a piecewise linear function. The main idea is to divide the feature space into a number of discrete regions according to some metric derived from the features themselves, so that, a set of 'specialized' LDAs can be trained for each region.

Let us consider a training set in the form of an input matrix $\mathbf{Q} = \{\mathbf{q}_n\}$, where $\mathbf{q}_n = [x_{n1}, \ldots, x_{nF}, 1]^T$, from which it is possible to derive a segmentation function such that: $k = \mathcal{F}(\mathbf{x}_n)$, $k = 1, \ldots, K$. Let us also consider that the segmentation function is used to divide set $\mathbf{Q}$ into K disjoint subsets: $\mathbf{Q}_1 \cup \cdots \cup \mathbf{Q}_K = \mathbf{Q}$. Defining $\mathbf{Q}_k$ as the set of C-dimensional binary target vectors associated to each observation $\mathbf{x}_n \in \mathbf{T}_k$, then it is possible to train a LS-LDA over the observations assigned to the k^{th} input subset with:

$$\mathbf{V}_k = \mathbf{T}_k {\mathbf{Q}_k}^T \left(\mathbf{Q}_k {\mathbf{Q}_k}^T\right)^{-1},\ k = 1, \ldots, K, \tag{4.3}$$

where $\mathbf{V}_k$ is one of K 'specialized' weight matrices of dimensions $C \times (F+1)$ given by:

$$\mathbf{V_k} = \begin{bmatrix} w_{k11} & \ldots & w_{kF1} & b_{k1} \\ \vdots & \ddots & \vdots & \vdots \\ w_{k1C} & \ldots & w_{kFC} & b_{kC} \end{bmatrix}. \tag{4.4}$$

Since this chapter of the thesis deals with detection, let us consider a binary classification problem, so that $C = 1$ and $\mathbf{v}_k = [w_{k1}, \ldots, w_{kF}, b_k]^T$. In that scenario the output of the proposed K-LDA classifier for a given observation $\mathbf{x}_t$ can be obtained as:

$$y = b_k + \sum_{j=1}^{F} w_{kj} x_{tj},\ \text{with: } k = \mathcal{F}(\mathbf{x}_t). \tag{4.5}$$

Please note that the computation of the output is equivalent to that of a regular LS-LDA, first described in section 2.7.4. The particularity is that the weights and bias have first to be selected among K candidates, by finding the correspondent spatial region in which $\mathbf{x}_t$ resides, which requires previous computation of $\mathcal{F}(\mathbf{x}_t)$.

To better illustrate the basic behavior of the proposed classifier let us formulate a toy problem. Let us consider a pair of two dimensional normally distributed variables with zero mean but different standard deviation. We want to to classify a set of random observations $\mathbf{x}_n = [x_1, x_2]^T$ as belonging to distribution A or B. With a linear classifier, the error rate would tend to 50% since it can only separate 2D space with a line. However, if we have some a priori knowledge about the obervations, such as that both variables have zero mean in both its components, we can take advantage of this knowledge to segment the feature space accordingly. On a real problem we would not have knowledge about the true distributions, thus, defining μ_1 and μ_2 as the mean value of the samples in the training set, the feature space can be divided into 4 zones with:

$$k = \begin{cases} 1, & \text{if } x_1 \geq \mu_1 \text{ and } x_2 \geq \mu_2 \\ 2, & \text{if } x_1 \geq \mu_1 \text{ and } x_2 < \mu_2 \\ 3, & \text{if } x_1 < \mu_1 \text{ and } x_2 \geq \mu_2 \\ 4, & \text{if } x_1 < \mu_1 \text{ and } x_2 < \mu_2 \end{cases} \tag{4.6}$$

Figure 4.2 illustrates this problem, by showing the classification boundaries obtained by a a LS-LDA and a a K-LDA using a specialized classifier for each quadrant of the feature space, delimited by the mean value of the samples in the training set. This toy problem clearly illustrates how the accuracy of the classification can sometimes be improved while using linear combinations by simply trying to find some underlying structure within the feature space.

In order to be able to use the proposed K-LDA classifier with different sets of data; we need a reliable method to divide the feature space, that does not require any a priori knowledge. Consequently, we have decided to use clustering as the segmentation method, more specifically, the well known K-means algorithm described in section **??**. From a clustering perspective, $\mathbf{X}$ is some data

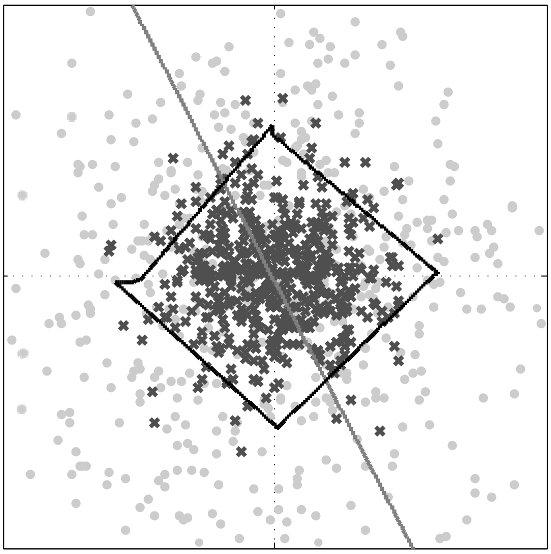


Figure 4.2: Classification boundaries of a LS-LDA and a K-LDA ($K = 4$) for 2 classes with 2D gaussian distributed features.

set that gets partitioned into K subsets $\mathbf{X_k} = \{\mathbf{x}_n \in \mathbf{z}_k\}$, in this particular case using K-means. Once the train set has been clustered, defining $\mathbf{t}_n$ as a binary target vector associated to each $\mathbf{x}_n$, is then possible to train a LS-LDA over the observations assigned to one of K clusters using expression (4.3).

This way the proposed K-LDA classifier has a hybrid learning process with an initial unsupervised clustering stage followed by traditional supervised training for each of the clusters (i.e., subsets of the training data). The improvement obtained with the use of K classifiers is directly related to the existence of K 'valid' clusters in the data. Clustering algorithms tend to find clusters in the data whether or not any clusters are really present. Furthermore, automatically determining the number of clusters is one of the most difficult problems in data clustering. We have found that for most classification applications, a good initial guess is to set the number of clusters K equal to the number of classes C.

Assuming that some target processor is capable of performing a Multiply-ACcumulate (MAC) operation[1] as a single operation, and dismissing memory operations; the computation of a regular LS-LDA output takes $F \times C$ operations. In comparison, the proposed classification scheme, using the L^2 norm, requires approximately $K \times F^2$ additional operations. This computational overhead is composed of: F subtractions and F MACs for each of the K squared L^2 norms, and $K - 1$ comparisons for finding the minimum distance. Considering $K = C$, finding out which specialized classifier to use, takes approximately F times more operations than the classification itself. One might think that using a simpler distance measure, such as the L^1 norm, can even things out, however any distance measure is going to require a number of operation proportional to F. Using the L^1 norm, multiplications of differences are exchanged for the absolute value of differences, and since the evaluated distances are squared, it requires an additional multiplication, thus, computational complexity stays in the same order as that of the squared L^2 norm.

[1] A MAC operation can be defined as the sequence of two elementary operations: two operands b and c are multiplied and the result is added to accumulator a, such that $a \leftarrow a + (b \times c)$.

4.4 Multichannel Detection

This section describes how the local decisions taken by a group of nodes are combined to generate a 'global' detection with the objective of obtaining a result more trustworthy than that of any local detection. We have decided to use decision fusion, again, for the sake of efficiency. Compared to decision fusion, data fusion in the time domain (i.e., beamforming) requires much higher bandwidth (audio signals need to be continuously exchanged with the network), and an ordinary implementation[2] also requires thigh time synchronization and knowledge about the geometry of the network and the source location. In contrast to this, combining multiple outputs, or multiple feature vectors, greatly reduces data transmissions since usually $L \gg F \gg C$. Regarding synchronization, the detection system is very likely to work with framed audio signals, thus the synchronization constraints when combining either features or decisions are greatly relaxed. Even more so, assuming that the combined transmission time of all the messages needed to perform the data fusion is lower than $\frac{L}{2f_s}$, there is no real need for inter-node synchronization. The technical differences between feature fusion and decision fusion are narrower. Fusing feature vectors typically requires a larger number of operations than fusing decisions, although, with a single 'unified' feature vector only one evaluation is needed (i.e., ideally computed using a distributed algorithm). However, using a naive approach, decision fusion is overall the simplest data fusion method, and it has been shown to outperform both feature fusion [MSDC10] and beamforming [GPKM14].

4.4.1 Proposed Decision Fusion scheme

Let us consider a WASN formed by N nodes, where every node is loaded with the same detection algorithm. For simplicity, let us also consider binary local detectors ($C = 1$) capable of producing a probabilistic output $y_n(t) \in \mathbb{R}$, $n = 1, \ldots, N$, where t is the frame index. Using the average fusion rule, the global decision taken by the ensemble of detectors can de expressed as:

$$D(t) = \begin{cases} 0, & \text{if } y(t) < y_0 \\ 1, & \text{if } y(t) \geq y_0 \end{cases}, \text{ with: } y(t) = \frac{1}{N} \sum_{n=1}^{N} y_n(t), \tag{4.7}$$

where y_0 is the threshold value for considering the averaged probabilistic output of the ensemble as a detection. In order to avoid problems due to malfunctions or false local detections, it is good practice to clip the individual outputs at the value assigned to the labels of the training set, e.g., $0 \leq y_n(t) \leq 1$.

Expression (4.7) defines how the global decision is obtained for every time frame (i.e., instantaneous decision). However, it is worth remembering that it is very likely for nodes on a WASN to be separated by distances greater than the time difference between consecutive frames times the speed of sound. Assuming perfect local detection, the detection instant at each channel (i.e., node) is directly related to the local ToA of the event. At this point there is no knowledge about the location

[2] Ordinary beamforming involves the weighted combination of multiple time-aligned data channels, where the weights and time lags depend on the array geometry an the source location (array response). There is also a group of techniques known as 'Blind Beamforming' capable of enhancing the SNR of a signal without explicit knowledge about the array response.

of the source, thus, the TDoA can not easily be obtained. Differences among arrival times cause the outputs of each channel to have a unknown time lag between them that affects the instantaneous global decision making it unstable. Under a perfect local detection assumption, average fusion is equivalent to majority voting, thus, the global decision can be seen as that taken by the majority of the detectors in the ensemble. In fact, as a direct result of sound propagation, it is possible to have various simultaneous local detections and no global detection and vice versa.

In order to avoid multiple detections of a single acoustic event, and to minimize data transmissions, we propose to use a windowed detection algorithm. The main idea is to only compute the global decision, once an event has been detected locally, and to only do so for a fixed amount of time. Defining the local decision of the n^{th} node as $D_n(l)$, which is computed applying a threshold function to y_{nl}; the frame index of a given detection is ι. We denote the global detection window length as T_D so that the the global decision is computed for $l = \iota, \ldots, \iota + T_D - 1$. This way, once a local detection has been produced, the result of the windowed detection can be expressed as:

$$D = \underset{l \in \mathcal{W}}{\arg\max} \left(D(l) \right), \text{ where: } \mathcal{W} = \{ l \in \mathbb{Z} : \iota \leq l < \iota + T_D \}. \tag{4.8}$$

The critical factor for the success of this operation in avoiding multiple detections of a single event is the window length T_D. The window length should be set to match the expected duration of a given event detection (i.e., the number of active decision frames). Multiple detections can also be reduced by only considering $D_n(l)$ a local detection when it produces a rising edge, that is $D_n(l) = 1$ with $D_n(l-1) = 0$. This way it is possible to avoid the start of a new detection cycle right after the ending of the previous one.

Even after combining multiple channels, the output of the detector can still be too noisy. It is commonplace to use some post-processing on the decision vector, such as applying median filtering. The median filter is a nonlinear digital filter, often used to remove noise from a signal. The algorithm runs through a signal frame by frame, replacing each entry with the median of a number of neighboring frames. The number of frames is called the ‘window’, which slides though the signal resulting in a ‘smoothing’ effect. The median filter window length can be also adjusted to the expected length of a detection (in frames) so that sudden decision changes can be dismissed.

It is important to mention that, as long as the maximum distance between microphones has an associated time lag smaller than the expected duration of the detection, there is no need to align the individual outputs. However for very large networks time lag could posse a problem, so further processing is required. Also, this detection strategy lacks the quick response needed in the presence of multiple audio events in rapid succession (e.g., a shootout in a war zone), although in that situation the bigger concern would be dealing with overlapped audio events, that is multiple target events from different sources inside a single time frame.

Figure 4.3 shows the probabilistic outputs (grey) and decision outputs (black) of a multichannel detection system ($N = 8$) where the light grey area indicates the detection window. The global decision is post-processed with a median filter (3 frames window length), the dashed line represents the raw decision output that has a sudden decision change removed in the filtered version. Note that in this example only one detection cycle is initiated since none of the channels has a rising edge after the decision window.

The proposed collaborative detection process has the following steps:

1. Every node in the network continuously monitors its input stream frame by frame with the selected local detector.
2. If a node detects an event locally, it broadcasts the detection to the network, and gets in charge of the global detection.
3. Every node in the network acknowledges by sending the probabilistic output of their local detector to the node in charge and set a counter $a = 1$.
4. The nodes wait for the next frame and then send the new output to the node in charge and increase their counter $a \leftarrow a + 1$.
5. If $a < T_D$, the previous step is repeated.
6. If $a = T_D$, the node in charge computes the average of the received signals and checks for a global detection. The result is broadcasted to the network.
7. If there was a global detection every node sets a local flag $D_F = 1$.
8. The nodes wait for the next frame, if they do not get a local detection they set $D_F = 0$ and go back to step 1.
9. If $D_F = 1$ the previous step is repeated.

Considering that the local probabilistic output is computed in single-precision floating-point format (32 bits), the required wireless data rate during a windowed detection cycle (ignoring transmission overhead) is: $32 \times N \times C \times T_D \times \frac{f_s}{L_H}$, where L_H is the hop length in samples, i.e., the number of non overlapped samples between consecutive frames.

4.5 Experimental Work and Results

This section describes the experiments conducted to evaluate the performance of the impulsive sound event detection scheme proposed in this chapter, as well as a discussion of the obtained results. The evaluation was done using a multichannel database of real impulsive source recordings on a spatially diverse scenario, that is described bellow.

4.5.1 Description of the Database

To evaluate the proposed algorithm, we purposely recorded a spatially diverse database of impulsive sound sources using a wired microphone array. The impulsive noises were generated on site using various sources at multiple locations. The acoustic events of interest included on the database correspond to one of the 5 following impulsive sounds:

- Firecracker explosion.

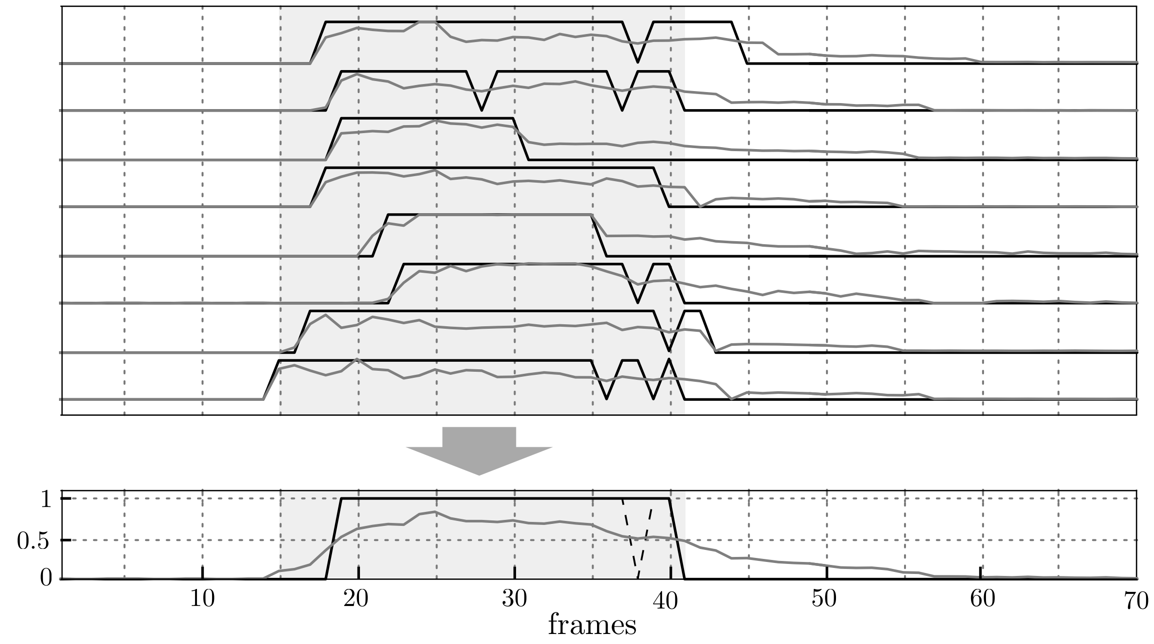


Figure 4.3: Probabilistic outputs (grey) and decision outputs (black) of a multichannel detection system. Top: Local signals, Bottom: Global signals.

- Balloon pop.
- Compressed air gun shot.
- Drum hit.
- Handclap.

Recordings were taken on a laptop computer using the following equipment:

- *Sony ECM-TL3* compact omnidirectional electret condenser microphones (frequency response 20-20,000Hz and -35dB sensitivity according to specification).
- *TASCAM US-16x08* USB sound interface.
- *T-bone PPA 200* phantom power adapters for the electret microphones.

All sounds were recorded using digital audio editor 'Audacity' on 24bit/44100Hz WAV format without any preprocessing.

Recording took place on a backyard in a rural environment. The recording area was a grass yard with several trees and surrounded by roads and small buildings. Sound was recorded using a 12 microphone "unstructured" array, with a maximum microphone distance of 16.2 m and a minimum microphone distance of 1.3 m. The location of the sources was set at random throughout the recording area (close to 1000 m^2), but trying to cover as much ground as possible. The maximum distance between a source and a microphone is 39.5 m, while the minimum distance is 3.2 m. In total there are $12 \times 16 = 192$ unique propagation paths, providing an adequate amount of spatial diversity.

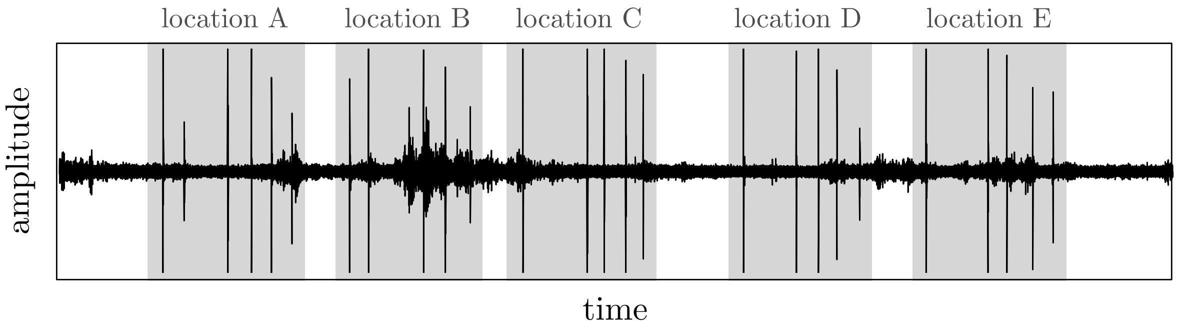


Figure 4.4: Recording extracted from the database showing 25 impulsive events of interest and the temporal windows of 5 source locations.

The database recording was divided in two sessions with a time lapse between them of around 4 hours (i.e., morning and evening). Each session is composed of recordings of the 5 sound sources at 16 different locations (every sound was generated at every location), adding up to 80 sound events per session. Locations are mantained within sessions, thus, each location appears twice on the database. The total number of sound events included in the database is: $5 \times 12 \times 16 \times 2 = 1920$. Each session is composed 3 sets of 12 track recordings (one per microphone), that captured the sound events generated at various locations (5 locations for the first two takes and 6 locations for the third take). The total running time of the recordings (i.e., adding up every channel) is approximately 6.5 hours. The recordings include natural occurring background noise such as speech, human activity and passing cars. For the most part, unless a noise source is in close proximity to one of the microphones, the SNR of the of the impulsive events is very high due to their elevated energy, which is to be expected. The microphones were covered with foam windshields, but especially on the first session, wind can still be heard on ocassion. Figure 4.4 shows one of the raw recordings included in the database, where the relative amplitudes of the target impulsive events and background noise can be appreciated.

Labeling was done manually by selecting the class, ‘onset’ time and ‘offset’ time of every event on every recording as precisely as possible. We define onset time as the time instant when a sound first reaches a microphone (i.e., start time), and offset time as the time instant when the majority of the energy has been received, excluding reverberations (i.e., finish time). The average time duration of the labeled sound events is approximately 150ms. Please note that time series labeling is based on a subjective assessment that it is prone to human error, specially in the case of offset time labeling. In contrast to this, class labels are perfectly defined since the impulsive sounds were generated following the same exact sequence each time, thus, the time labels can ben regarded as ‘weaker’ than the class labels. It is important to mention that there is an additional set of time labels that are related to the source locations. These labels mark the temporal window in which the 5 sound events generated at one location are contained within the recorded signal. The temporal windows for each microphone-source location combination were obtained adding a guard interval of 3s before the onset time of the first event and after the offset time of the fifth event. The average time duration of the location windows is approximately 43s. On Figure 4.4 the time windows of 5 locations are represented by a grey area.

For the purpose of testing the prosed detection system, the 5 class labels were collapsed into a single ‘activity’ label. This way the database is divided into target impulsive audio events and back-

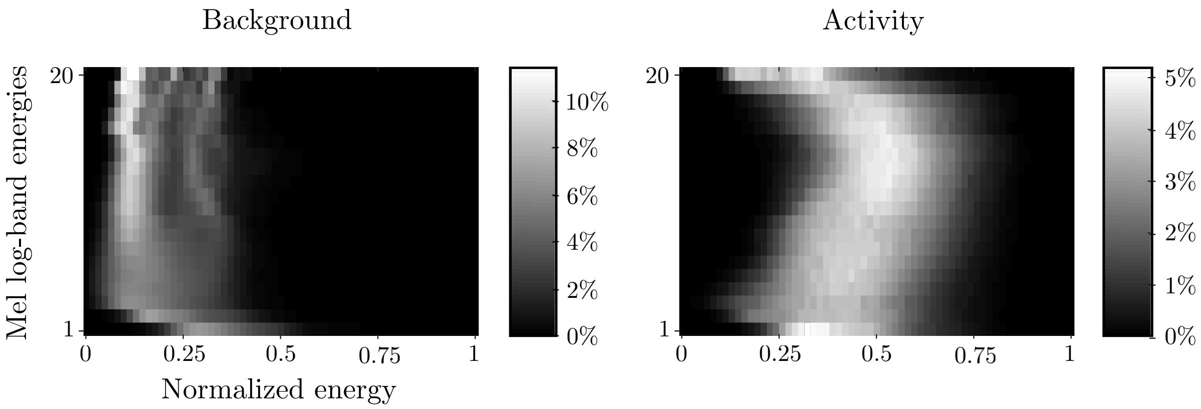


Figure 4.5: Histogram of 20 normalized Mel log-band energies with 11.6 ms frames and 5.8 ms hop length. Background represents frames labeled as ‘0’, while Activity represents frames labeled as ‘1’.

ground noise without explicitly pointing out the type of source that produced the events. Figure 4.5 shows the histogram of 20 normalized Mel log-band energies with 11.6 ms frames and 5.8 ms hop length, where Background represents frames labeled as ‘0’, while Activity represents frames labeled as ‘1’. The histograms clearly show that the impulsive audio events have an overall higher energy in all bands. Note that, the artifacts appearing in the higher bands of the Background histogram are the result of the reverberation of the target events not being included in the activity labels.

4.5.2 Description of the Experiments

The features are obtained from the STFT of the signals included on the database with a frame length $L = 512$ and 50% overlap, resulting in a hop time of approximately 5.8 ms. For every frame of the STFT 20 log Mel-band energies are computed as described on section 4.2. In order to test the validity of the proposed time context scheme, the results are presented with three different set of features containing: 5-frame context, resulting in a feature vector length $F = 100$, 3-frame context, resulting in $F = 60$, and 9-frame compressed context with ratios $\mathbf{c}_R = [1, 2, 2, 4, 4, 4, 10, 10, 20]$, adding up to a feature vector of length $F = 60$. The features are normalized (between 0 and 1) according to the mean and standard deviation of the training set

Our objective is to test the detection system in as close to real conditions as possible. This implies that, for any given event, both the source location and the receiving microphone are going to be new to the system (i.e., not included in the samples used to train the detector). Since our database is not large enough for the complexity of the problem, a simple division of the data into two equally sized train set and test set is not efficient. In order to obtain the presented results we used a cross validation technique known as Leave-One-Out Cross-Validation (LOOCV) [Efr79]. In LOOCV a single observation is used as the test set and the remaining observations $(O - 1)$ as the training set. The process is repeated O times, until every observation has been tested, and then, the results are averaged to obtain the final performance metrics. In this particular application, one observation is considered as the signal recorded by one microphone during the time window corresponding to one source location. Therefore, $O = 12 \times 16 = 192$ observations. Note that the database contains two recording sessions that share identical microphone and source locations, thus, there are $5 \times 2 = 10$ impulsive events per observation. This way, at every iteration of the LOOCV, the database is divided into a test set that contains the signal captured by the microphone under test, during the tested source location time window; and a train set that contains the bulk of the database: the signals captured with the remaining 11 microphones, excluding the time window in which the tested events are generated (i.e., 15 of the locations plus background sound).

By using this training scheme, we can assess how the results of the detection will generalize to a larger dataset. More importantly, since the classifier of each microphone (i.e., channel) is trained using a specific train set designed to isolate it, the results obtained on a multichannel setup can be considered independent to one another. Note that for every microphone-source location combination, the training is done without using any information about the tested source location at the remaining microphones, therefore, it is fair to use decision fusion to obtain multichannel detection results.

The results for each microphone are obtained from the concatenated outputs of the 16 source-location specific detectors. Each channel is then presented with $16 \times 5 \times 2 = 160$ impulsive events of interest. Multichannel results are obtained by averaging all the possible combinations of $M = 1, \ldots, 12$ channels. Multichannel detection is implemented as described on section 4.4 with a detection window $T_D = 20$. Prior to the averaging, the amplitude of the individual outputs is clipped to the interval $[0, 1]$. For any number of channels, the thresholded output (i.e., decision vector) is post-processed with a median filter (3 frames window) in order to smooth it, so as to avoid spurious detections.

The evaluation metrics for the detection are event-based and are calculated using an onset-only condition with a collar of 150 ms (average event label length). That is, a given detection is considered a TP if the difference between the detection time and the onset label of the event (true ToA) is smaller than the collar time. Multiple detections of one event are considered a single detection as long as they fall within the allowed collar time (i.e., refractory period). Any further detection of an event outside this time window is considered a FP. This way, a single acoustic event can produce one TP and various FP. Consequently, if an event does not produce a suitable detection it is considered a FN. Also, since the evaluation metrics are event-based, TN are not being computed. Figure 4.6 shows a schematic representation of the evaluation process, in which one FN, one TP and two FP are generated. Please note that, in the multichannel results, true ToA is considered as the average time of the onset label of M microphones. Compared to a frame-based evaluation, event-based metrics offer a better representation of the behavior of the detection system in the presence of a series of well defined impulsive events, moreover, the intent is to design a detection system capable of producing a single detection per event.

For the experiments, we compared the performance of the proposed K-LDA classifier and two well established classifiers using the three different set of features. The classifiers are a RndF using $N_T = 16$ trees, and a fully connected feedforward ANN, or MLP, with a number of neurons per layer $N_L = [20, 10, 1]$, trained for 300 epochs, with 0.5 parameter regularization, and assigning an error weight $E_w = [1, 1000]$ to the train set samples in order to counter class imbalance. The results of the K-LDA were obtained for various values of K. During the experiments, the exact same data set divisions were used for every classifier tested.

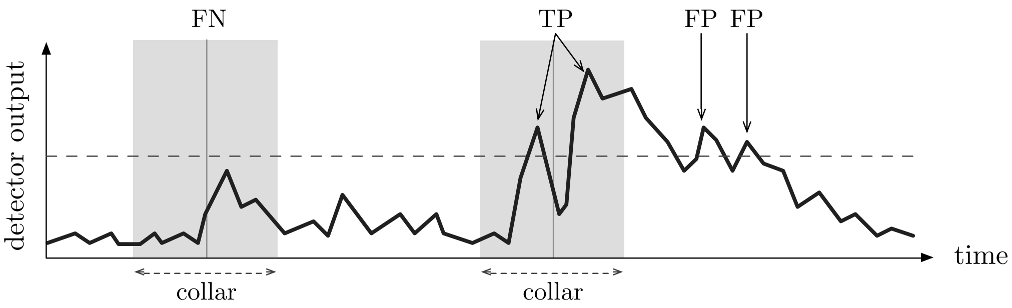


Figure 4.6: Schematic representation of the event-based evaluation metrics.

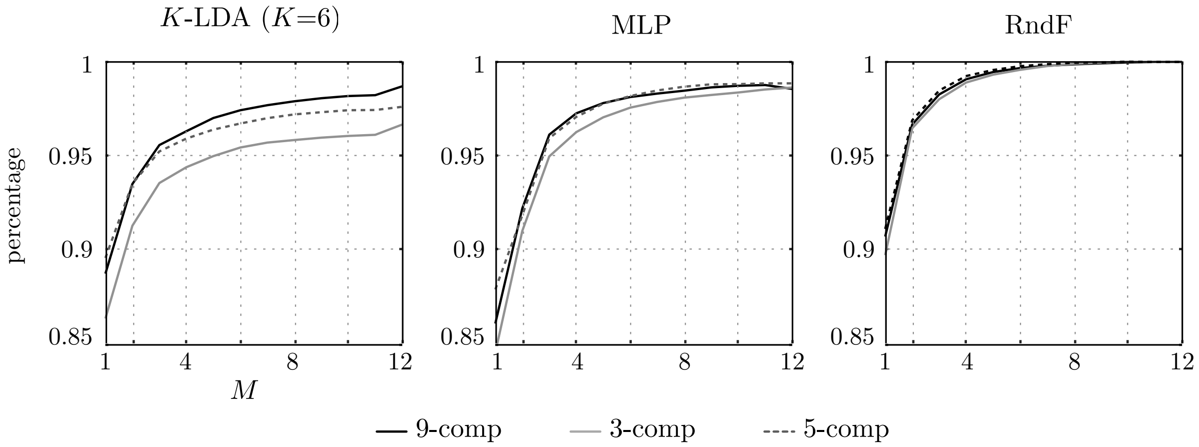


Figure 4.7: F1-Score of various detectors for different number of microphones (M) using 3 different temporal contexts.

4.5.3 Experimental Results

Tables 4.1 and 4.2 show the F1-score obtained with various detectors for different number of microphones (M) and using 3 different temporal contexts on the feature vector, namely: 3-frame and 5-frame full context, and the proposed 9-frame compressed context. Table 4.1 shows the F1-score of the proposed K-LDA with $K = 2$ and $K = 6$, while Table 4.2 shows the F1-score of the MLP and the RndF. In all cases the detection threshold is fixed at $y_0 = 0.5$. For the sake of clarity the most relevant results from the tables are represented graphically on Figure 4.7

From these results it is clear that all the detectors, except the K-LDA with $K = 2$, are capable of achieving a remarkable performance with any of the tested temporal contexts. The best results of the MLP and the RndF are obtained with 5-frame full temporal context, which seems reasonable since the number of features is much larger than that of the other two cases ($F = 100$ vs $F = 60$). Although the K-LDA detector obtains its best performance with the proposed 9-frame compressed temporal context ($F = 60$). In general, for most multichannel setups with any of the detectors, the proposed temporal context outperforms the classic 3-frame full context with an equal feature vector length ($F = 60$) in both cases.

From the results in tables 4.1 and 4.2 it may seem that using multichannel detection has little benefit given the elevated F1-score when only one microphone is considered. However it is worthwhile to explore the results a little further to see the true impact of spatial diversity on the detection.

Table 4.1: F1-Score of the proposed K-LDA detector (K=2 and K=6) for different number of microphones (M) and using 3 different temporal contexts.

	***K*-LDA** $(K = 2)$			**K-LDA** $(K = 6)$		
M	9-comp	3-full	5-full	9-comp	3-full	5-full
1	93.5% ±2.1	56.0% ±5.8	77.1% ±4.1	88.7% ±3.3	86.3% ±4.1	89.5% ±3.9
2	93.4% ±1.7	45.3% ±11.1	71.3% ±5.5	93.5% ±2.7	91.3% ±3.2	93.5% ±3.7
3	93.5% ±2.0	38.5% ±14.1	67.8% ±7.5	95.6% ±2.1	93.5% ±3.7	95.2% ±3.6
4	93.4% ±2.4	33.6% ±16.4	64.8% ±9.8	96.3% ±2.1	94.4% ±4.0	95.9% ±3.7
5	93.4% ±2.8	29.9% ±17.9	62.6% ±11.8	97.0% ±2.2	95.0% ±4.4	96.4% ±3.8
6	93.4% ±3.2	27.2% ±19.0	61.0% ±13.5	97.4% ±2.3	95.4% ±4.8	96.7% ±3.9
7	93.5% ±3.5	25.2% ±19.8	59.5% ±14.9	97.7% ±2.3	95.7% ±5.1	97.0% ±3.9
8	93.6% ±3.8	23.6% ±20.4	58.4% ±16.1	97.9% ±2.4	95.8% ±5.3	97.2% ±4.0
9	93.6% ±4.1	22.3% ±21.0	57.6% ±17.3	98.0% ±2.6	95.9% ±5.5	97.3% ±4.1
10	93.5% ±4.4	21.4% ±21.4	57.2% ±18.6	98.2% ±2.7	96.0% ±5.7	97.4% ±4.2
11	93.6% ±4.7	20.5% ±21.9	56.4% ±20.6	98.2% ±2.7	96.1% ±6.1	97.4% ±4.3
12	93.7% ±5.3	20.0% ±22.4	55.3% ±22.6	98.7% ±3.1	96.6% ±6.5	97.6% ±4.5

Table 4.2: F1-Score of the MLP and the RndF for different number of microphones (M) and using 3 different temporal contexts.

	MLP $(N_L = [20, 10, 1])$			**RndF** $(N_T = 16)$		
M	9-comp	3-full	5-full	9-comp	3-full	5-full
1	86.1% ±8.8	84.5% ±9.7	87.9% ±8.5	90.7% ±3.1	89.7% ±2.8	91.1% ±3.0
2	92.2% ±5.2	91.0% ±6.3	91.8% ±5.9	96.6% ±1.6	96.4% ±1.5	96.9% ±1.3
3	96.1% ±3.9	94.9% ±4.5	95.9% ±4.1	98.2% ±1.0	98.0% ±1.1	98.5% ±0.9
4	97.2% ±3.1	96.2% ±3.5	97.0% ±3.1	99.1% ±0.6	98.9% ±0.8	99.2% ±0.5
5	97.8% ±2.8	97.0% ±3.0	97.8% ±2.5	99.4% ±0.5	99.3% ±0.6	99.5% ±0.4
6	98.1% ±2.5	97.5% ±2.8	98.2% ±2.3	99.6% ±0.4	99.6% ±0.4	99.7% ±0.3
7	98.3% ±2.3	97.8% ±2.6	98.5% ±2.1	99.8% ±0.4	99.8% ±0.3	99.9% ±0.2
8	98.4% ±2.2	98.1% ±2.6	98.7% ±2.0	99.8% ±0.3	99.9% ±0.1	99.9% ±0.1
9	98.6% ±2.0	98.2% ±2.6	98.8% ±1.9	99.9% ±0.2	99.9% ±0.1	100.0% ±0.1
10	98.7% ±2.0	98.3% ±2.5	98.8% ±2.0	99.9% ±0.1	100.0% ±0.0	100.0% ±0.0
11	98.7% ±1.9	98.5% ±2.5	98.8% ±2.0	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0
12	98.5% ±2.2	98.6% ±2.7	98.8% ±2.1	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0

Table 4.3: TPR (white background) and FPR (grey background) of various detectors for different number of microphones (M) and using 2 different temporal contexts.

M	**K-LDA** ($K = 6$)		**MLP** ($N_L = [20, 10, 1]$)		**RndF** ($N_T = 16$)	
	9-comp	5-full	9-comp	5-full	9-comp	5-full
1	96.5% ±2.6	95.9% ±3.3	99.8% ±0.5	100.0% ±0.0	99.6% ±1.0	99.7% ±0.7
	21.3% ±7.1	18.5% ±8.2	34.7% ±25.4	29.7% ±23.3	20.3% ±7.5	19.4% ±7.2
2	96.0% ±3.5	95.4% ±4.8	100.0% ±0.1	100.0% ±0.0	99.8% ±0.5	99.9% ±0.4
	9.4% ±3.1	8.7% ±4.1	17.6% ±12.7	18.7% ±14.6	6.8% ±3.4	6.4% ±2.4
4	96.5% ±4.2	95.6% ±5.7	100.0% ±0.0	100.0% ±0.0	99.9% ±0.3	100.0% ±0.2
	3.9% ±1.1	3.6% ±2.1	5.9% ±6.9	6.3% ±6.8	1.8% ±1.1	1.5% ±1.0
8	97.0% ±4.8	95.7% ±7.0	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0
	1.1% ±0.8	1.0% ±1.0	3.3% ±4.6	2.8% ±4.2	0.3% ±0.6	0.1% ±0.2
12	97.6% ±5.7	95.6% ±8.1	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0	100.0% ±0.0
	0.0% ±0.0	0.0% ±0.0	3.1% ±4.7	2.4% ±4.4	0.0% ±0.0	0.0% ±0.0

Table 4.3 shows the True Positive Rate (TPR) and the False Positive Rate (FPR) of three of the tested detectors. with two temporal contexts. Note that the offered metrics are event-based, hence, TPR represents the number of detected events in relation to the total number of events, while FPR represent the number of FP in relation to the total number of events (i.e, random FP plus multiple detections of one event). The table shows that fusing multiple detections does not justify the effort in terms of TPR, however, it greatly reduces FPR in all cases, obtaining an overall drop of around 20%. This effect can be explained by the influence of decision averaging. While the chosen decision fusion technique can not produce a global positive when most of the outputs are lower than the threshold, it is very effective at reducing the impact of spurious detections in some of the channels. The obtained results indicate that decision fusion is a valuable tool for obtaining a robust impulsive event detection system capable overcoming spatial uncertainty. It is important to highlight that, when the 12 microphones are combined with either temporal context, the results are remarkable. The RndF is capable of perfect detection, while the proposed detector obtains a 0% FPR and the MLP a 100% TPR.

Although not shown, it is worth mentioning that a regular LS-LDA was also tested. We opted for omitting its results since it completely failed to achieve an adequate performance. Upon inspection, the cause of its lackluster performance is that the output level of the detector rarely reaches the threshold, furthermore, by combining multiple channels the problem is worsened. Hence, for the LS-LDA detector to work, the threshold must be lowered. This can be explained due to the extreme class imbalance of the database (less than 2% of the frames are labeled as true) which the LS-LDA is not capable of overcome, since it assigns the same weight to each observation during training.

In relation to database imbalance, the performance of the proposed detector in Table 4.1 shows the importance of the clustering stage. Compared to a regular LS-LDA, the K-LDA with $K = 2$ and the proposed temporal context obtains much better results (with $y_0 = 0.5$). In this particular case, the improvement is due to K-means clustering doing most of the detection, that is, the clustering stage is capable of separating target events and background well enough that class imbalance

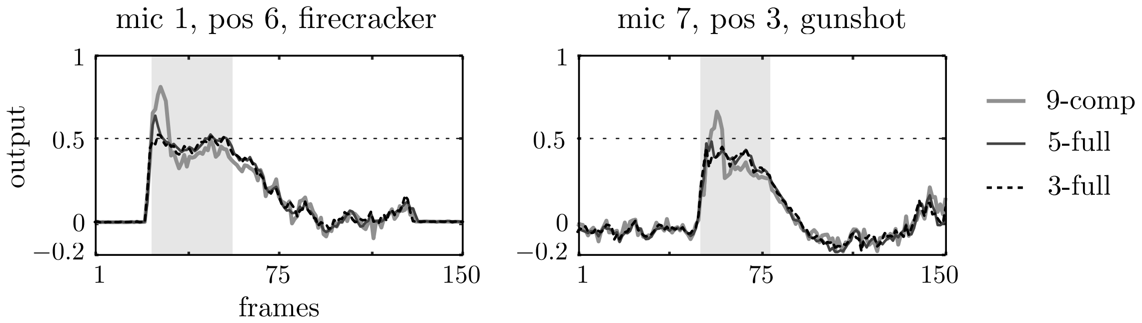


Figure 4.8: Probabilistic output of the K-LDA detector with $K = 2$ for two acoustic events using three different temporal contexts. Labelled frames shown as a gray area.

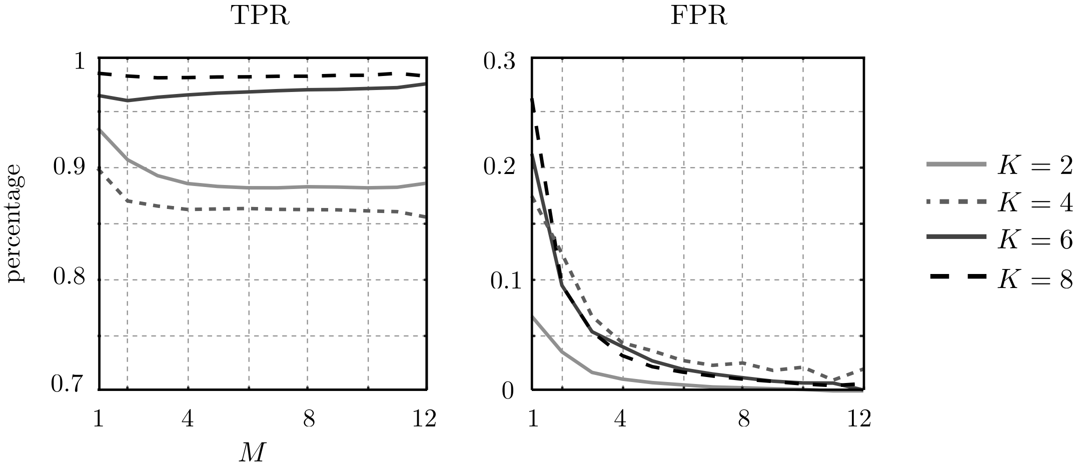


Figure 4.9: Multichannel detection TPR and FPR for various values of K using the K-LDA classifier with 9-frame compressed feature context.

becomes much less of an issue. This way, the specialized LS-LDA detectors trained on samples belonging to one of two clusters are capable of 'refining' the class separation. However, the ability of K-means clustering to separate the classes depends on the chosen feature set. Figure 4.8 shows the differences in the probabilistic output of the detector when the feature vector is changed. From the figure, is possible to see that clustering ($K = 2$) using the proposed temporal context is better at separating the "attack" of the impulsive events from the main body of the database, as shown by the sudden drop in the detector's output. The bad performance of full temporal contexts on Table 4.1 for $K = 2$ can then be explained by the low level outputs, in the same was as that of the LS-LDA.

Again, the output level of the detectors plays an important role in the results. For $K = 2$ and $K = 4$, combining multiple microphones is decremental to the TPR since their low output level makes it harder to obtain an average higher than the selected threshold. Moreover, using a wrong number of clusters affects class imbalance which in turn is shown on the results. Figure 4.9 shows the influence of the number of clusters on TPR and FPR for different number of microphones using the proposed temporal context. The effect of class imbalance on the clusters can be seen when comparing TPR and FPR for 2 to 6 clusters, where $K = 4$ shows the poorest results due to the classes not being correctly captured by the clustering. These results also show that increasing the number of clusters past a certain point does not significantly increase the performance of the classifier.

In order to asses the effect of the threshold in the detection, we also computed the ROC curve

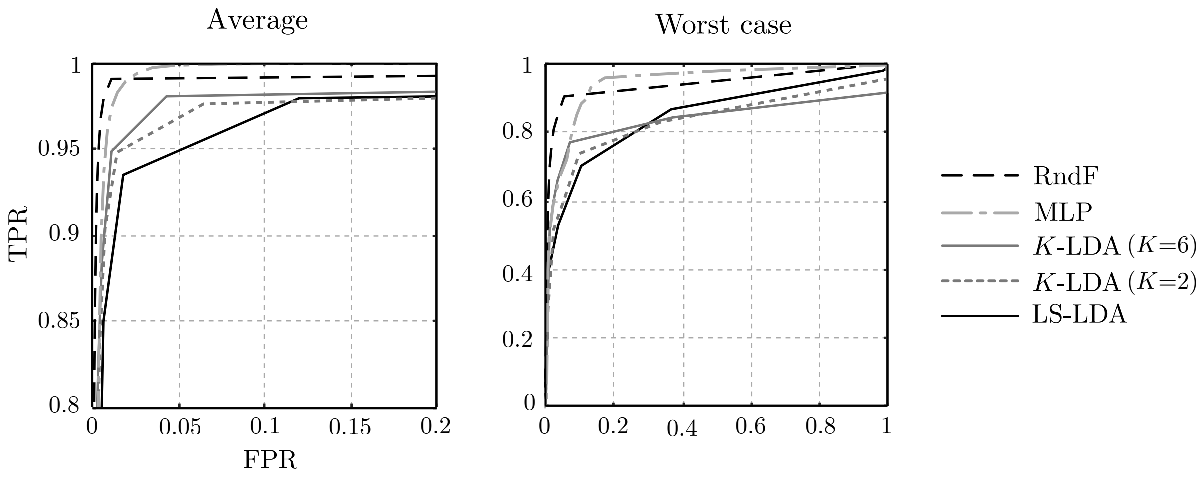


Figure 4.10: Average ROC and Worst-case ROC for various detectors using the proposed temporal context.

of some of the tested detectors. It is important to mention that in order to do so, the performance metrics had to be changed from event-based to frame-based. This is due to event-based metrics lacking a measure of the True Negative Rate (TNR) required for computing the FPR as used in the ROC. Please note that, with event-based metrics FPR refers to the number of FP in relation to the number of true events, while with frame-based metrics, FPR refers to the to the number of FP in relation to the number of frames labelled as false. We computed the ROC curve of various detectors with 9-frame compressed context in in two scenarios: *Average*, computed with the average TPR and FPR over all microphones and source locations; and *Worst-case*: computed with the worst TPR and FPR of all possible microphone-source location combinations. The curves were obtained by evaluating the frame-based performance of every classifier under both scenarios, by gradually increasing the threshold from 0 to 1 ($y_0 = [0, 0.05, \ldots, 0.95, 1]$).

Figure 4.10 shows the ROC curve of various detectors in both scenarios. Some important remarks can be made about these results. First, the ROC curve of the averaged detectors shows how good the detection is with any of the tested classifiers; even a LS-LDA with the correct threshold obtains a TPR of 95% with only a 5% FPR. Again, this may lead to see the problem as trivial, since the more complex detectors obtain an improvement that may not be relevant enough. However averaged results are somewhat representative of the results obtained by fusing the decisions of 12 detectors. Second, by looking at the Worst-case results a more proper representation of the spatial-dependence problem can be seen. Note that the scale of the Average scenario corresponds to that of the top-left grid quadrant in the Worst-case scenario. In this case the ROC curves show a large degradation mainly due to previously ‘unseen’ propagation paths. Third, ROC curves of both scenarios together with Tables 4.1 and 4.2 show that different classification techniques obtain different degrees of improvement from decision fusion. Generally, The better the classification of a single observation is, the lesser the impact of fusing multiple observation of the same event, and vice versa.

In conclusion, the proposed K-LDA classifier manages to improve the results of regular LS-LDA by taking advantage of K-means clustering. The results obtained with the proposed classifier are closer to that of more complex algorithms while having a lower computational complexity. K-LDA can be seen as an overall improvement upon traditional LS-LDA that is not limited to impulsive event detection. In addition to the classification algorithm, a compressed temporal context scheme for the feature vector was also proposed. The proposed approach outperforms the classic 'full-frame' approach with all the tested detectors when same feature vector length is used, and does so without adding significative computational load to the detection system. It is also important to mention that frame length was selected to be as low as possible (≈ 5.8 ms) in order to avoid the computation of large DFTs. A shorter frame length makes the spectral variations from frame to frame larger, thus providing some temporal context becomes a relevant matter in obtaining a low noise detection (i.e., free of rapid changes).

The presented results also show that detection of high SNR impulsive events is an easy task, although, the difficulty comes from being able to do so in diverse (and unknown) spatial conditions. While an isolated detector may not obtain an adequate performance in a spatially diverse scenario, from the results it is also clear that combining the decisions of a group of detectors is an efficient method to improve the global detection results. In general, the proposed impulsive sound event detection system offers a good performance while keeping the complexity of the algorithms involved as low possible, which makes it a good choice for automated surveillance on low-cost WASNs.

4.6 Complementary Results

This section presents some results intended to support the claims made about the proposed K-LDA classifier.

4.6.1 Computational Complexity

We claim that the proposed classifier has a lower computational complexity than the other tested algorithms, excluding the LS-LDA. To support this claim we have obtained the approximated number of clock cycles that each algorithm needs to evaluate one observation on an ARM Cortex-M4F[3] embedded processor. The selected platform is a low-cost, low-power 32 bit processor (up to 200 MHz) that includes floating-point arithmetic functionalities, making it ideal for WASNs.

The tested classifiers are the same used in the previous section, we consider $F = 60$ features in every case and $K = 6$ clusters for the K-LDA. The average number of nodes and and the average tree depth of the RndF were obtained empirically from the classifiers trained in Matlab, resulting in approximately 5800 nodes and 36 levels for any given decision tree. The path length of the decision taken for a given observation on a given decision tree may be greater or smaller than the average tree depth, but for simplicity, we consider every path length equal to the average tree depth in the calculations.

The computational complexity was computed assuming single-precision floating-point format (32 bits), and the number of clock cycles per operation was obtained from the ARM Cortex-M4 technical manual [ARM10]. Please note that fetching a value from RAM memory takes 1 cycle, while fetching a value from Program Memory (PM) takes 2 cycles, and that, contrary to DSPs, loops can not be advanced in a single clock cycle and have a 3 cycle overhead. Additionally, there is no indication of the number of cycles required to perform an exponentiation operation, so we considered it twice

[3] ARM Cortex-M4F is the denomination for processors in the M4 family that include a floating point unit coprocessor.

Table 4.4: Estimated number of clock cycles (cc) on a Cortex-M4F for various functions. Looped operations indicated by 'xN', where N is the number of iterations.

$y = b + \sum w_n x_n$		$y = \sum (w_n - x_n)^2$		$y = \mathrm{argmin}(x)$		$y = \frac{e^x}{e^x+1}$		$y = \mathrm{node_B}(x)$	
operation	cc	operation	cc	operation	cc	operation	cc	operation	cc
fetch b	2	loop overhead	3N	$z = 0$	1	fetch x_n	1	fetch n	2
$y = b$	1	fetch w_n	2N	loop overhead	3N	$a = e^x$	28	fetch z	2
loop overhead	3N	fetch x_n	1N	fetch x_n	1N	$b = a{+}1$	1	fetch B_1	2
fetch w_n	2N	$z = w_n{-}x_n$	1N	$x_n < z$	1N	$y = a/b$	14	fetch B_2	2
fetch x_n	1N	$y = y{+}z^2$	3N	if-else	1N			fetch x_n	1
$y = y{+}w_m x_n$	3N			$y = n$	1N			$x_n > z$	1
								if-else	1
								$y = \mathrm{B}_2 \vee \mathrm{B}_2$	1

Table 4.5: Estimated number of clock cycles of the tested classifiers ruining on a Cortex-M4F for various implementations.

Classifier	Looped	Looped	Unrolled	Unrolled
	PM load	RAM load	PM load	RAM load
LDA	543	483	363	303
K-LDA	4186	3766	2908	2488
MLP	12828	11418	8598	7188
RndF	8640	-	6912	-

as expensive as the most expensive single operation described in the manual (division and square root). Table 4.4 shows a breakdown of the estimated clock cycles of the various functions required to implement the tested classifiers.

It is also important to mention that the Cortex M4 family has a maximum RAM size of 512 kBytes, and a maximum PM size of 2 MBytes (depends on the manufacturer and model), thus, memory pressure is a relevant issue. Considering that branching and feature indexes of a given decision tree node are stored in 16 bit unsigned integer format, just storing all the information required to implement the RndF takes: $16 \times 5800 \times (32 + 3 \times 16)/8/1024 \approx 906$ kBytes, while storing the weights for the MLP and the K-LDA takes 5.6 kBytes and 2.8 kBytes respectively. This way, the efficiency of both the MLP and the K-LDA can be improved by moving their weight values to RAM (i.e., 1 clock cycle per weight fetching). In addition to this, the manufacturer advises to implement critical processes as 'unrolled loops', that is, a single large block of code listing all the individual operations. Unrolling the execution loops saves the 3 clock cycle loop overhead, further increasing efficiency at the cost of a larger program size.

Table 4.5 shows the estimated number of cycles required to implement the four tested algorithms considering: Looped execution with PM loaded parameters, Looped execution with RAM loaded parameters, Unrolled execution with PM loaded parameters, and Unrolled execution with RAM loaded parameters. The results clearly show that the proposed algorithm is considerably lighter than a MLP or a RndF, specially on the most efficient implementation. In fact, the increased number of clock cycles of the K-LDA with respect to a regular LDA is a direct result of computing to which

Table 4.6: Main attributes of 6 datasets from the UCI repository.

Dataset	Classes	Features	Samples	Class Balance
Iris flower	3	4	150	0.333/0.333/0.333
Adult income	2	14	32561	0.759/0.241
Wine origin	3	13	178	0.331/0.399/0.27
Car evaluation	4	6	1728	0.222/0.04/0.7/0.038
Breast cancer	2	9	699	0.646/0.354

of K clusters an observation belongs to. RAM loaded parameters with the RndF are not considered since it size in almost twice as large as the maximum RAM size of the cortex-M4. Please note that the presented results are intended as an indicative measure of the relative computational complexity of the tested classifiers, and that true clock cycle count can differ from the obtained estimates. Likewise, the presented figures are subjected to the target platform.

4.6.2 Evaluation of the K-LDA classifier

In order to further address the validity of the proposed K-LDA classifier, we tested its performance with the 5 most popular datasets available on the UC Irvine machine learning repository [DKT17]. Table 4.6 shows the main attributes of the selected datasets including their name, number of classes, number of features, number of samples, and percentage of samples per class (class balance). We tested 5 classifiers on each dataset, namely: a LS-LDA, the proposed K-LDA (using K clusters), a MLP (number of neurons per layer indicated by N_L), a RndF (with N_T trees) and a K-NN (using the K nearest points).

The results are evaluated in terms of accuracy. They were obtained using 'repeated random subsampling validation'. Each time a dataset-classifier combination was tested, the available samples were randomly divided in two equally sized, disjoint and equally balanced[4] train and test subsets. The split and test process was repeated 100 times, and then the results were averaged to obtain the final performance metrics. All the classifiers were tested using the same dataset division at each iteration. The testing was done in Matlab, all the classifiers, excluding the proposed one, are implemented with Matlab's built-in functions. Unless otherwise stated, the employed functions are set to their default configuration.

Table 4.7 shows the accuracy (and its standard deviation) obtained with the 5 classifiers on the 5 datasets, as well as the configuration used for each classifier. The obtained results show that, on 3 of the 5 datasets, the proposed K-LDA classifier performs at a comparable level to some popular classifiers with larger computational complexities, furthermore, it obtained the best accuracy on 2 of the 5 datasets. Special mention must be made about some of the datasets. In the Wine dataset, the best classifier is the LS-LDA, which suggests that it is a linearly separable problem, thus, more 'capable' classifiers suffer from overfitting as shown by the standard deviation of their accuracy. Also, on the car evaluation dataset the proposed classifier fails to perform adequately, even with a large number of cluster. The Car dataset is derived from a simple hierarchical decision model, thus

[4] In those cases in which the number of samples per class is odd, the train subset contains one sample more that the test subset, only for that class.

Table 4.7: Accuracy of various classifiers on 6 datasets from the UCI repository, and the configuration used in each case.

Dataset	LS-LDA	K-LDA	RndF	MLP	K-NN
Iris	-	$K = 3$	N_T =8	N_L =[8,3]	$k = 3$
	83.3% ±3.3	**96.1**% ±1.9	94.9% ±1.9	95.9% ±2.2	96.0% ±1.6
Adult	-	$K = 4$	N_T =16	N_L =[14,6,1]	$k = 3$
	80.9% ±0.2	83.3% ±0.5	**85.3**% ±0.2	84.1% ±0.6	81.8% ±0.2
Wine	-	$K = 3$	N_T =8	N_L =[12,3]	$k = 3$
	97.8% ±1.6	96.8% ±1.7	95.1% ±2.7	96.1% ±2.1	95.1% ±1.9
Car	-	$K = 8$	N_T =8	N_L =[12,6,4]	$k = 1$
	70.1% ±0.0	76.2% ±1.9	**91.8**% ±1.1	80.4% ±6.2	87.2% ±1.0
Cancer	-	$K = 2$	N_T =16	N_L =[18,9,1]	$k = 3$
	95.6% ±0.9	**96.4**% ±0.7	96.2% ±0.8	96.2% ±0.8	96.1% ±0.7

decision trees are specially well suited to this problem as shown by the accuracy of the RndF. It is also worth to mention that, LS-LDA is an ineffective classification method when there is a large class imbalances. Consequently, the worst performance of the proposed K-LDA classifier is found on the 2 most imbalanced datasets (Adult and Car).

4.7 Conclussions

In this chapter, an efficient impulsive sound event detection system for WASNs was presented. The detection system is rooted on a novel classification algorithm based on LS-LDA and K-means clustering, that takes standard audio analysis features with added temporal context as inputs. Each feature vector contains multiple past frames stacked using an compressive scheme that avoids a large increase of the dimensionality of the problem. The individual detections of a group of spatially diverse nodes equipped with the proposed detection system is then combined using decision fusion in order to increase the robustness of the system.

The proposed K-LDA classifier manages to improve the results of regular LS-LDA by taking advantage of K-means clustering to segment the feature space in order to train a specialized classifier for each of the K clusters. Feature space segmentation can be done in multiple ways, clustering was selected since it is a well-known unsupervised learning method that does not require any *a priori* knowledge of the data. K-means in particular is rather easy to implement and apply even on large data sets, furthermore, assigning an observation to one of the clusters is very simple.

Computational complexity was also considered in terms of feature vector length. Instead of using the standard full-frame stacking scheme for added temporal context, a modified version is proposed. The proposed temporal context scheme progressively reduces the length of past frames by computing a linear combination of their features, so that the total length of the feature vector does not increase linearly with the number of included past frames. The presented results show that compressed temporal context outperforms the classic one when the same feature vector length is used.

The proposed detection system not only minimizes computational complexity, but also data transmission. Considering that the probabilistic outputs are sent in single-precision floating-point format (32 bits), the required data rate during a detection cycle of the proposed multichannel detection scheme (using the parameters described in the experiments) is approximately 65 kbit/s, ignoring overhead. This figure can be halved just by using a shorter data format such as 16 bit fixed-point, and since the output range is defined, it is a very reasonable option. In any case the required data rate is perfectly assumable for current wireless technology. It is also important to remember that minimizing transmissions implies minimizing energy consumption.

It is important to mention that most work on impulsive 'noise' detection, does not take into consideration the spatial aspect of the problem, when actually, it is one of the most relevant issues. Moreover, detecting a high SNR impulsive event is an easy task, the difficulty comes from being able to do so in multiple (unknown) spatial conditions. The results presented in this chapter take into consideration the influence of the propagation path, hence, the detectors are tested with 'new' microphone and source locations not included in their training set.

Although the performance of the detection is remarkable (for all tested detectors), the possibility of having transmission errors or faulty sensors was not considered. Likewise, every sensor received the target events cleanly, with a high SNR and there are none other high energy sources in close proximity to the array. Thus, the presented results represent close to ideal conditions. Future work will then address less favorable conditions, that is, presence of transmission errors and/or faulty sensors, lower SNR and further non-idealities in the recordings, such as saturation, non line of sight, heterogeneous microphones, etc.

4.8 Summary

The main contributions described in the present chapter of the thesis are the following:

1. A simple method to increase the temporal context available on a single feature vector without excessively increasing its length, that is based in linear combinations of past frames.
2. An efficient classification algorithm that combines K-means clustering and LDA classification to obtain a piecewise linear classification boundary, that is capable of obtaining a competitive performance with limited computational complexity.
3. An efficient solution to the data fusion problem for an ensemble of detectors that uses the first local detection as the starting point for the multichannel fusion.
4. A multichannel impulsive sound event detection scheme capable of obtaining an adequate performance on a spatially diverse scenario, suitable for a WASN with restricted technical capabilities.

The contributions described in this chapter have been submitted for publication (currently under review). Further details can be found in [P4] in section: Publications.

CHAPTER 5 Impulsive Source Localization

5.1 Introduction

One of the most classic problems tackled in multichannel sound signal processing is source localization, thus, it frequently appears in the literature starting from a good number of decades ago to this day. Ongoing interest in sound source localization is just natural given that it is a vital part of many real world applications. These applications include some obvious ones, where source localization takes the spotlight, such as sniper localization systems, but it is also commonly present in less evident applications, such as most beamformer-based speech enhancement systems and even echolocation.

As it was already mentioned in the previous chapter, most interest in WASNs is generated by automated surveillance systems. Coincidentally, for many applications, event detection is not the main objective but a prerequisite to locate the source of the sound. Typically, the localization algorithm is subjected to previous event detection, just to ensure that the sound being located is in fact sound emitted by a target source and not some random noise. Some notable examples of this trend of research in WASNs include gunshot (sniper) or seismic localization systems, and vehicle tracking systems.

Localization algorithms tend to be more computationally intensive than other multichannel algorithms (excluding self-localization). The most typical approaches to sound source localization involve the computation of multiple cross-correlations and also the resolution of a minimization problem in order to produce a location estimate, which when put together require considerable computing power to be solved. Although, unless the localization process is intended as part of a source tracking system (i.e., continuous localization), there is no need for real-time execution, and obtaining the results within a reasonable time frame suffices.

As with almost any other algorithm, implementing source localization on a WASN has a series of problems that are not present in the classic ‘wired and centralized’ implementations. The most pressing issues in this case are: node synchronization, and perhaps more importantly, long inter-microphone distances. Thus, new solutions and adaptations of classic methodologies are needed for their successful implementation on a WASN.

5.2 GCC-based Localization

The typical approach to acoustic source localization when the location of the microphones is known and they share a common timebase, is to estimate the TDoA of every available microphone pair using the GCC. There is a good number of techniques that take a set of TDoAs and microphone

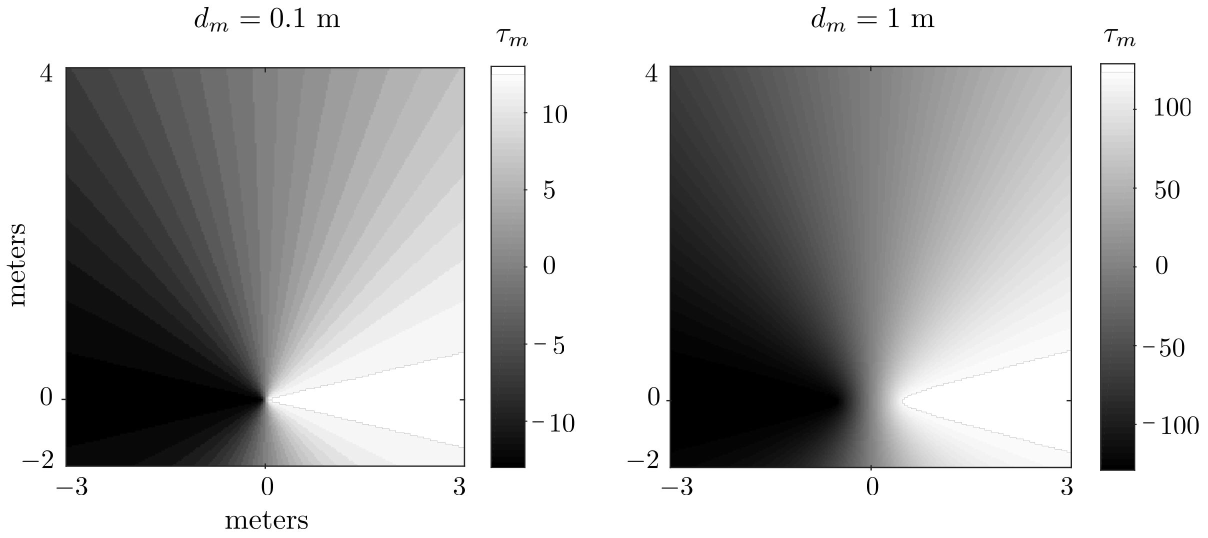


Figure 5.1: TDoA resolution for two inter-microphone distances at $f_s = 44100$ Hz. Array center at the origin of coordinates. Scale is TDoA in samples.

locations as inputs, and produce a source location estimation by solving a minimization problem. Depending on the particular method, the objective function may change but the concept remains the same, that is: given the microphone locations, find the point $\mathbf{p}_s$ in search space $\mathbf{P}$ that best predicts the estimated TDoAs. Some of the most popular source localization techniques that fit this criteria are triangulation, multilateration and SRP.

In general terms, the greater the distance between microphones and the more diverse their locations, the better the localization for a wider target area. When we discussed DoA estimation in chapter 3, we saw that the range of potential TDoA values is restricted to a finite interval determined by the physical separation of the microphone pair, the sampling frequency and the speed of sound. Therefore assuming that f_s is fixed, TDoA resolution can only be directly increased by increasing the inter-microphone distance. Although, it is worthwhile mentioning that it is possible to virtually increase the resolution of TDoA estimation using peak interpolation on the cross-correlation maximum, or resampling the signals received by the microphones (the latter inducing a significative efficiency loss). Figure 5.1 shows the range of possible TDoA values on 2D space as a function of the distance between the microphones, where each shade in the scale represents a discrete TDoA value. This way, for each discrete TDoA value there is a region of space (defined as an area between two hyperbolic curves), along which the source can be located with equal likelihood. Thus, by combining multiple estimates obtained from multiple microphone pairs, the feasible area grows smaller and by increasing the spatial diversity of the microphone locations, the precision of the localization increases.

Following this train of thought, given the wide spatial coverage that can be achieved with a WASN, it seems like a good option to use such a device to locate distant sound sources. However, even if we assume that node locations have been found, and that synchronization has been achieved, source localization on WASNs involves the solution of some additional problems.

It is worth remembering that cross-correlation is a measure of the similarity of signals as a function of their relative time displacement, therefore increasing inter-microphone distance entails a larger signal length. Let d_m be the distance between some microphone pair and L the length of an audio frame captured by the microphones. The maximum TDoA of the microphone pair is directly proportional to d_m (in samples) and given by: $\tau_d = d_m f_s / c$, hence, frame length needs to be $L > 2\tau_{d+}$ to contain all possible TDoAs, where τ_{d+} is the maximum TDoA of any microphone pair in the array.

While minimum signal length is not commonly a problem for wired arrays, on large WASNs it has two main implications. On the one hand, correlating longer signals requires higher computational loads, and assuming that source localization is not a continuous process, more data transmissions. On the other hand, and perhaps more importantly, long distances entails coherence loss between the signal received at each microphone due to propagation effects and a other problems such as interference caused by random sources in close proximity to some of the microphones. This way, cross-correlating the signals of distant microphones is both more expensive, and more prone to produce erroneous results.

In an array of M microphones, the total number of microphone pairs is $\frac{M(M-1)}{2}$. If we consider a fully connected WASN with one microphone per node, it makes sense to distribute the computing load equally among the nodes, so that each node is in charge of computing $\lceil \frac{(M-1)}{2} \rceil$ cross-correlations maximum. It is good practice to have the nodes send the DFT of their signals (pre-weighted for weighted-GCC) in order to alleviate the computational load, but in any case, the number of element-wise vector multiplications and IDFTs per node grows linearly with M. This way, the cost of computing the GCC on large WASNs (both in physical size and number of nodes) quickly escalates since a sizable number of cross-correlation of long signals needs to be computed.

5.2.1 Long Inter-microphone Distance SRP

When the signals of two far apart microphones are to be cross-correlated, the influence of the propagation path together with the appearance of overlapped strong reflections (either from nearby objets or even the ground) can transform the received signals enough so that they can not longer be considered a time lagged version of one another. Even the the speed of sound, largely considered a constant, has a weak dependence on frequency and pressure due to the elastic properties of the propagation medium [Kie77], and although the effect is most often negligible, a slight deviation from ideal behavior can become noticeable over a large enough distance. Therefore, the time lag estimate produced by such a cross-correlation is often erroneous.

The most common way of dealing with undesired effects when cross-correlating audio signals is to use GCC-PHAT, specially in those situations in which reverberation is the main problem (i.e., indoor talker localization). Unfortunately, PHAT-weighting is not that good of a tool for outdoor impulsive sound source localization. PHAT-weighting works by whitening the signals, which can be seen as equalizing the amplitudes of every frequency so that only phase differences have an impact on the result. However, impulsive acoustic events already have a rather flat spectrum (specially in short frames) and phase differences between channels are heavily influenced by the propagation path (propagation effects such as wind and temperature gradients can produce frequency-dependent phase differences between the signals). Thus, PHAT-weighted cross-correlations do not show any significant benefit compared to standard cross-correlation.

Since we are expecting large errors in the TDoA estimates, we decided to use SRP localization

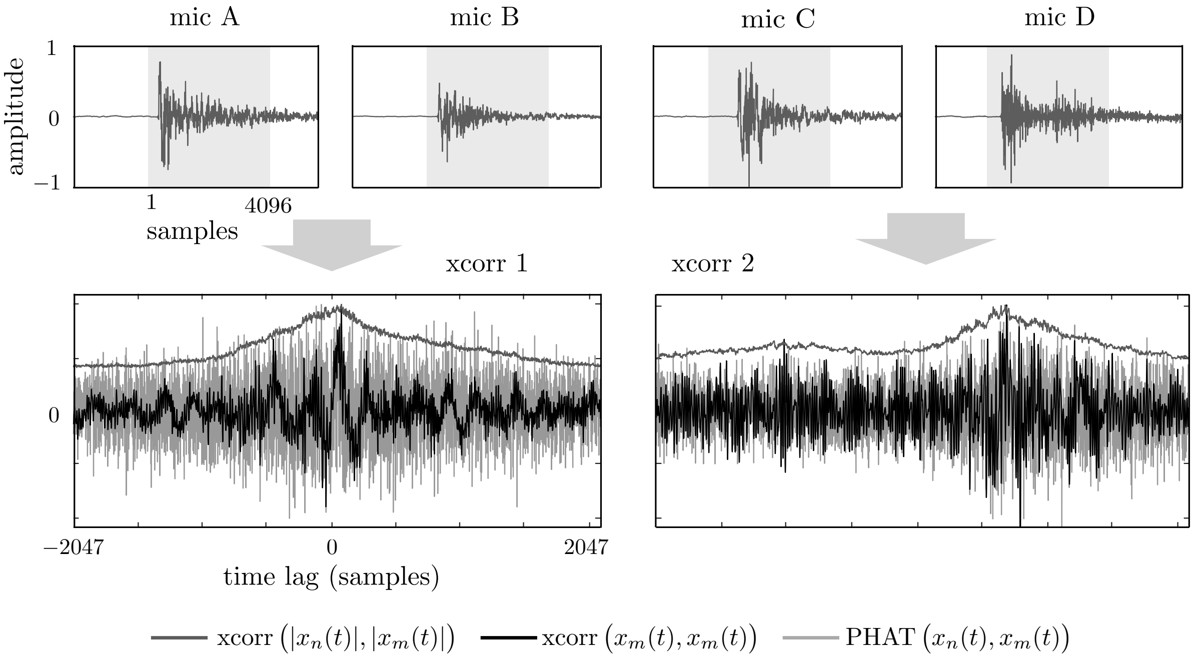


Figure 5.2: Two pairs of recordings of a firecracker explosion and their cross-correlation using different methods. The 4 recordings are from the same acoustic event taken at different locations.

given that instead of using a single estimate to find the source position it makes use of the 'whole' cross-correlation. Additionally, in order to avoid the problems arising from 'weak' cross-correlations we decided to cross-correlate the absolute value of the signals, so that:

$$r_{nm}(\tau) = \sum_{t=-\infty}^{\infty} |x_n(t)||x_m(t+\tau)|, \ \tau = 0, ..., \pm L \ , \tag{5.1}$$

This can be seen as cross-correlating the energy envelope of the signals, thus, in contrast to GCC-PHAT, the influence of phase differences is being reduced.

Figure 5.2 shows two pairs of recordings of a firecracker explosion and their cross-correlation using different methods. From the figure it is possible to see that the cross-correlations obtained with the proposed method look like those of the envelope of the signals, and that the standard and PHAT-weighted cross-correlations present very noisy results. While this kind of signal is expected from GCC-PHAT, in an ideal scenario the peak corresponding to the TDoA should be much higher, so that the cross-correlation resembles a perfect impulse. Although in many cases the maximum peak of the correlation does correspond to the actual TDoA, additional peaks corresponding to strong reflections are often present. When multiple cross-correlations are combined for obtaining the SRP surface, noisy signals in conjunction with small timing errors make for a very irregular surface, hence, its maximum rarely corresponds to the actual source location.

Although the proposed method can be deemed as counterproductive to obtain an accurate TDoA estimation, for SRP-based localization we are interested in the accumulation of 'probable' source locations obtained from multiple microphone pairs. SRP can be interpreted as a method to find the location that maximizes the output of a delay-and-sum beamformer, thus even when a single microphone pair can not precisely determine the spatial region where the source is located, the combination of multiple such coarse regions forms a smooth SRP surface.

5.3 Detection-aligned Cross-correlations

It is very likely for any localization system to be specialized in some type(s) of source(s), such as human speech, gunshots, vehicles, etc. The most common situation is then to have the localization process be a 'slave' to some sort of detection system. Without a specific target, the localization process will most likely be either triggered by random high-energy sources, or ran continuously, thus, producing results that quickly jump between dominant sources. The preferred method is then to only locate a sound when it is thought to have been produced by a target source. However, the influence of the detection system on the localization process is typically limited to taking the decision on when to locate a source. That is, some acoustic event detection, either local or global, causes the system to launch the source localization process using audio signal frames of length $L > 2\tau_d$ taken by every microphone at the same time instant.

In order to lessen the problems associated with the cross-correlation of signals acquired with large inter-microphone distances, we propose to use the impulsive sound event detection system described in the previous chapter to reduce the length of the signals being correlated, which in turn can improve the quality of TDoA estimation. The main idea is to use the time stamps associated to local detections to select the time interval of each signal frame, such that the full range TDoA of a given microphone pair can be obtained from the cross-correlation of their respective signal segments and their detection time stamps. We refer to the proposed method as detection-aligned cross-correlation.

Let us have a WASNs composed of M nodes, where every node contains one microphone. At this point we assume that the locations of the nodes are known, and that the network is synchronized. Considering that every node is capable of perfect event detection, we denote a local detection instant, or time stamp, as Γ_m, $m = 1, \ldots, M$. Let the audio signal captured by the m^{th} microphone be $x_m(t)$, where $t \in \mathbb{Z}$ is the time index. We can then define M signals of length L_s, such that: $\Gamma_m - L_{s1} < t \leq \Gamma_m + L_{s2}$, where L_{s1} and L_{s2} denote the start time and end time of the signal segment in relation to the detection instant, and $L_{s1} + L_{s2} = L_s$. Let $X_{m\Gamma}(k)$ be the L-point DFT of the m^{th} signal segment, so that the cross-correlation of any two segments is obtained with:

$$\tilde{r}_{nm}(\tau) = \sum_{k=0}^{L-1} X_{n\Gamma}(k) X^*{}_{m\Gamma}(k) e^{j\frac{2\pi}{L}k\tau}, \ \tau = 0, ..., L-1 \ . \tag{5.2}$$

Then, the relative TDoA o the n^{th} and m^{th} segments is simply found as:

$$\tilde{\tau}_{nm} = \arg\max_{\tau}(\tilde{r}_{nm}(\tau)). \tag{5.3}$$

Note that the tilde represents, both $\tilde{r}_{nm}$ and $\tilde{\tau}_{nm}$ being subjected to Γ_n and Γ_m, thus, the signals segments do not share a common time base and the obtained time lag is relative.

Let us denote τ_{nm} as the proper TDoA of some acoustic event, that is obtained from the cross-correlation of two signals that share a common time base and have a length $L_{d+} > 2\tau_{d+}$, where $L_{d+} \gg L_s$. If we consider both TDoA estimates to be free of error, it is then possible to obtain τ_{nm}

from $\tilde{\tau}_{nm}$ just by adding the difference between the detection instants as a bias, so that:

$$\tau_{nm} = \tilde{\tau}_{nm} + \Delta\tau_{nm}, \text{ with: } \Delta\tau_{nm} = \Gamma_n - \Gamma_m. \tag{5.4}$$

Proof of expression 5.4 can be easily traced back from the definition of TDoA as a difference of ToAs: $\tau_{nm} = \tau_n - \tau_m$. If we consider perfect detection time stamps, that is: $\Gamma_n = \tau_n$ and $\Gamma_m = \tau_m$, then: $\tilde{\tau}_{nm} = 0$, therefore $\tau_{nm} = \Gamma_n - \Gamma_m$. This way, whenever the local detection time stamps are precise enough, and L_s suffices to correctly estimate the TDoA; it is possible to bypass the theoretical minimum L and obtain full range TDoAs using detection-aligned signal segments of length $L_s \ll L$. However, most audio event detection systems work on a per-frame basis, hence the resolution of the detection time stamps is dictated by L_H, the frame hop length. This way, the resolution of Γ_m is limited, so far as it can only be obtained in L_H (samples) intervals. Please note that this fact has to be taken into consideration for the selection of L_s, as well as L_{s1} and L_{s2} in order to successfully obtain the TDoAs using the proposed method.

The effect of obtaining the TDoA using shorter detection-aligned signals is two fold: first, it minimizes the likelihood of a stronger source being contained within the signals, and second, in case of failing to correctly estimate the TDoA, the maximum error is reduced. Both these effects are closely related. When the signal emitted by a high-energy undesired source is overlapped with the signal emitted by the target source, it is very likely for the peak of r_{nm} to be caused by the undesired source. Hence, defining τ'_{nm} as the TDoA of the undesired source, the error induced by failing to obtain the desired time lag is: $\tau_{nm} - \tau'_{nm}$, consequently, the larger L, the larger the possible error.

In order to illustrate the possible benefits of using detection-aligned cross-correlations, let us set a simulated scenario where an 8-microphone array is used to locate a target source (s_1) in the presence of 2 additional noise sources (s_2 and s_3) with SRP-PHAT. The emission time of the sources is proportional to their index and their distance to the array is inversely proportional to it, that is, s_1 is the first source to emit and the most distant. Also, in relation to s_1, the signal emitted by s_2 is twice as powerful and 8 times shorter, while that of s_3 is half as powerful and equally long. The length of of the target signal is 4096 samples, thus standard framed cross-correlations need a frame length $L = 8192$ in order to contain it whole. We assume perfect detection, hence, the detection-aligned signals are obtained using the ToF as Γ_m and have a length $L_s = 256$. Figure 5.3 shows the 8 signals to be cross-correlated using framed cross-correlations and detection-aligned cross-correlations, where signal overlapping and the effect of source distance is clearly visible on the 'full' length signals. These 2 sets of signals are then used to estimate the location of s_1, using SRP-PHAT with a grid resolution of 0.1 m. Figure 5.4 shows a visualization of SRP-PHAT with both sets of signals over the whole search space, where circles are microphones, numbered crosses are sources, and diamonds are the estimated source location.

From the results it is clear that the standard approach fails to locate s_1 (it is the second largest peak) due to interference from other sources, while the proposed approach is unaffected due to reduced signal length. It is important to mention that this comparison is not completely fair. Although not shown in the figures, on this example, classic detection-based localization would employ signal segments selected around the global detection time (i.e., average ToF) with a length $L = 2048$ (the minimum power of 2 greater than the largest microphone pair time lag). Given the shorter time duration, s_2 only appears in the last quarter of the signals (approximately), thus the localization of

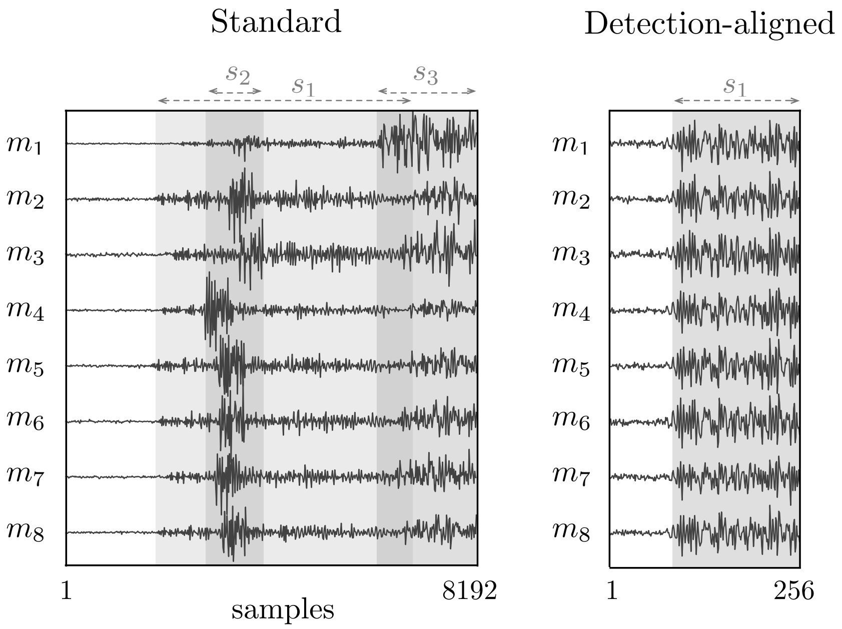


Figure 5.3: Example of signals to be cross-correlated (normalized amplitude) using two different approaches.

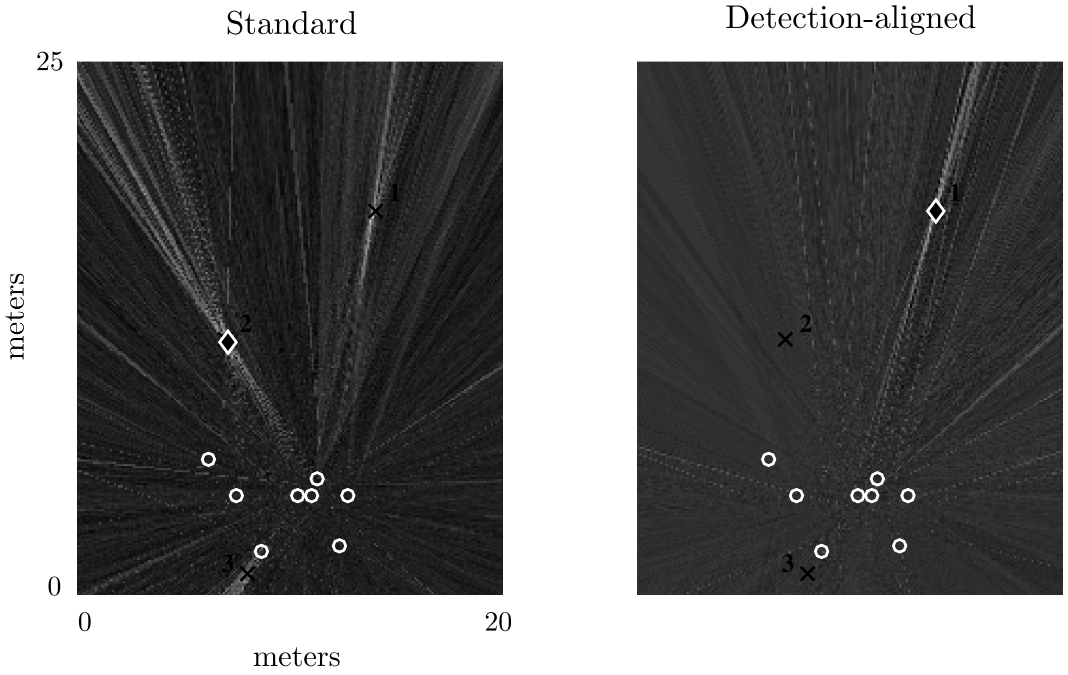


Figure 5.4: SRP-PHAT of the signals represented on Figure 5.3. Circles are microphones, numbered crosses are sources and diamonds are the estimated source location.

s_1 is successful. However, the brief appearance of s_2 suffices to impact the location estimate. The localization error obtained with detection-aligned signals and $L_s = 256$ is approximately 5 cm, while that of the standard detection-based approach with $L = 2048$ is approximately 17 cm.

Please note that these results should be seen as idealistic since perfect sample-accurate detection time stamps are used in both cases, and the signal propagation model does not account for non-linear effects which are a big concern in outdoor scenarios.

5.3.1 Detection Onset Estimation

Taking into consideration the detection system proposed in the previous chapter; in the event of a global detection, there is no guarantee of having a local detection on all channels. Moreover, even if there is a local detection, its frame index may not accurately reflect the ToA of the event. Thus in order for detection-aligned cross-correlations to be feasible, a solution must be found to estimate Γ_m at every channel.

Instead of relying on local detections, we propose to use the probabilistic output of the m^{th} detector to estimate Γ_m using curve fitting to find its onset[1]. This way, in the event of a global detection, Γ_m is estimated by fitting a step function to the probabilistic output of each local detector, regardless of whether or not they produced a local detection.

Let $y_m(l)$ be the probabilistic output of the m^{th} detector, and Γ be the frame index of a true global detection. We can then affirm that Γ_m is in the environment of Γ, so if we define a search space $\ell = \{l \in \mathbb{Z} : l + \Gamma - \frac{T_\Gamma}{2} < l < l + \Gamma + \frac{T_\Gamma}{2}\}$, where T_Γ is an odd integer that controls the length of a temporal window with Γ at is center, the probabilistic output of a given detector during that temporal window should ideally approximate a shifted unit step function, or Heaviside step function, given by:

$$u(l - o_m) = \begin{cases} 1, \text{ if: } l - o_m \geq \Gamma_m \\ 0, \text{ if: } l - o_m < \Gamma_m \end{cases}, \tag{5.5}$$

where o_m is the frame index of the m^{th} local detection, hence, on an ideal scenario $o_m = \Gamma_m$.

Curve fitting can be defined as the process of finding the parameters of a mathematical function (i.e., curve), that has the best fit to a series of data points, usually subject to constraints. Curve fitting is closely related to other optimization problems, consequently, by casting the fitting of $u(l - o_m)$ on $y_m(l)$ as a LS problem, a given detection onset can be estimated as:

$$o_m = \arg\min_{\ell} \Big(\sum_{l > \Gamma - \frac{T_\Gamma}{2}}^{l < \Gamma - \frac{T_\Gamma}{2}} \big(y_m(l) - u(l - o)\big)^2 \Big), \tag{5.6}$$

Estimating o_m is a simple problem that can be easily solved doing an exhaustive search. For the sake

[1] Onset is a term commonly used in audio signal processing to refer to the starting time of a sound event, in this particular case we are using it to refer to the rising edge of a detector's output.

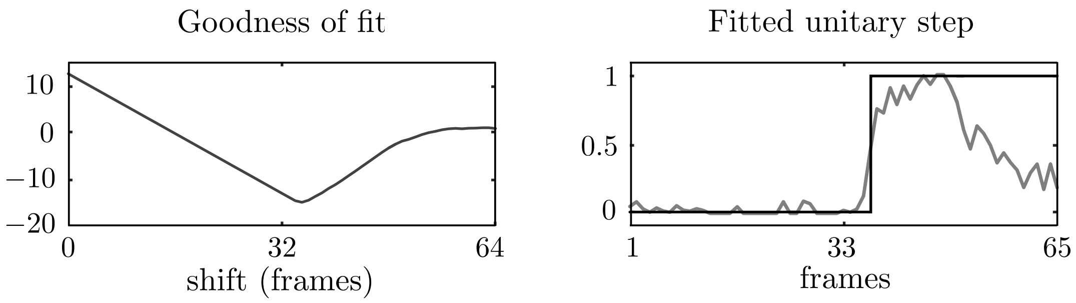


Figure 5.5: Example of unit step function fitting on a probabilistic detector output.

of clarity, let us define $t1, \dots, T_\Gamma$ as the frame index inside search space t, so that:

$$S = \sum_{l=1}^{T_\Gamma} \left(y_m(l) - u(l-o)\right)^2 \tag{5.7}$$

$$= \sum_{l=1}^{o} \left(y_m(l) - 0\right)^2 + \sum_{l=o+1}^{T_\Gamma} \left(y_m(l) - 1\right)^2 \tag{5.8}$$

$$= \sum_{l=1}^{o} y_m^2(l) + \sum_{l=o+1}^{T_\Gamma} y_m^2(l) + 1 - 2y_m(l) \tag{5.9}$$

$$= \Big(\sum_{l=1}^{T_\Gamma} y_m^2(l)\Big) + (N - o - 1) - 2\sum_{l=o+1}^{T_\Gamma} y_m(l), \tag{5.10}$$

where o is the frame index of the unit step function shift. The sum of $y_m^2(l)$ in expression 5.10 is a constant for any time shift, so it can be dropped without affecting the minimization. Then the estimation of o_m can be simplified to:

$$o_m = \arg\min_{\ell} \Big((N - o - 1) - 2 \sum_{l > \Gamma - \frac{T_\Gamma}{2}}^{l < \Gamma - \frac{T_\Gamma}{2}} y_m(l)\Big). \tag{5.11}$$

This is an efficient method of estimating o_m that only requires basic operations. It can be solved by doing an exhaustive search over ℓ without inducing a significant computational load. Figure 5.5 illustrates an example of the estimation of some detection's offset using expression 5.11 by showing the fitted unitary step, where 'goodness of fit' represents the value of the error function for every possible shift of the unit step function.

Note that the proposed method produces a robust estimation of Γ_m. Compared to simpler methods such as searching for the maximum value of $y_m(l)$ or the minimum l such that $y_m(l) > y_0$, the proposed method takes into consideration the likelihood of a detector's probabilistic output being "active" for a number of consecutive frames. This way, even if the output does not reach the detection threshold, or in the presence of sudden drops or spikes in its level, Γ_m is found by computing at which point $y_m(l)$ first became active.

5.4 Experimental Work and Results: GCC-based localization

This section describes the experiments conducted to evaluate the performance of the proposed detection-aligned localization algorithm. The evaluation was done using real multichannel impulsive source recordings.

5.4.1 Description of the database

The evaluation of source localization algorithms was done using the same database described on section 4.5.1. The database is composed of 12 channel recordings of 160 impulsive acoustic events generated at 16 unique locations (10 events per location). Microphone locations were selected by following three simple rules, which ordered by priority are: first, there must be direct line of sight between microphones; second, the array must cover as large an area as possible given the available equipment (i.e., finite amount of cables); and third, the array must mimic a random geometry. Source locations were selected following just one general condition: sources must evenly cover as large an area as possible given the recording area topology.

In order to obtain the 2D location of each element, a full set of microphone pairwise distances and multiple source-microphone distances were measured with a *BOSCH* GLM 250 VF laser distance meter, capable of measuring distances from 0.05 m to 250 m with a typical measuring accuracy of ±1 mm. Microphone locations were estimated with MDS using Matlab's *mdscale* function, which took the 12×12 dissimilarity matrix (i.e., pairwise microphone distances), and returned an estimate of 12 microphone locations in 2 dimensions. Microphone locations were then transposed so that the location of the first microphone is at the center of coordinates. Source locations were instead estimated one by one using the algorithm described in section 5.6 assuming gaussian noise. The algorithm was modified to work with distances rather than time but the main methodology is exactly alike. It is important to mention that there are missing distance measures due to trees and other obstacles blocking the line of sight from some source locations to various microphones. There is a minimum of 5 distance measures for any of the source locations. Figure 5.6 shows the numbered locations of the microphones (circles) and sources (crosses) superimposed on a schematic map of the recording area. Grey areas represent grass while white areas are roads and asphalt. Please note that only large obstacles are represented (i.e., trees as circles and buildings as rectangles).

As mentioned in the previous chapter, recording took place in two sessions in which 5 impulsive noises were generated with different sources at each of the 16 locations. An air temperature of approximately 23° C was measured during the break between sessions, which translates into an speed of sound $c \approx 345$m/s.

The maximum distance between microphones is 16.2m, thus the minimum signal length required to obtain all the possible TDoA values by cross-correlating the signals is $L = 4142$ samples. It is also important to mention that the database was recorded with a wired microphone array so the 12 channels are perfectly synchronized.

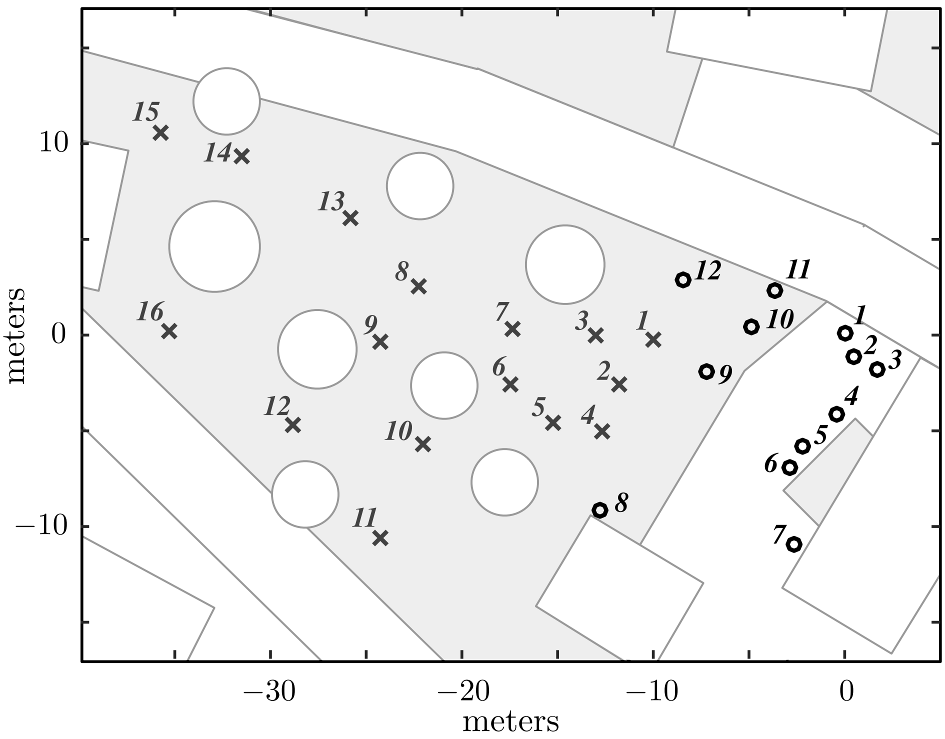


Figure 5.6: Locations of the microphones (circles) and sources (crosses) superimposed on a schematic map of the recording area.

5.4.2 Description of the Experiments

In order to evaluate the performance of the proposed approach to sound source localization, two SRP-based methods were tested using the multichannel database of real impulsive sounds described in the previous section. The tested methods are:

- SRP localization in conjunction with the standard detection-based approach ($L > 2\tau_{d^+}$, channels sharing a common timebase).
- SRP localization in conjunction with the proposed detection-aligned cross-correlations ($L_s < L$, each signal aligned to a local detection).

Both methods were tested using multiple array sizes (M), from 3 to 12 microphones. Results for each array size were computed using up to 100 random combinations of M microphones per impulsive event in the database (for $M = \{10, 11, 12\}$ the number of combinations is $66, 12$ and 1 respectively). Microphone combinations are the same in every experiment regardless of the localization method or other parameters. The database only contains 16 distinct source locations, but since every evaluation is independent for any given event, the results for each array size and localization method were obtained by averaging the localization error of every event and microphone combination.

The results of the detection-aligned method were obtained using the detection system proposed in the previous chapter (i.e K-LDA with $K = 6$ and 9-frame compressed temporal context). It is important to mention that the selected detection system does not achieve perfect detection, hence, only 156 out of 160 events are considered for the results. Detection onset estimation (Γ_m) was computed as described in section 5.3.1 using a time window of 65 frames (T_Γ), centered around the global detection frame index obtained by the selected detection system with $M = 12$.

The results of the the standard detection-based approach were obtained assuming perfect detection, thus, onset labels were used. The onset labels were quantized according to the frame hop length used in the experiments of the detection system described in chapter 4. The quantized labels were obtained dividing the labels (time instant in samples) by the hop length (256 frames) and then rounding the result to the closest integer. This way, the quantized labels are used as an approximation to the frame index obtained by a perfect local detector for the detection-aligned cross-correlations. Global time labels were obtained as the average of the local time labels of those channels under evaluation.

Both methods were tested with various signal lengths. For every case, source locations were obtained using SRP with a grid resolution of 0.1m, and cross-correlations were obtained using the absolute value of the audio signals as discussed in section 5.2.1.

5.4.3 Experimental Results

Table 5.1 shows the mean localization error (and standard deviation) of SRP-based localization with the proposed detection-aligned scheme and standard signal alignment for various cross-correlation lengths and array sizes. Note that the selected signal lengths keep a relation $\Xi = 8$ in pairs (e.g., $8 \times 1024 = 8192$). From the table it is clear that both methods are capable of of obtaining reasonable accuracies when the cross-correlations are of sufficient length. The proposed method is able to match, and even surpass, the accuracy of the standard approach while requiring significantly less resources (computational power and bandwidth). Overall, the best of the tested SRP-based localization schemes is the one based on the proposed detection-aligned signal segments with a length 1024 samples (23.2 ms at f_s=44100Hz), although the differences with some of the other tested configurations are not relevant enough. Either method fails to maintain accuracy with the shorter signal length tested ($L_s = 256$ and $L = 2048$). The drop in performance is explained by the misalignment of the signals produced by the coarse resolution of the detection time stamps, that, in some cases 'pushes' the onset of the events outside of the selected signal segment. It is important to mention that the standard deviation of the localization error is in every case, at least of the same order as the mean localization error. Large variances are caused by 'failed' localizations. Each of the presented results was computed averaging the localization error of 156 individual acoustic events and multiple random combinations of microphones for each event. Therefore, the estimated location of a given events also depend on the microphone combination. Figure 5.7 show the scatterplot of the mean localization error of all the random microphone combinations for each event as a function of the source distance to the origin of coordinates. The figure shows that the localization error is related to both the source's location and the emitted signal (specially at larger distances).

Regarding the precision of the detection onset estimation, Figure 5.8 shows the histogram of the estimation error obtained from the onset labels and the estimated detection time stamps (Γ_m). Note that the former have a resolution of 1 sample while the latter have a resolution of 0.5 frames (256 samples). The standard deviation of the estimation error is 111.5, 98% of the estimates have an

Table 5.1: Mean error of SRP-based localization with the proposed method and standard signal alignment for various cross-correlation lengths and array sizes (M).

	Detection-aligned			Standard		
M	$L_s = 256$	$L_s = 512$	$L_s = 1024$	$L = 2048$	$L = 4096$	$L = 8192$
3	10.7m ±8.9	8.1m ±8.0	**7.7**m ±7.8	11.0m ±9.1	8.0m ±8.0	7.8m ±7.9
4	7.9m ±7.7	5.3m ±6.0	**5.0**m ±5.7	9.0m ±8.5	5.3m ±6.0	5.1m ±5.8
5	6.6m ±6.7	4.0m ±4.6	**3.8**m ±4.3	8.1m ±8.1	4.0m ±4.6	**3.8**m ±4.4
6	5.7m ±6.1	3.4m ±3.9	**3.1**m ±3.6	7.5m ±8.0	3.3m ±3.8	3.2m ±3.6
7	5.0m ±5.5	2.8m ±3.3	**2.6**m ±2.9	6.8m ±7.7	2.8m ±3.2	2.7m ±3.0
8	4.7m ±5.2	2.6m ±3.1	**2.5**m ±2.8	6.4m ±7.5	2.7m ±3.0	**2.5**m ±2.9
9	4.2m ±4.8	2.4m ±2.8	**2.2**m ±2.5	6.1m ±7.3	2.4m ±2.6	**2.2**m ±2.5
10	3.8m ±4.4	2.1m ±2.5	**2.0**m ±2.2	5.8m ±7.1	2.2m ±2.4	2.1m ±2.3
11	3.5m ±4.1	2.0m ±2.3	**1.8**m ±2.0	5.4m ±6.9	2.0m ±2.2	1.9m ±2.1
12	3.3m ±3.9	**1.7**m ±2.0	**1.7**m ±1.9	5.4m ±7.0	1.8m ±1.9	1.8m ±2.1

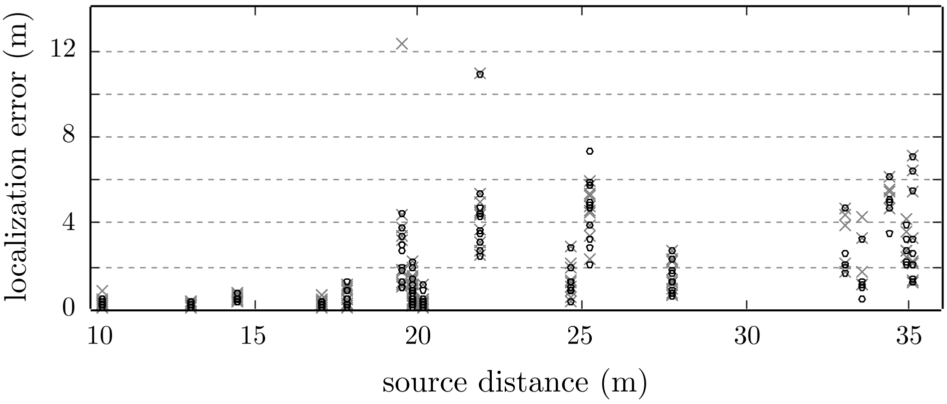


Figure 5.7: Scatterplot of the localization error as a function of the source distance. Circles: Detection-aligned ($L_s = 1024$), Crosses: Standard ($L = 8192$).

accuracy of ± 1 frame, thus, it can be said that the estimation is performing more than adequately. Interestingly, almost 38% of the errors come from onset estimates preceding the onset label, which may be an indication of human error in the onset labels.

Given the accuracy of the onset estimation, differences in performance between the two methods are limited mostly to the quality cross-correlations of the selected audio signal segments. On section 5.2.1 we claimed that the main source of error is coherence loss between channels due to the large inter-microphone distances involved in TDoA estimation. This claim can be easily justified by showing an example of some impulsive acoustic event recorded at various microphones. Figure 5.9 shows the signals recorded by 6 of the 12 microphones included in the database for a 'balloon pop' generated at two distant locations. The figure represents the signal segments selected to compute the cross-correlations using the two tested methods for either source location. It is interesting to see the large differences in the received signals even when the microphones are relatively close to each other (m_1, m_2 and m_3), and consequently, estimating the TDoA of some impulsive event using standard cross-correlation is prone to large errors. Figure 5.10 shows the cross-correlation

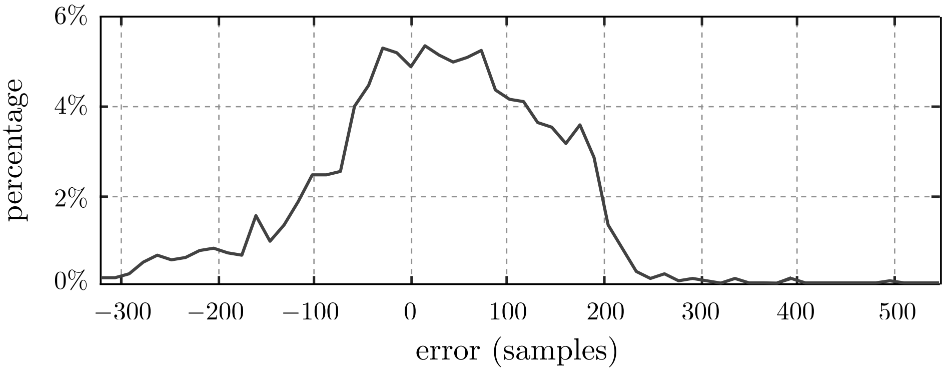


Figure 5.8: Histogram of the detection onset estimation error.

of some of the signals represented in Figure 5.9 (location 1, detection-aligned) using three different cross-correlation methods, namely: standard cross-correlation, PHAT-weighted cross-correlation, and the used absolute value cross-correlation. Figure 5.11 shows the SRP surface obtained with the signals from Figure 5.9 (location 15, detection-aligned) using two cross-correlation methods, where is possible to see that the location estimate using standard cross-correlations is on the edge of the search space and the SRP surface is quite noisy. Note that the localization results obtained with standard cross-correlations and GCC-PHAT are fairly worse than the presented ones, therefore they were dismissed.

In conclusion, the proposed local detection-aligned cross-correlation method is capable of achieving the same, or even better results than the standard detection-based approach, while significantly reducing the computational complexity. Note that reducing the length of the signals being cross-correlated not only alleviates computational load but also reduces the bandwidth requirements for the implementation on the algorithm.

It is important to mention, that the adequate performance of the proposed localization method is due to the high SNR of the recorded impulsive events. If the target events where to be masked by undesired noise or in the presence of additional strong sources, SRP-based source localization estimates would become much less reliable.

5.5 ToA-based Localization

Typically, acoustic source localization is tackled using cross-correlations between the signals recorded by different microphones because there is no knowledge about the signal emitted by the source. We refer to this approach as TDoA-based source localization, however, in certain situations, it is also possible to use ToA-based source localization.

When the localization process is preceded by a multichannel sound event detection system, it could be said that, while the exact signal emitted by the target source is still largely unknown, there is certain knowledge about its ToA. In chapter 3 we were able to obtain the ToA of the n^{th} source at the m^{th} microphone with sample level precision because we knew the signal that was being emitted. In the current situation, we can obtain a rough estimation of the local ToA using the detection onset estimation (Γ_m) described in the previous section. Although, the resolution of Γ_m is too coarse to

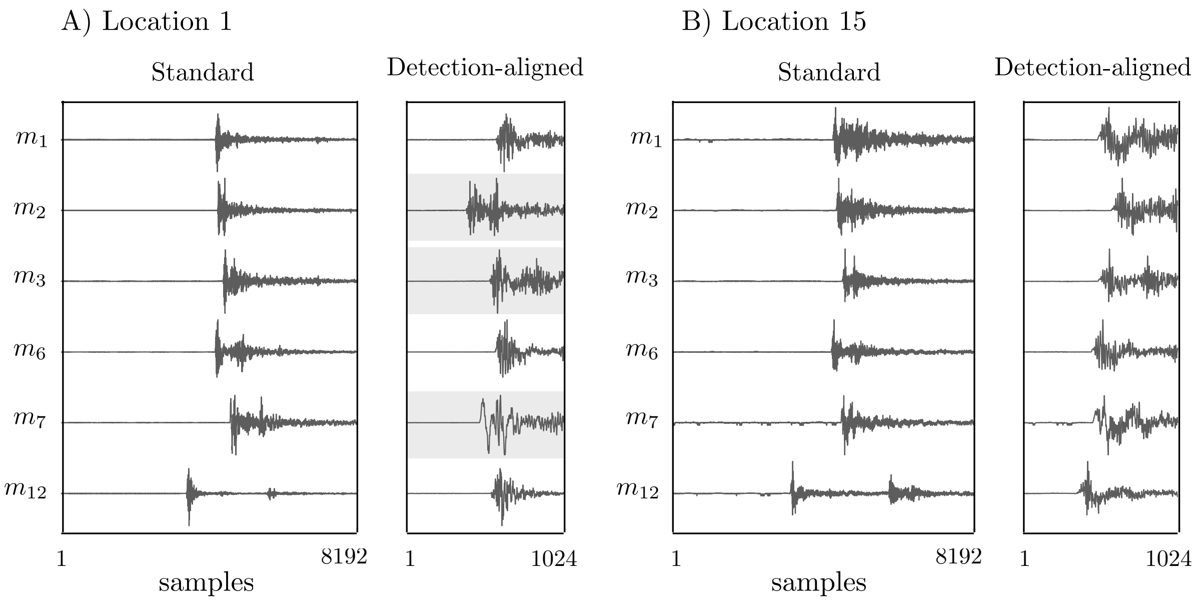


Figure 5.9: Example of signals to be cross-correlated (normalized amplitude) using two alignment approaches. Recordings of a 'balloon pop' generated at two locations.

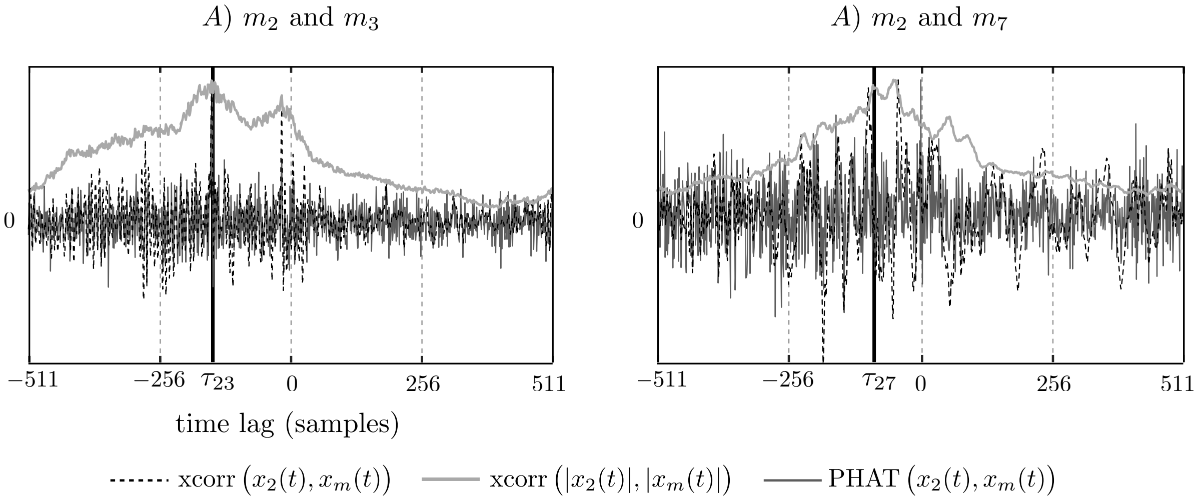


Figure 5.10: Cross-correlation of some signals from Figure 5.9 (location 1, detection-aligned) using different methods. Actual TDoA represented as a vertical black line.

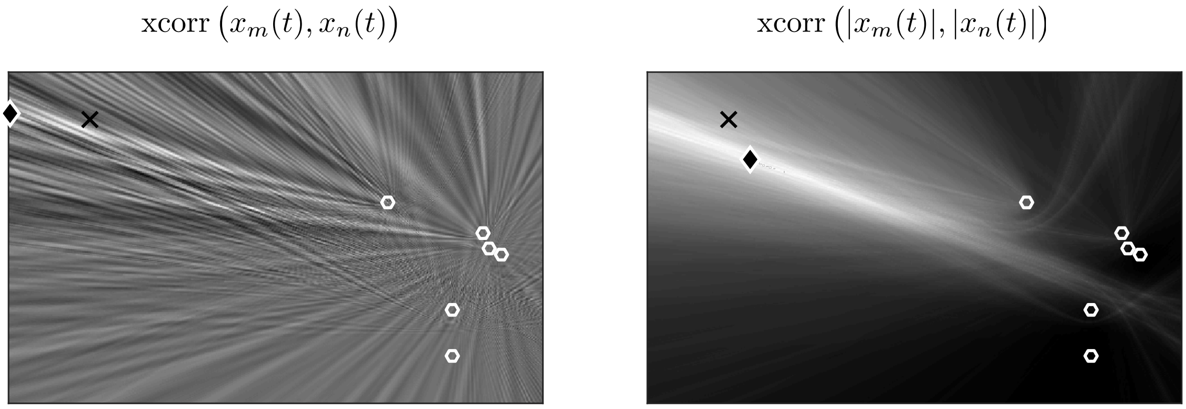


Figure 5.11: SRP surface of a 'balloon pop' in location 15 using two cross-correlation methods. Circles are microphones, crosses are the source location and diamonds the estimated locations.

produce a good source location estimation due to the wide time period between consecutive frames. However, it is possible to entirely bypass the need for cross-correlations by estimating the local onsets of a target impulsive event in the time-domain.

5.5.1 Time-domain Onset Estimation

In order to perform acoustic source localization without the need for cross-correlations, we propose to use a double onset estimation scheme, intended to obtain the ToA of a target acoustic event at every channel. Local ToA estimation begins with a global event detection followed by a local detection onset estimation as described in section 5.3.1. Once Γ_m has been obtained, it is used to select an audio signal segment over which time-domain onset detection is used to estimate the local ToA.

Detecting the onset of an acoustic signal in the time domain is harder than doing so with the probabilistic output of a detector. Onset detection of acoustic events is most often performed in the time-frequency domain, where different existing methods exploit various sources of information such as changes is spectral patterns or differences in phase between frames. In the presence of multiple sound sources, it is generally not possible to detect onsets directly on the time-domain, that is, unless the target events present an impulse-like onset [BDA+05] as it is the case. Such acoustic events are characterized by large energies and sudden onsets that are reasonably simple to detect just by looking at the energy evolution of the received signal. Thus, the first step to estimate the ToA of an event is to obtain a representation of the input signal in terms of received energy.

It is common knowledge that impulsive audio events have a wide spectral content, and that most undesired noise is going to be concentrated in the lower part of the spectrum because most sources do not produce high frequency sounds and even if they do, the higher the frequency the faster its dissipation in air. Therefore, it makes sense to filter out the low frequency components since it is very likely that most high frequency energy is coming from the target event. Let $x_m(t)$ be the signal received at the m^{th} microphone, and $h_1(t)$ be the coefficients of a high-pass Finite Impulse Response (FIR) filter of length L_{F1}. We can then define the high-pass filtered signal as $x'_m(t) = x_m(t) * h_1(t)$. The next step is to obtain the relative energy level of the signal. This can easily be done by taking the square of every sample and applying a moving average filter ($h_2(t)$ of length L_{F2}) in order to approximate the energy envelope of the signal. Finally, the logarithm of the output is computed, since energy is best evaluated on a logarithmic scale. This way, the approximate energy evolution of the input signals can be obtained with:

$$q_m(t) = \log\Big(\big(x_m(t) * h_1(t)\big)^2 * h_2(t)\Big). \tag{5.12}$$

Note that filtering can be performed either on the time-domain or the time-frequency domain, and more importantly, as long as every channel uses the same filters, there is no need for correcting group delay because it is equal in every channel. Ideally, in the presence of an impulsive event and over a long enough period of time, $q_m(t)$ should resemble a shifted exponential decay function so that the onset of the event can be detected by different means, such as curve fitting. However, in outdoor environments the appearance of strong reflections will likely result in an irregular decay as shown by Figure 5.12.

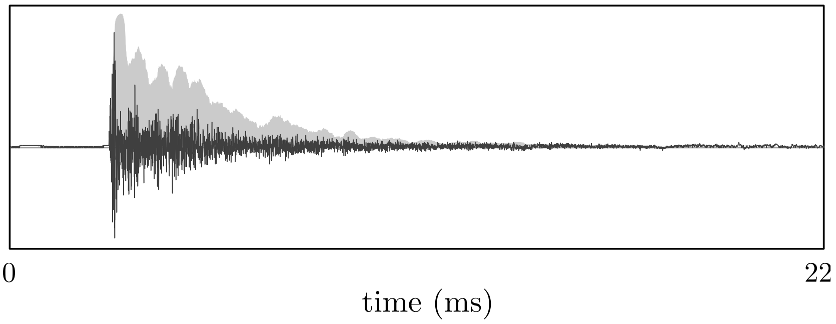


Figure 5.12: Impulsive audio event and its logarithmic energy envelope (light gray area).

Obtaining the energy envelope involves certain processing that can be quite demanding, likewise, the estimation of the event's ToA gets more complex the longer the signal is (i.e., larger search space). In order to produce the estimation as efficiently as possible, we propose to use the local detection onset estimates to select an input signal segment, so that assuming perfect detection, it can be affirmed that τ_m is in the environment of Γ_m. The methodology is equivalent to that of the detection onset estimation, moreover, by selecting a suitable segment length the onset can be estimated again with a shifted step function.

This way, a signal segment of length T_Λ samples is selected around Γ_m, such that it contains enough of the signal for the onset to be contained within it, taking into consideration the coarse resolution of Γ_m. Each segment is then pre-processed as described by (5.12) to obtain its energy envelope ($q_m(t)$). It is important to mention that before computing the logarithm, the signal can be decimated by an integer factor (Ξ) in order to further increase the efficiency of the algorithm. Note that a shorter signal implies a smaller number of logarithm computations and a smaller search space for the curve fitting. Decimation can easily be performed by substituting the moving average filter for a Cascaded Integrator-Comb (CIC) filter[2]. Once the signals has been pre-processed, instead of using the Heaviside step function for the curve fitting, we propose to use the logistic function to better model the finite rise time of real acoustic events. The logistic function is a common sigmoid curve given by the following expression:

$$\mathcal{F}(t-\tau) = \frac{1}{1+e^{\left(-a(t-\tau)\right)}}. \tag{5.13}$$

where τ is the the sigmoid's midpoint and a is the steepness of the curve. Note that parameter a does not need to be estimated, it can be fixed *a priori* to adjust to the expected rise time of the target events.

Using a LS approach, the m^{th} ToA is estimated by finding the value of τ in search space Θ that minimizes:

$$\tau_m = \arg\min_{\Theta}\left(S\right), \text{ with: } S = \sum_{t=1}^{T_\Lambda}\left(\mathcal{F}(t-\tau) - q_m(t)\right)^2, \tag{5.14}$$

where $\Theta = \{\tau \in \mathbb{R} : t + \Gamma_m - T_{\Lambda 1} < \tau < t + \Gamma_m + T_{\Lambda 2}\}$, and $T_{\Lambda 1} + T_{\Lambda 2} = T_\Lambda$. In contrast to the method described in section 5.3.1, this time around, the minimization is best solved using an iterative approach (i.e., gradient descent) given the increased complexity of each objective function

[2] A CIC filter can be seen an efficient implementation of a moving-average filter followed by a decimation stage that is obtained by rearranging the operations needed to perform both operations.

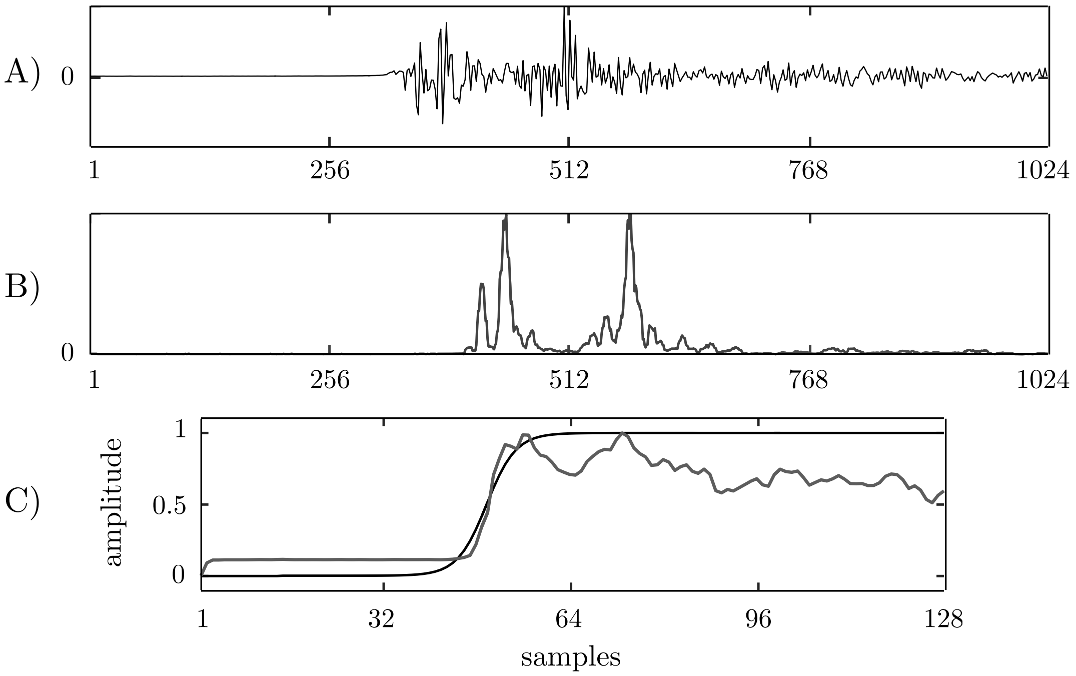


Figure 5.13: The 3 stages of time-domain onset estimation. A) raw audio signal segment, B) energy envelope approximation, C) decimated ($\Xi = 8$) log-energy and fitted logistic function.

evaluation. Note that this has the added benefit of making it possible to obtain a non integer value of τ as the solution, thus, in case $q_m(t)$ was decimated, the resolution of τ_m is not proportional to the decimation factor (i.e., resolution $>$ Ξ samples). It is also important to mention that for the curve fitting to work, either $q_m(t)$ has to be normalized (between 0 and 1) or the logistic function has to be transposed and scaled to match the span of $q_m(t)$.

Figure 5.13 shows an example of time-domain onset estimation by representing the employed signals ($f_s = 44100$Hz) at three stages. From top to bottom, A: raw audio signal, B: energy envelope and C: decimated ($\Xi = 8$) log-energy and fitted logistic function. From the figure it is possible to see that by selecting a proper T_Λ (approximately 2.3ms), exponential decay can be ignored and the log-energy can be approximated by a step function.

Onset estimation in the presence of multiple sources (i.e., low SNR) is a difficult task, which makes onset detection a very active area of research, specially in those sub-fields related to music processing (e.g., finding the transients of the notes played by one of various instruments in a recording). The proposed approach can be regarded as the simplest version of onset detection (i.e., energy-based detection). It is important to mention again that this approach can only work with sound events that have a fast transient, and only under high SNR conditions. Fortunately enough, impulsive events are likely to meet both conditions due to their high sound pressure and wide spectral content.

5.6 Maximum Likelihood Estimation

Most classic acoustic source localization algorithms are based on TDoA estimates. Therefore, in order to perform ToA-based source localization, a non-classical algorithm has to be used. We propose to estimate the location of the source of some target acoustic event, from the known locations of M microphones and M ToA estimates, using a ML estimator. The proposed method is derived from the general self-localization problem formulation in which every source and microphone are independent to one another and there is no knowledge of their positions or timings. For this particular application, it is possible to ignore most of the unknowns considered in the original formulation (i.e., microphone locations and clock offsets), so that the problem becomes finding the location and the emission time of one source.

Let us consider an acoustic source inside an euclidean space located at the point defined by a vector $\mathbf{p}$, and let us also consider M receivers (i.e., nodes equipped with one microphone), located at the points defined by $\mathbf{p}_m$, $m = 1, \ldots, M$. At this point we are going to assume a planar space, but the methodology can be adapted to work in 3D space. Concerning timing, we consider t as the emission time of the source and Δ_m as the clock offset of the m^{th} node. This way, the ToA at the m^{th} microphone is composed of three unknowns, and can be expressed as:

$$\tau_m = \delta_m + t + \Delta_m, \text{ with: } \delta_m = \frac{||\mathbf{p} - \mathbf{p}_m||}{c}, \tag{5.15}$$

where δ_m is the ToF between the source and the m^{th} microphone, and c is the speed of sound.

If we consider that the microphone locations are known, and that the nodes are synchronized ($\Delta_m = 0$), the objective then becomes to find the most likely source location and emission time ($\mathbf{p}$ and t) that explain a set of ToA estimates. We propose two solutions to the ML estimator, assuming that the distribution of the observed ToAs is governed either by a Gaussian PDF or a Laplacian PDF.

5.6.1 Gaussian Distribution

Let us consider that a full set of M ToA estimates of some acoustic event is available. In a first approach we are going to assume that the error or said estimates can be modeled as a gaussian distribution with zero mean and variance σ^2, so that the PDF of a given estimate can be described with the next equation:

$$\mathcal{F}(\tau_m; \mathbf{p}, t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{1}{2\sigma^2}(\tau_m - \delta_m - t - \Delta_m)^2\right)} \tag{5.16}$$

The objective is to estimate the most likely source location and emission time, which can be done from the likelihood function associated to a set of ToA observations, given by:

$$\mathcal{L} = \prod_{m=1}^{M} \mathcal{F}(\tau_m; \mathbf{p}, t). \tag{5.17}$$

The solution is then obtained by finding the values of the variables that maximize $\mathcal{L}$, or more conveniently the log-likelihood. The log-likelihood function associated to all the observations obtained

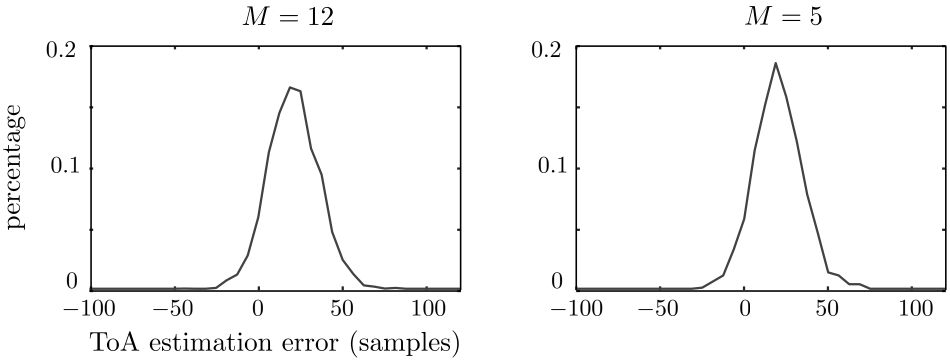


Figure 5.14: Histogram of the ToA estimation error for two array sizes using the database described on section 5.4.1.

from some source can be determined using:

$$\log(\mathcal{L}) = \sum_{m=1}^{M} -\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}(\tau_m - \delta_m - t - \Delta_m)^2 \tag{5.18}$$

Since we are interested in the relative value of $\log(\mathcal{L})$, its maximization can be performed just by evaluating those terms in (5.18) that depend on the target variables. Thus, to maximize $\log(\mathcal{L})$ is equivalent to minimize the next expression:

$$E = \sum_{m=1}^{M} (y_m - \delta_m - t - \Delta_m)^2 \tag{5.19}$$

At this point we can introduce the assumption of synchronization ($\Delta_m = 0$) and rearrange the objective function so that the solution to the problem can be found by solving an equivalent LS problem given by:

$$\{\mathbf{p}_s, t_s\} = \arg\min_{\{\mathbf{p},t\}} \Big(\sum_{m=1}^{M} \Big((||\mathbf{p} - \mathbf{p}_m||/c + t) - y_m \Big)^2 \Big), \tag{5.20}$$

where $\mathbf{p}_s$ and t_s are the estimated source location and emission time respectively.

The proposed estimation is a non-linear least squares problem, therefore solving it iteratively is the easiest option. Nevertheless, the number of iterations can be reduced by using a second-order iterative algorithm such as Levenberg-Marquardt. Defining the error $e_m = \tau_m - ||\mathbf{p} - \mathbf{p}_m||/c - t$ the Jacobian can be easily computed as:

$$\frac{\partial e_m}{\partial t} = -1, \text{ and: } \frac{\partial e_m}{\partial \mathbf{p}_{mx}} = -\frac{\mathbf{p}_x - \mathbf{p}_{mx}}{c\delta_m}, \tag{5.21}$$

where subindex x represents the coordinate axis.

5.6.2 Laplace Distribution

The real distribution of the ToA estimation error does not necessarily follow a Gaussian distribution, specially when a small number of nodes is used to locate a source. Figure 5.14 shows the histogram of the ToA estimation error for two array sizes in order to illustrate this claim.

Taking this into consideration, as a second approach, we are going to assume that the Laplace distribution better describes the error of the estimates, so that the the PDF of a given estimation can be expressed as:

$$\mathcal{F}(\tau_m; \mathbf{p}, t) = \frac{1}{2b} e^{\left(-\frac{1}{b}|\tau_m - \delta_m - t - \Delta_m|\right)}, \tag{5.22}$$

where operator $|.|$ is the L^1-norm and $b > 0$ is the scale parameter of the distribution.

In this case, the log-likelihood function associated to a set of ToA observations can be determined using the following equation:

$$\log(\mathcal{L}) = \sum_{m=1}^{M} -\log(2b) - \frac{1}{b}\left|\tau_m - \delta_m - t - \Delta_m\right|. \tag{5.23}$$

Again getting rid of the terms that do not depend on the target variables, to maximize the log-likelihood is equivalent to minimize a simpler function, in this case a L^1-norm error given by:

$$\tilde{E} = \sum_{m=1}^{M} |\tau_m - \delta_m - t - \Delta_m| = \sum_{m=1}^{N} \tilde{e}_m^2, \tag{5.24}$$

where, by assuming synchronization:

$$\tilde{e}_m = \sqrt{\left|\left(\frac{||\mathbf{p} - \mathbf{p}_m||}{c} + t\right) - \tau_m\right|} = \sqrt{|e_m|}, \tag{5.25}$$

This time the problem can not be directly solved as a LS problem, although, note that expression 5.25 relates the error obtained for the Laplacian distribution ($\tilde{e}_m$) with that obtained for the gaussian distribution (e_m). Thus, the estimation can still be solved as a non-linear LS with some minor modifications. Moreover, computation of the Jacobian is still rather simple, it can be determined with:

$$\frac{\partial \tilde{e}_m}{\partial \mathbf{p}_{mx}} = -\frac{\operatorname{sgn}(e_m)}{2\tilde{e}_m}\frac{(\mathbf{p}_x - \mathbf{p}_{mx})}{c\delta_m}, \tag{5.26}$$

$$\frac{\partial \tilde{e}_m}{\partial t} = -\frac{\operatorname{sgn}(e_m)}{2\tilde{e}_m}, \tag{5.27}$$

where $\operatorname{sgn}()$ is the signum function.

5.7 Experimental Work and Results: ToA-based localization

This section describes the experiments conducted to evaluate the performance of the proposed ToA-based sound source localization algorithm.

5.7.1 Description of the Experiments

The ToA-based localization method was tested using the database described in section 5.4.1. Additionally, the configuration of the experiments is the same as that described in section 5.4.2 for the SRP-based methods, that is, an array size (M) from 3 to 12 microphones and up to 100 random combinations of M microphones per impulsive event. Detection onset estimation (Γ_m) is also the shared with section 5.4.2: the outputs of the K-LDA detector are processed as described in section 5.3.1 using a time window of 65 frames. Please note that only 156 out of 160 events are considered for the results of ToA-based localization.

ToA estimation was computed as described in section 5.5.1 using audio signal segments of length 1024 samples (T_Λ) each of them centered around their respective Γ_m. Raw audio segments were pre-processed first with a 64th order high-pass FIR filter (2000 Hz cut-off frequency) an then with a 8th order moving average filter. Regarding time-domain onset detection, curve fitting was performed with 3 decreasingly complex variants, namely: Logistic function fitting on the full length signal segments; Logistic function fitting on signal segments decimated by a factor of 8 ($\Xi = 8$); and unitary step sitting (the same used for detection onset estimation) on signal segments decimated by a factor of 8 ($\Xi = 8$). In both cases, logistic function fitting was done with Matlab's curve fitting tool.

Source locations were obtained using the ML estimator proposed on section 5.6 with both types of probability distribution. The minimization was solved using Matlab's implementation of the Levenberg-Marquardt algorithm. In order to avoid local minima, we consider that source locations are estimated by every node using a different starting point, thus, the number of evaluations is proportional to M. It is worthwhile mentioning that, convergence of the proposed ML estimator depends on the assumed PDF. Location estimates assuming a Gaussian distribution were obtained as the best among M evaluations (in terms of the objective function), while estimates assuming a Laplace PDF showed some convergence problems and were evaluated twice per node ($2M$ evaluations).

Finally, the effect of clock offset in the localization results is tested with the proposed methods (detection-aligned SRP and ToA-based localization). Note that the database recordings are properly synchronized, thus, clock offset was simulated by offsetting the audio signals a random number of samples. Clock offset was modeled as a normal distribution with multiple increasing variances (σ_Δ^2) in order to evaluate the influence of synchronization on the precision of the localization.

5.7.2 Experimental Results

Tables 5.2 and 5.3 show the mean localization error (and standard deviation) for various array sizes of the proposed ToA-based localization assuming Gaussian an Laplace PDFs respectively, using different sets of ToA estimates. The results show that for any of the tested ToA estimation methods, the localization error is lower than those obtained with the SRP-based approaches. The best results are obtained assuming a Gaussian PDF and estimating the ToAs by fitting a Logistic function as described on section 5.5.1, expect for $M \leq 4$ where the Laplace PDF with the onset labels outperforms it. Note that the results of the decimated version of the Logistic function ($\Xi = 8$, $T_\Lambda = 128$) are practically undistinguishable from those obtained using the whole energy envelopes ($T_\Lambda = 1024$), while being fairly more efficient in terms of computational load per ToA estimate. Also, the results obtained with the simplest ToA estimation method, that being the unitary step fitting ($\Xi = 8$, $T_\Lambda = 128$) are not far behind. Moreover, estimating the time-domain onsets with the unitary step approach produces better results that any of the tested SRP-based methods. Special mention must be made about the results obtained when the onset time labels are feed to the localization algorithm.

Table 5.2: Mean localization error of the proposed ML estimator assuming Gaussian noise for various array sizes (M) and different sets of ToA estimates.

M	**Onset Labels**	**Logistic**	**Logistic** (Ξ=8)	**Unitary Step** (Ξ=8)	**Best SRP***
3	6.3m ±15.8	6.1m ±15.3	5.9m ±14.7	6.3m ±15.2	7.7m
4	3.1m ±8.7	3.0m ±8.0	3.0m ±8.3	3.3m ±8.8	5.0m
5	2.1m ±5.6	**2.0**m ±5.0	2.1m ±5.9	2.3m ±6.1	3.8m
6	1.6m ±1.9	**1.5**m ±2.1	**1.5**m ±2.2	1.7m ±2.5	3.1m
7	**1.4**m ±1.9	**1.4**m ±2.3	**1.4**m ±2.6	1.6m ±2.5	2.6m
8	**1.3**m ±1.6	**1.3**m ±2.1	**1.3**m ±2.2	1.5m ±2.5	2.5m
9	1.2m ±1.4	**1.1**m ±1.8	**1.1**m ±1.8	1.4m ±2.3	2.2m
10	**1.1**m ±1.4	**1.1**m ±1.7	**1.1**m ±1.8	1.3m ±2.3	2.0m
11	1.1m ±1.3	**1.0**m ±1.4	1.1m ±1.5	1.3m ±2.1	1.8m
12	**1.0**m ±1.2	**1.0**m ±1.3	**1.0**m ±1.4	1.3m ±2.1	1.7m

*Best results obtained by the SRP-based methods.

For the most part, the results are equal (or worse) to those obtained with the Logistic Function. This points out at manual labeling of the event's onset times being subjected to an error comparable to that obtained with the automated estimation. Although, the standard deviation of the localization error is consistently lower for the onset labels.

Overall, source localization using the proposed ToA-based approach seems like the superior method. This has two main implications. On the one hand, the computational complexity of obtaining the ToA estimates is lower than that of computing the GCC needed for the SRP, specially if we take into consideration that for any network size there are M ToA estimates compared to $M(M-1)/2$ cross-correlations. On the other hand, the minimization, which commonly is the most costly procedure of any localization algorithm, has the potential of being simplified by taking some additional assumptions.

The current implementation is using the proposed ML estimator to find the solution iteratively, however there are a number of closed form algorithms that could be used instead [GS08, APH14] as a tradeoff between accuracy and computational complexity. Note that such algorithms typically work with TDoA estimates, which can easily be obtained from the estimated ToAs, and while it is also possible to use a closed-form solution with TDoAs obtained from cross-correlations, at least for the signals in the present database, the obtained TDoA estimates are not precise enough. In relation to this, it is worthwhile mentioning that we tested the ML estimator with ToAs estimates obtained from the GCC without success.

While on the matter of the proposed ML estimator, from tables 5.2 and 5.3 it seems like assuming a Laplace PDF produces worse results than assuming a Gaussian PDF for almost every case. However, taking a closer look at the results, the problem lies in the convergence rate of both methods. Figure 5.15 shows the histogram of the localization error of all the experiments for two array sizes ($M=5$ and $M=10$) using the proposed ML estimator with Gaussian PDF (dashed line) and Laplace PDF (solid line), where it is possible to see that the number of individual experiments with the smaller possible error is higher for the Laplace PDF. Overall, the Laplace version of the algorithm obtains

Table 5.3: Mean localization error of the proposed ML estimator assuming Laplacian noise for various array sizes (M) and different sets of ToA estimates.

M	**Onset Labels**	**Logistic**	**Logistic** (Ξ=8)	**Unitary Step** (Ξ=8)	**Best SRP***
3	**4.6**m ±9.6	5.1m ±11.6	4.9m ±10.9	5.4m ±12.1	7.7m
4	**2.7**m ±4.2	2.8m ±5.4	**2.7**m ±5.2	3.1m ±6.3	5.0m
5	**2.0**m ±2.5	**2.0**m ±3.8	**2.0**m ±3.8	2.2m ±4.2	3.8m
6	1.6m ±2.0	1.7m ±3.2	1.7m ±3.2	1.9m ±3.5	3.1m
7	1.5m ±1.9	1.6m ±2.7	1.6m ±2.6	1.8m ±3.1	2.6m
8	1.4m ±1.8	1.4m ±2.1	1.4m ±2.1	1.6m ±2.5	2.5m
9	1.3m ±1.7	1.3m ±1.7	1.3m ±1.7	1.5m ±2.1	2.2m
10	1.3m ±1.6	1.2m ±1.6	1.3m ±1.6	1.4m ±2.1	2.0m
11	1.2m ±1.6	1.3m ±1.6	1.3m ±1.7	1.4m ±2.1	1.8m
12	1.1m ±1.5	1.2m ±1.6	1.2m ±1.6	1.3m ±1.9	1.7m

*Best results obtained by the SRP-based methods.

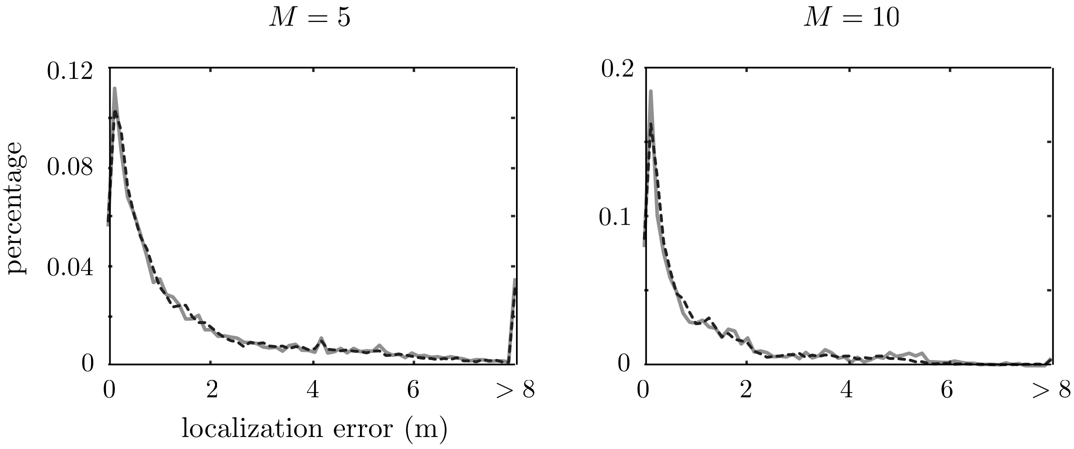


Figure 5.15: Histogram of the localization error of all the experiments for two array sizes using the proposed ML estimator with Gaussian PDF (dashed line) and Laplace PDF (solid line).

noisier results with a larger proportion of large localization errors, which points out as convergence being a greater problem for that version of the ML estimator.

5.7.3 Effect of the Synchronization on the Localization

Up until now, the presented results were obtained assuming synchronized nodes. The database was recorded with a multichannel sound interface, therefore every channel shares a common time-base. In order to test the influence of synchronization errors in the localization results we introduced random clock offsets to every channel, modeled as a normal distribution with various standard deviations (σ_Δ).

Figure 5.16 shows the the mean localization error for various array sizes of the proposed ToA-based localization (Gaussian PDF) and detection-aligned SRP ($L_s = 1024$) when presented with

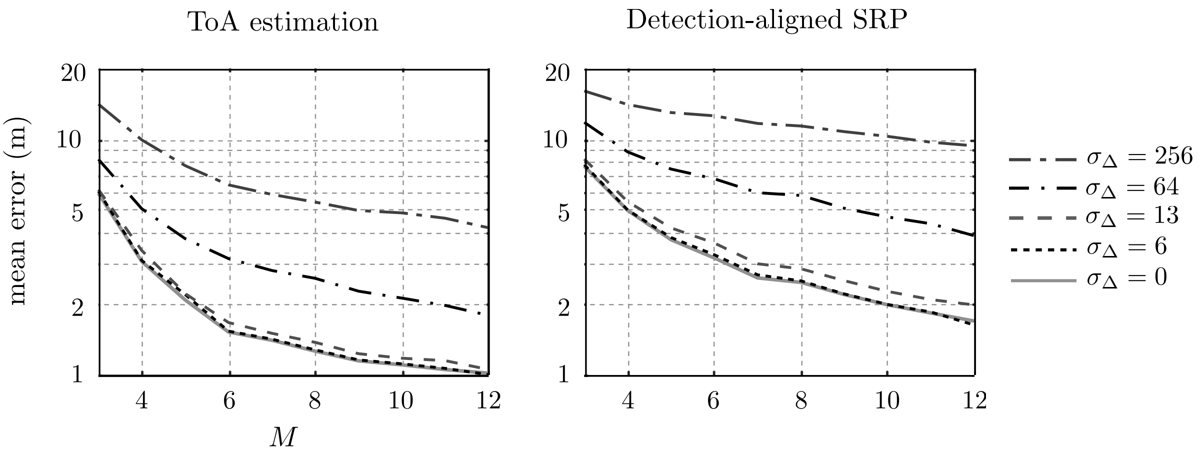


Figure 5.16: Effect of clock offset in source localization with the two proposed methods. Clock offset modeled as a normal distribution with standard deviation σ_Δ (samples).

synchronization uncertainty. The shown results were obtained by introducing 4 increasingly larger random clock offsets to the raw audio signals governed by σ_Δ. The tested values of σ_Δ are: 6 samples (the smallest clock offset error obtained in chapter 3), 13 samples, 64 samples and 256 samples. The results show that synchronization uncertainty definitely has a significant impact on the localization error, specially for larger values of σ_Δ. The localization error with $\sigma_\Delta = 256$ is around 4 times larger than that obtained without synchronization uncertainty for most array sizes and with both algorithms. Overall, the proposed ToA-based ML estimator is capable of dealing with larger clock offsets, although it should be noted that it is more accurate than the SRP-based algorithm to begin with.

The results for the lower values of σ_Δ (≤ 13 samples) are interesting since they do not produce a significant difference in the localization results. This is most likely due to the error inherent to the microphone and source location measures. During the recording of the database, special care was taken to place the microphones at the same height, nevertheless, the 2D space assumption in conjunction with manual distance measurements are a certain source of small error. Synchronization uncertainty can also be understood as an equivalent spatial uncertainty. For the tested values of σ_Δ, from smaller to larger, it corresponds to approximately: 5 cm, 10 cm, 0.5 m and 2 m, at $f_s = 44100$Hz and with $c = 345$ m/s. Note that neither of the tested localization algorithms takes into consideration the existence of clock offsets between the channels, thus, timing differences have the same effect as their equivalent error in terms of microphone locations.

Generally speaking, these results prove that synchronization is a main concern for acoustic source localization in WASNs. Clock offsets of just 1.5 ms (≈ 64 samples at $f_s = 44100$Hz) can easily double the localization error, thus, tight synchronization within the network is a basic requirement for acoustic source localization.

On a side note; although we have not tested it, the proposed ML estimator can be modified to also estimate clock offsets. However, for this to work multiple sets of ToAs need to be obtained from different sources before the minimization can be solved, and the convergence rate and computational complexity of said minimization would be notably affected. Taking it even further, after collecting ToA estimates from a large enough number of sources, even microphone locations could be estimated. However, in that scenario the algorithm would become a passive self-localization method, similar to those proposed in [GKH13] or [PP17], which falls outside the scope of this chapter.

5.8 Conclussions

In this chapter two efficient methods for sound source localization have been proposed. The first method is a modification of the classic approach based on cross-correlating signals captured by different microphones. It takes advantage of the presence of a proper multichannel sound event detector in order to select a shorter signal segment that that dictated by the sound propagation model, thus increasing efficiency. The second method completely bypasses the need for cross-correlations and works with a set ToA estimates of some acoustic event obtained at every channel also by exploiting the detection capabilities of the system.

The main novelty resides in the ToA-based method, it demonstrates how the solution to a seemingly independent problem (i.e., sound event detection) can be exploited to formulate new solutions to a classic multichannel signal processing problem. Furthermore, the experiments showed that the ToA-based method is more reliable (specially the Gaussian version) for the kind of signals expected from impulsive noise sources recorded outdoors. Also, in terms of efficiency the ToA-based proposal is superior to the GCC-based approach, even more so when the scalability of the algorithms is taken into account.

Both proposed methods tackle some of the problems associated with performing classical source localization in WASN where large inter-microphone distances call for long cross-correlations that are heavily affected by coherence loss between the signals. The presented results show that the main problem is again the large differences between propagation paths that make signals captured at distant locations no longer resemble a time lagged version of one another.

The effect of synchronization uncertainty took a second place in the experiments, mainly due to the recordings used for testing the algorithms being properly synchronized. However, synchronization together with accurate knowledge about the microphone locations are part of the basic requirements for most source localization algorithms. In fact, the experiments with simulated clock offsets, show that the microphone and source location measurements used as the ground truth are not free of error, thus the presented results are inherently affected by them.

It is important to mention that although source localization is not often required to be a real-time process, the faster its solution the better. Sound source localization is closely related to self-localization, so much so that the former can be seen as a simplified version of the latter. Contrary to self-localization, which ideally only has to be solved when the WASN undergoes some physical modification (i.e., changing the geometry or adding a new node) and it is a vital step for most multichannel algorithms, source localization is less critical and more frequent. That is, failing to locate some acoustic event does not represent a critical failure for the system since additional information can still be extracted from the received sound (i.e., identifying the type of source), and if the source were to produce any new acoustic events that would represent a new chance to locate it. Given the close relation of source localization and self-localization, is only natural that some of the conclusions extracted on chapter 3 can also be applied to the present chapter. Most notably, assessing the quality of the estimates is still a main issue. With any of the tested algorithms the results could be improved using a weighted optimization approach where the ‘quality’ of each estimate is taken into consideration.

Future work will (again) address spurious estimates, and contemplate possible transmission errors and other compilations arising from wireless data transmissions (e.g., collisions), what were not considered in the present version. In addition to this, the validity of the results should be proven by repeating the experiments with a larger database. Such a database should include more SNR conditions, a larger number of elements (both microphones and sources) and most importantly, greater source location diversity that contemplates new non-ideal situations (e.g., non direct line of sight, very large source-microphone distances, multiple undesired sources, etc.).

5.9 Summary

The main contributions described in this chapter of the thesis are the following:

1. An efficient method to perform GCC-based localization that exploits local detections to minimize the length of the signals being cross-correlated.
2. A simple unitary step fitting approach to obtain the onset of the local detections.
3. A novel ML-based source location estimator, that works with ToA estimates, hence, bypassing the need for cross-correlations.
4. Two solutions for the ML estimator that assume either a Gaussian PDF or a Laplace PDF that can be easily modified to absorb synchronization problems.
5. A simple method for obtaining the onset of the events in the time-domain (ToA) based on curve fitting.

The contributions described in this chapter are being prepared for submission to a peer reviewed journal. Further details can be found in [P5] in section: Publications.

CHAPTER 6 Impulsive Event Classification

6.1 Introduction

The last problem that is going to be addressed in this thesis is the classification of impulsive audio events. As part of an automated acoustic surveillance system, the objective of classification is to tell apart sound events signaling a potentially dangerous situation from similar but 'harmless' sounds (e.g., a window being broken vs a coffee mug shattering). The line between classification and detection gets blurred when these techniques are to be applied to continuous stream of data and in that case, perhaps multi-class detection is a more proper term. Although, if we consider a hierarchical system in which a generalist sound event detection system is followed by a classification stage, then differentiating between both terms still makes sense since classification would work with segmented acoustic events.

Impulsive sound event classification poses a harder problem than detection, specially when applied within a same "family" of sounds. This is mainly due to the differences between impulsive sources being much less pronounced than the differences between impulsive noise and other sounds, and on top of that, the captured signals are strongly related to the recording environment and the positioning of the sources and receivers. Any acoustic event recorded at a distance from the source is going to be convoluted with the environment and mixed with unwanted noise which negatively affects the accuracy of the classification system, because it is not possible to predict all the possible variations a signal might be subjected to during the training stage. Albeit, this problem can be lessen by using a spatially diverse receiver such as a WASN. Such a system can easily provide multiple observations of an acoustic event taken at different locations, making it possible to obtain a more robust classification by fusing the available data. Even more so, if the spatial relation between the source and the receivers is known, this information can be used to aid the classification process in different ways.

In this chapter the multichannel classification problem is tackled in two different approaches: first, assuming that the network geometry is unknown due to technical limitations, hence, the spatial relations have to be estimated by every node in isolation; and second, assuming the various functionalities described in the previous chapters of this thesis have been successfully implemented, therefore, both node and source locations are already known.

6.2 Unknown Spatial Relations

In an ideal scenario the locations of the nodes and the source would be known *a priori* so that various array-based strategies could be applied to aid the classification. These could range from enhancing the target signal using beamforming to, having the source localized, selecting the decision taken by the closest node as the most trustworthy. However, as we have seen in previous chapters of this thesis, in the context of a WASN, precise node localization and inter-node synchronization are basic requirements (for array processing) that are not easily met.

Even when the lack of internode synchronization or the poor knowledge about the location of the nodes preclude the use of classic array processing, having multiple spatially diverse information sources can be successfully exploited using decision fusion. Assuming that every node in the network is capable of perfect detection, then the global decision regarding the class of a detected acoustic event can be obtained using decision fusion without the need for synchronization. Asynchronous decision fusion can easily implemented just by setting a time window (starting from the first detection) inside which, those nodes that have detected the event can share their local decision to be fused. This should be seen as an oversimplification, but serves to illustrate a possible solution problem. Going further, it is also possible for the nodes to extract some spatial information about an acoustic event on their own. A single channel system can be able to (approximately) tell if an impulsive source is close or far away based on various cues such as: SNR, reverberation level or even spectral content (i.e.,long propagation paths have a low pass effect), which could also prove beneficial.

In this first approach we are going to tackle gunshot classification on a WASN with unknown geometry assuming that proper gunshot detection is already available. For this scenario, we propose to have each node estimate their relative to the source so that spatial information can also be used during the decision fusion stage.

Weapon acoustic analysis has practical applications in many fields such as forensics, security, gun control or military tactics to name a few. Thus, the acoustic signature produced by explosive propelled weapons has been subject of study for some decades now [WK74b,FTCP93,Mah07]. In recent years, this field has become more relevant mainly due to the development of sniper detection and localization systems [KPD06] aided by sensor fusion techniques and WASN. Renewed interest in this topic has produced multiple approaches to gunshot detection over the last decade. Whereas, acoustic weapon classification has not been widely studied yet, hence there are few available precedents (e.g., [CER05,SHV$^+$11]) and the spatial dependence of gunshots is rarely addressed.

The biggest problem in this field, on top of the problems common to any impulsive noise, is the strong dependence on the shooter's location and orientation shown by the recorded waveforms, mostly because the acoustic disturbance created by firearms is highly directional [MS10]. In fact, this high directivity combined with undesired acoustic phenomena, commonly makes the differences between recordings of the same weapon at distant locations greater than those of different weapons recorded at the same location. Even recordings of the same gun firing different ammunition can show large differences [BNM11], making acoustic gunshot classification a difficult task.

6.2.1 Gunshot Acoustic Model

Common firearms produce their characteristic sound as a result of the sudden expansion of gases generated at the end of their barrel by the explosive charge used to propel the projectile, formally known as 'muzzle blast'. A simple approach that can be used to understand the acoustical excitation produced by this kind of phenomenon is *Weber's spectrum* model, accounted for in ISO norm 17201-2

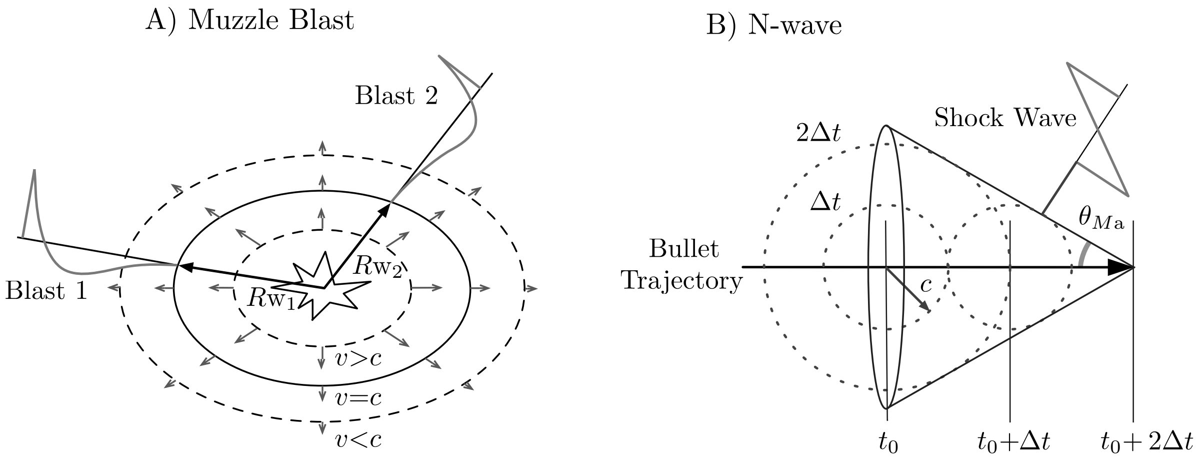


Figure 6.1: Schematic representation of a gunshot sound components. A) Blast wave radiation from an ellipsoid volume at two points. B) Geometric model of shock wave propagation.

[IC06]. This model gives us an estimation of the Fourier spectrum of a blast wave on free air [FBP06] as a function of the radius of the expanding gas sphere (Rw) created by the charge in the precise instant that its propagation speed decreases enough to match the speed of sound (c). The energy of the explosion is directly related to the volume of displaced gases, shifting the spectrum to lower frequencies as the radius of the sphere increases. However, in the case of firearms, the constraining effect of the barrel on the expansion of gases has a big impact on the emitted sound, making the muzzle blast strongly directional. Applying Weber's radius model, directivity can be explained as a result of the divergence in the expelled gases shape from a perfect sphere, implying a dependence between the listener location and the perceived radius [KWH13]. Figure 6.1-A shows the differences in amplitude and time duration of the blast waves created by an ellipsoid volume at two points.

The second main component of a gunshot acoustic signature is the shock wave produced by a projectile traveling at supersonic speed. While the muzzle blast can be seen as a global event, due to the extensive range reached by the acoustical excitation generated by the propeller explosion, shock waves have a local influence (i.e., for small sized bodies), they only appear when the microphone is close enough to the trajectory of the projectile. For a projectile with a speed $V > c$, defining *Mach number* as $M\mathrm{a} = V/c$, the generated shock wave propagates in conic shape forming an angle $\theta_{M\mathrm{a}} = \arcsin(1/M\mathrm{a})$ with the bullet trajectory as shown in Figure 6.1-B. This acoustic disturbance is commonly known as 'N-wave' due to its characteristic geometry resembling a capital "N". Its most relevant parameter, its duration, can be approximated by knowing the physical dimensions of the bullet, its velocity and the closest distance between the microphone and the projectile trajectory. These physical relations are the basis for acoustic projectile trajectory estimation algorithms[1].

It is important to mention that in a real scenario the recorded waveforms may be very different from that described by the model, due to non-idealities such as reflections, signal saturation and most of all, the influence of the propagation path. Figure 6.2 illustrates this claim with two recordings of a .45 caliber handgun gunshot taken at distant locations.

[1] N-waves can not be used to estimate the shooter's location since they are emitted by the projectile

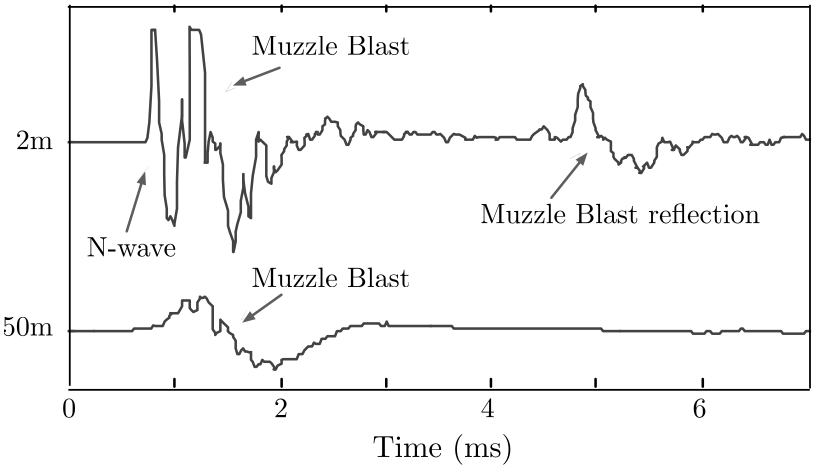


Figure 6.2: Recorded waveforms of a .45 caliber handgun gunshot at two distant locations.

6.2.2 Model-based Features

In contrast to detection, classification is not necessarily a continuous process. It can be integrated as part of hierarchical system governed by acoustic event detection in the same way as the source localization system proposed in the previous chapter. Coincidentally, this is a fairly common approach for gunshot detection, where many proposals use a simple acoustic event detector followed by a classification stage (gunshot vs other sounds) based on classic pattern recognition techniques [CER05, FAJ10, AUM13].

Under an assumption of perfect detection, given that the target acoustic event has already been located in time by a detector, there is no longer need for instantaneous features. Since the acoustic signature of gunshots has a short time duration (i.e., direct signal without considering reverberation), it makes sense to compute the feature vector from a signal segment that encompass the whole signal and use global descriptors. Note that in this scenario, adding temporal context to the feature vector simply by stacking previous frames becomes less relevant so that different approaches may be preferred.

In this particular case we propose to compute a set of specialized global features based on the ideal acoustic model of a gunshot, intended to capture some temporal information that is not clearly reflected by standard features. We refer to theses features as model-based features.

Model-based features are computed from a signal segment ($s(t)$) selected using the gunshot detection that ideally includes the 'whole' direct signal of the muzzle blast. The first model-based feature is the the TDoA between the N-wave and the muzzle blast. Since the muzzle blast is always appearing (perfect detection is assumed) and it is the main source of energy, a secondary energy source preceding it with a lower energy level and shorter time duration must be an N-wave. N-waves only appear with supersonic ammunition, and even then, the microphone has to be close enough to the trajectory of the projectile to capture it. From here, two conclusions can be made: weapons firing subsonic ammunition (such as shotguns) do not have N-waves and neither do those recordings taken from the back of shooter. On top of this, the longer the distance to the shooter the larger the TDoA between the N-wave and the muzzle blast. This information is obtained using the

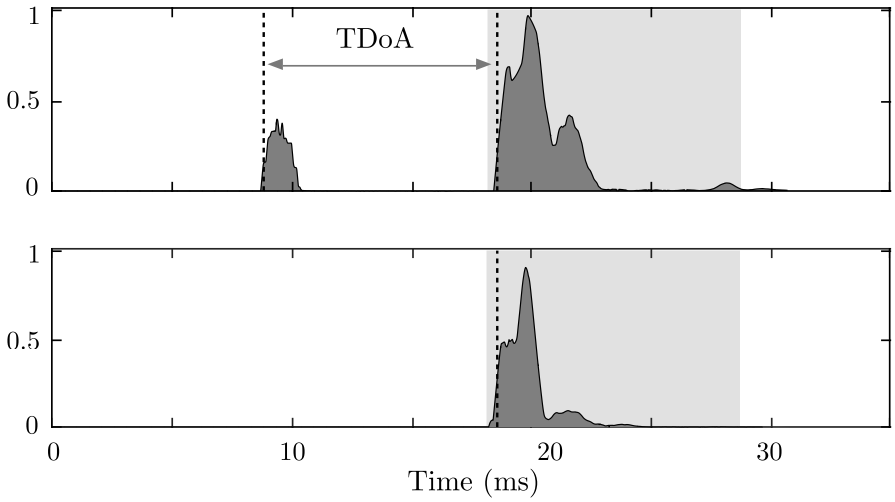


Figure 6.3: Energy moving average of a gunshot recorded at two locations. Signal segment containing the muzzle blast represented by a light gray area.

TDoA between energy clusters obtained from the energy moving average of the raw audio signal. In case only one cluster appears, it is set to a default value (zero). These two scenarios are represented of Figure 6.3 which shows the energy moving average of a gunshot recorded at two locations.

The remaining model-based features are focused on the muzzle blast waveform. They have more to do with the spatial aspect of the problem since the shape of the muzzle blast is a good range indicator (overlapping reflections at close range and low-pass effects at long range).

We propose to extract some shape descriptors using the most prominent tipping points (peaks and valleys) of the selected signal segment ($s(t)$). Since raw recordings of gunshots are prone to be quite noisy due to turbulences, peak finding can be eased by first smoothing $s(t)$. The typical approach in such cases is to low-pass filter the signal, so that high frequency noise is eliminated while the basic 'shape' of the signal is maintained, although, low-pass filtering affects the steepness of rising edges in fast transients which is far from ideal for this particular case. To avoid this problem we propose to process the signal segment using Total Variation Denoising (TVD) [FDON06], also known as total variation regularization. TVD gets rid of the small variations in the signal but unlike low-pass filtering it preserves sharp edges, it is commonly used for image processing (on its 2D form) but it can also be applied to 1D signals.

Once the tipping points have been found, taking the index value of the first two (p_1 and p_2), their Time Difference (TD) can be used as a representative value of the duration of the decay of the muzzle blast (i.e., the duration of the positive phase), and their amplitude ratio (A_1/A_2) as a symmetry measurement. Additionally, by finding the zero-crossing points of the the first cycle they can be used to compute the half cycle ratio (TS_1/TS_2) of the muzzle blast. These three features are related to the Weber's radius of the muzzle blast, that although it depends on the shooter's orientation, it is also a good indicative of the weapon type. The number of tipping points and the time difference between the first and the last found tipping points can also be used as features, since they are a good indicator of the presence of early reflections (most likely from the ground) which in turn gives a good measure of the distance of the microphone to the discharge (i.e., on long range recordings the reflections are likely to be outside of the selected signal segment). Finally, the

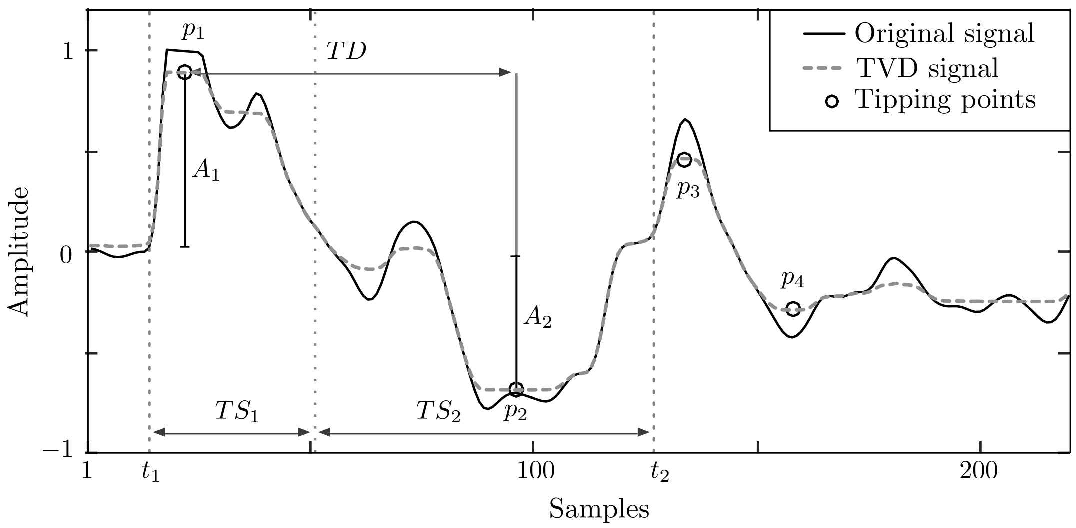


Figure 6.4: Original signal, TVD processed signal and reference points for model-based feature extraction.

energy of the first cycle of the signal can also be computed ($20\log_{10}(\sum_{n=t_1}^{t_2} s(t)^2)$), which ideally corresponds to the muzzle blast. This last feature is somewhat redundant but in conjunction with other features, it can provide some information about the range or the energy level of the weapon being fired.

Figure 6.4 shows a TVD processed signal and the reference points used for computing the proposed model-based features

The proposed model-based features are:

- TDoA between the N-wave and the muzzle blast.
- 6 muzzle blast shape descriptors:
  - Time difference of the tipping points of the first cycle.
  - Amplitude ratio of the first cycle.
  - Half cycle ratio of the first cycle.
  - Number of tipping points found in the signal segment.
  - Time difference between the first and last tipping points.
  - Log-energy of the first cycle.

6.3 Spatial Segmentation

In order to take advantage of spatial information without resorting to classic localization algorithms, we have reformulated some of the classic problems of the field, turning them into simpler problems that do not require the use of multiple information sources to be solved.

Using a single microphone, it is no longer possible to triangulate the shooter's location. However, some information can be obtained by making a rough estimation of his/her proximity to the node.

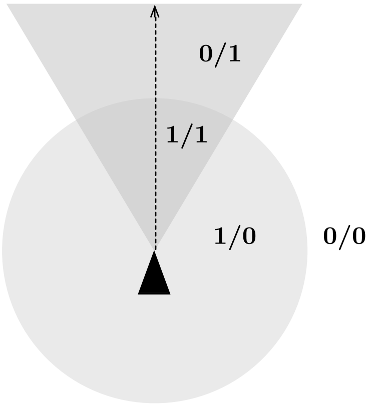


Figure 6.5: Schematic representation of the spatial segmentation performed by the binary classifiers (Range/Aligment).

For the time being, we are discerning between close and long range discharges, although, the proposed methodology is valid for further subdivisions. Trajectory estimation also suffers from the lack of spatial references. It has been replaced by a rough estimate of the proximity of the sensor to the trajectory of the bullet into two broad alignment categories: on-axis and off-axis. On-axis implies that N-waves might appear at a given microphone location, while off-axis represents any other location.

We propose to segment the space into four regions by employing the outcomes of a pair of binary classifiers (range and alignment), so that each node has a certain degree of knowledge about its location in relation to the gunshot that can be exploited. Figure 6.5 shows a schematic representation of the spatial segmentation.

Although segmenting the space into four regions does not seem like much, it is enough to improve the accuracy of local classifications by means of a Divide and Conquer approach. Given the big differences between gunshots recorded at distant locations, specialized classifiers can be trained to work within each region so that the differences between weapons become more relevant than the differences between locations. On top of that, by having various non-identical classifiers in the ensemble, it is possible to derive a new fusion rule to take advantage of that situation and improve the global classification.

6.3.1 LS-LDA Classification Tree

Since spatial effects on the signals are the main problem for the classification, we propose to analyze different spatial regions independently. This is achieved with a classification tree that uses the outcome of two binary classifiers to segment the space by selecting between a set of localized weapon classifiers. The localized classifiers are designed using a specific subset of events, so that, they do not contemplate the existence of the other regions. The main idea is the same one behind the K-LDA classifier proposed in chapter 4, only this time, the feature space segmentation is performed using the outcomes of two binary classifiers that happen to be assessing the source-node spatial relation. Likewise, we are using LS-LDA on every stage of the classification tree for the sake of efficiency. Although the same approach could be used with any other classifier.

Multi-class LS-LDA is commonly tackled defining C binary classes and applying a one-against-all scheme, which entails that only one class is labeled as true for a given observation (one-hot encoding). This way, the output is a vector $\mathbf{y} = [y_1, ..., y_C]^T$ where each element represents the "score" of one of the classes, which can be seen as the estimated probabilities of the observation belonging to each class. Final decision D is then taken by selecting the largest element of $\mathbf{y}$ with:

$$D = \arg\max_c (y_c). \tag{6.1}$$

Let us consider a training set composed of N observations, where each element is $\mathbf{q}_i = [1, x_{i1}, ..., x_{iF}]^T$, $i = 1, ..., O$. In matrix form, we can define an input matrix $\mathbf{Q}$ of dimensions $(F+1) \times O$ and a weight matrix $\mathbf{V}$ of dimensions $C \times (F+1)$, where C is the number of target classes, so that the output is computed as $\mathbf{Y} = \mathbf{V} \cdot \mathbf{Q}$. If we also define a binary target matrix $\mathbf{T}$ of dimensions $C \times O$ representing the desired outputs (true class labels in one-hot encoding), the error can be expressed as $\mathbf{E} = \mathbf{V} \cdot \mathbf{Q} - \mathbf{T}$.

In order to obtain the weight matrices for the classifier tree, let us begin with the first stage by defining two target vectors $\mathbf{t}_A$ and $\mathbf{t}_B$ of length O that represent the true range and alignment labels respectively. From these two vectors and input matrix $\mathbf{Q}$, using expression (2.45), the weights for the first stage are computed with:

$$\mathbf{v}_A = \mathbf{t}_A \mathbf{Q}^T (\mathbf{Q}\mathbf{Q}^T)^{-1}, \text{ and } \mathbf{v}_B = \mathbf{t}_B \mathbf{Q}^T (\mathbf{Q}\mathbf{Q}^T)^{-1}. \tag{6.2}$$

From where the range and alignment estimations of each observation are obtained as:

$$\mathrm{R}_i = \begin{cases} 1 \text{ if } \mathbf{v}_A \mathbf{q}_i > 0.5 \\ 0 \text{ if } \mathbf{v}_A \mathbf{q}_i \leq 0.5 \end{cases}, \text{ and: } \mathrm{A}_i = \begin{cases} 1 \text{ if } \mathbf{v}_B \mathbf{q}_i > 0.5 \\ 0 \text{ if } \mathbf{v}_B \mathbf{q}_i \leq 0.5 \end{cases}. \tag{6.3}$$

Now, using the outputs of the first stage, let us make a partition of $\mathbf{Q}$ into four subsets $\mathbf{Q}_k$, $k = 1, ..., 4$ containing the observations of each of the four spatial regions and defined as:

$$\mathbf{Q}_1 = \{\mathbf{q}_i | \ \mathrm{R}_i = 0 \text{ and } \mathrm{A}_i = 0\}, \tag{6.4}$$

$$\mathbf{Q}_2 = \{\mathbf{q}_i | \ \mathrm{R}_i = 0 \text{ and } \mathrm{A}_i = 1\}, \tag{6.5}$$

$$\mathbf{Q}_3 = \{\mathbf{q}_i | \ \mathrm{R}_i = 1 \text{ and } \mathrm{A}_i = 0\}, \tag{6.6}$$

$$\mathbf{Q}_4 = \{\mathbf{q}_i | \ \mathrm{R}_i = 1 \text{ and } \mathrm{A}_i = 1\}. \tag{6.7}$$

The reason why we are using the outputs of the first stage instead of the true range and alignment labels for the division of the training set, is to lessen the restrictions on the segmentation by letting the classifiers select which observations belong where.

Since each of the observations belongs to one of the 4 subsets and has a true weapon label associated, we can define matrix $\mathbf{T}_k$ as the target matrix of those observations within $\mathbf{Q}_k$, so that the weight matrices of the localized weapon classifiers are obtained with:

$$\mathbf{V}_k = \mathbf{T}_k {\mathbf{Q}_k}^T (\mathbf{Q}_k {\mathbf{Q}_k}^T)^{-1}, k = 1, ..., 4. \tag{6.8}$$

Note that this expression is the same as (4.3), thus, the spatial segmentation being performed here is equivalent to the feature space segmentation employed in the K-LDA classifier.

6.4 Maximum Likelihood Decision Fusion

We propose a fusion rule based on ML estimation that takes advantage of a diverse classifier ensemble to improve upon classic decision fusion techniques. This decision rule is based on giving different weights to the decisions taken by each classifier in the ensemble, based on a discrete set of performances known *a priori*, which in this case are derived from the location of a given node in relation to the source.

Let us consider that a set of estimations of the likelihood of a given acoustic event belonging to each class is available, where the estimates produced by each node have an associated error that depends on its location in relation to the source's location (i.e., the propagation path), on which classifier among a discrete set was used, and also on the true class of the event. The objective is to obtain the best possible estimate of the true class of the event by fusing the available information. Let us also consider that the outputs of the m^{th} node $\mathbf{y}_m = [y_{m1}, ..., y_{mC}]$ follow a gaussian distribution so that its PDF is given by:

$$\mathcal{F}(\mathbf{y}_m; \mathbf{t}, \mathbf{C}_m) = \frac{1}{\sqrt{(2\pi)^C |\mathbf{C}_m|}} e^{\left(-\frac{1}{2}(\mathbf{y}_m - \mathbf{t})^T \mathbf{C}_m^{-1} (\mathbf{y}_m - \mathbf{t})\right)}, \tag{6.9}$$

where $\mathbf{t}$ is a binary target vector (i.e., class labels) and $\mathbf{C}_m$ is the covariance matrix of the PDF. In this general case we consider different covariance matrices for each member of the ensemble.

Following this approach it is possible to to use a ML estimator to obtain the most likely class of some acoustic event $(\mathbf{t})$ given a set of observations $\mathbf{y}_m$. Thus, the solution can be found by maximizing the the log-likelihood $\log(\mathcal{L})$ of some observation set, which is given by:

$$\log(\mathcal{L}) = \sum_{m=1}^{M} \log(f(\mathbf{y}_m; \mathbf{t}, \mathbf{C}_m)). \tag{6.10}$$

Replacing equation (6.9) in (6.10) and simplifying, the expression to be maximized is obtained:

$$LL = \frac{1}{2} \sum_{m=1}^{M} \Big(-\log((2\pi)^C |\mathbf{C}_m|) - (\mathbf{y}_m - \mathbf{t})^T \mathbf{C}_m^{-1} (\mathbf{y}_m - \mathbf{t})\Big). \tag{6.11}$$

In order to maximize $\log(\mathcal{L})$, expression (6.11) has to be differentiated with respect to $\mathbf{t}$ and equaled to zero, leading to the following system of equations:

$$\sum_{m=1}^{M} \mathbf{C}_m^{-1} \mathbf{t} = \sum_{m=1}^{M} \mathbf{C}_m^{-1} \mathbf{y}_m, \tag{6.12}$$

from where, $\mathbf{t}$ can be cleared arriving to:

$$\mathbf{t} = \sum_{m=1}^{M} \left(\sum_{m=1}^{M} \mathbf{C}_m^{-1} \right)^{-1} \mathbf{C}_m^{-1} \mathbf{y}_m. \tag{6.13}$$

At this point, the solution can be easily obtained with the following expression:

$$D = \arg\max_{c}(\mathbf{t}), \tag{6.14}$$

Finally, we can extrapolate this conclusion to define a fusion rule for our classifier ensemble, so that node weights are obtained with:

$$\mathbf{W}_m = \left(\sum_{m=1}^{M} \mathbf{C}_m^{-1} \right)^{-1} \mathbf{C}_m^{-1}, \tag{6.15}$$

where $\mathbf{W}_m$ is a $C \times C$ matrix and the weights for each node and class are found using all the covariance matrices of the ensemble. Now, defining an auxiliary vector $\mathbf{z}_m = \mathbf{W}_m \mathbf{y}_m$, the final decision becomes:

$$D = \arg\max_{c} \left(\sum_{m=1}^{M} z_{mc} \right). \tag{6.16}$$

In the case of a multi-observation classification ensemble, where all the observations are evaluated with the same classifier and influence of node locations has not been characterized, we have $\mathbf{C}_m = \mathbf{C}$, thus, the presented fusion method becomes equivalent to average fusion. Defining $\mathbf{I}$ as the identy matrix and $\mathbf{B} = \mathbf{C}^{-1}$, we know that $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$ and $(\alpha\mathbf{B})^{-1}\mathbf{B} = (1/\alpha)\mathbf{I}$, therefore $w_m = 1/M$. This result does not hold true when there is some degree of known error diversity and $\mathbf{C}_m$ takes different values.

6.4.1 Covariance Matrix estimation

In order to take advantage of the ML decision fusion, we first need to characterize the covariance matrix of every classifier on the ensemble. Let us assume a hypothetical scenario in which both the location of the nodes and the sources is fixed. It would then be possible for each node to use its own classifier trained for that particular scenario, from where $\mathbf{C}_m$ could be easily estimated. However, on a real scenario, the relative position of the sources and the nodes is prone to change, notably affecting the classification accuracy. Albeit, if we were able to properly characterize the relation between the classifier output and the relative node-source location we could then estimate $\mathbf{C}_m$ as a function of the relative location, even when there is a single classifier for the whole ensemble similarly to [DH04].

For the time being, let us consider some discrete spatial division composed of K zones. Then, for each region, the classifier has an associated covariance matrix $\mathbf{C}_k$, $k = 1, .., K$. This entails that, for every new observation, each node in the ensemble determines in which in zone it resides and assigns $\mathbf{C}_m$ to one of the possible $\mathbf{C}_k$ based on this decision. Note that this solution is scalable, the more spatial information available about the source, the greater the spatial segmentation can be.

In order to compute $\mathbf{C}_k$, lets consider an output matrix $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and a target matrix $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_M]$ so that the class estimation error is $\mathbf{E} = \mathbf{Y} - \mathbf{T}$. The sample covariance of the estimation is:

$$\mathbf{C} = \frac{1}{N}(\mathbf{Y} - \mathbf{T})(\mathbf{Y} - \mathbf{T})^T = \frac{1}{N}\mathbf{E}\mathbf{E}^T, \tag{6.17}$$

For the LS-LDA tree, replacing in this expression with the variables from 6.3.1 we can estimate the covariance matrix of the specialized classifiers as $\mathbf{C}_k = \frac{1}{N}\mathbf{E}_k{\mathbf{E}_k}^T$, where $\mathbf{E}_k = \mathbf{V}_k\mathbf{Q}_k - \mathbf{T}_k$.
Since training the classifier using a mean-square-error criterion gives outputs that approximate posterior class probabilities [Gis90] it is good practice to saturate the outputs using the target interval as limits (probability 0 to 1) before computing the error.

We have found that for those scenarios in which computational complexity is an issue, it is possible to discard the values of $\mathbf{C}_k$ that are not part of the main diagonal which is equivalent to assuming uncorrelated classes. While this method yields slightly worse results, it has the benefit of turning matrix operations into element wise operations, since by doing this, matrix $\mathbf{W}_m$ becomes vector $\mathbf{w}_m$. This is specially relevant for the inversion of the sum. Given that the values of $\mathbf{C}_k$ are fixed during training, they can be stored as $\mathbf{B}_k = \mathbf{C}_k^{-1}$, making the inversion of the sum the only matrix inversion computed during run time. Expanding further into computational complexity, it is possible to completely avoid matrix inversions by storing all the possible values of $\left(\sum_{m=1}^{M}\mathbf{C}_m^{-1}\right)^{-1}$ on a lookup table. In a system with M maximum nodes and K possible covariance matrices, adding a zero matrix $\mathbf{C}_0$ as an additional covariance matrix, the table length can be found as the number of combinations with repetition using:

$$\binom{(K+1)+M-1}{M} - 1 = \frac{(K+M)!}{M!K!} - 1, \tag{6.18}$$

where $\binom{a}{b}$ represents the combinatorial number. The table is valid for any WASN with a number of nodes less or equal to M, since $\mathbf{C}_0$ is added in order to consider null contributions to the sum. This solution is a clear tradeoff between memory and computing power, so whether or not it is advisable to use a lookup table will depend on the target platform. With $M = 8$ and $K = 4$ there are 494 possible combinations. If $C = 3$ and considering single-precision (32-bit) floating-point representation, the table would require 17.37 kilobytes of memory.

6.5 Experimental Work and Results: Unknown Geometry

After introducing the specifics of the proposed gunshot classification system, in this section we will describe the experimental setup and the obtained results. We are working with three broad categories: rifles, handguns and shotguns.

6.5.1 Database Description

The gunshot sounds that make up the database used for evaluating impulsive sound event classification with unknown spatial relations are commercially available as part of a sound library offered by the company *BOOM Library* under the name *“GUNS - Construction Kit”* [BOO11].

This sound library is intended for the creation of sound effects for audiovisual media. It contains multiple recordings of 17 weapons (7 rifles, 6 handguns and 4 shotguns) including gunshots and various mechanical noises. All the signals included in the library are recorded at 96000Hz using various high-quality microphones and recording equipment. Raw gunshot recordings of each weapon are available at 12 locations, 2 of which were excluded of the database; one of them because the recording location is not the same for all weapons and another one because the recordings are highly saturated and do not hold any resemblance to the model described in section 6.2.1. In addition to this, 3 weapons were excluded from the database. This was done mainly to keep balance within classes, that is, to keep the number of weapons within each class as similar as possible. Another reason was to maintain coherence within the “rifle” class in particular. While 5 of the included rifles are high power military weapons, 2 of them are low caliber (.22) sporting carbines, which are completely different weapons. Since at that point we had 4 shotguns and 5 rifles, we decided to dismiss one of the handguns to keep the database balanced as previously stated. The excluded handgun is a lower caliber version of another available handgun.

The vendor of the sound library offers information about every sound, including the weapon model, which microphone was used and a vague description of its location. Further information regarding the recording hardware is not available. While the provided spatial information is indeed valuable, it is not detailed enough to tackle location-based fusion or array processing for that matter. On top of this, various microphone types were used for the recordings, thus there are differences between recordings in terms of frequency response and directivity. Having different microphones can be beneficial towards diversity, although, it is far from ideal for array processing, where the common approach is to assume identical microphones.

The database generated from this sound library contains recordings of 14 weapons: 5 handguns, 5 rifles and 4 shotguns. There are 6 recordings (shot repetitions) of every weapon at 10 different locations (6 observations per weapon-location combination), adding up to a total of 840 gunshots. Of the 10 unique locations, 4 are labeled as short-range and 6 as medium range, whereas 6 are labeled as on-axis and 4 as off-axis. N-waves only appear in 22.1% of the recordings, not appearing at all for 6 of the weapons (2 handguns and every shotgun) given that they use subsonic ammunition.

6.5.2 Description of the Experiments

The proposed feature set is composed of 29 features: 22 standard features and 7 model-based features. It is computed using a signal segment $s(t)$ of length 10.7 ms (1024 samples at 96kHz) that contains the muzzle blast. This segment is selected from the starting point of the Muzzle blast which is detected using a moving average of the squared input signal with a rectangular window (length 0.67ms), shown in Figure 6.3 as a light gray area. The standard features are composed of two temporal features extracted from $s(t)$ and a series of spectral descriptors extracted from its DFT $S(k)$. From $s(t)$ we compute its energy level in decibels (as $20\log_{10}(\sum s^2(t))$) and its zero-crossing rate. From $S(k)$ we compute 16 MFCCs, and we extract 4 spectral descriptors, namely: centroid, kurtosis, slope and roll-off [Ler12]. Model-based features are computed as described in section 6.2.2. The tipping points of each signal segment are obtained from the TVD processed signal using Matlab’s *findpeaks* function (the negative section is first inverted). Local peaks are avoided by setting conditions to the peak finding algorithm (based on heuristics), specifically, minimum peak prominence and minimum distance between peaks.

Regarding training, our objective is to test the system in as close to real conditions as possible. This implies that, for any given gunshot, both the shooter's location and the fired weapon are going to be new to the system. Since our database is not large enough for the complexity of the problem, a simple division of the data into a train set and test set is not efficient. In order to obtain the presented results we used a hybrid cross validation method that mixes LOOCV [Efr79] and *Repeated random sub-sampling validation*.

In every iteration the database is randomly divided into a train set and a test set equally sized in terms of included positions (5 positions per set). Weapons are tested using LOOCV so that the test set is formed by the sounds of one gun at 5 positions while the training set contains the sounds of the remaining weapons at the remaining positions (13 guns × 5 positions). This is done in order to maximize the information available to the classifier about weapon classes while testing it with a previously "unheard" gun recorded at unknown locations. To maintain class balance, in every iteration, b random sounds of the train set that belong to the tested class are duplicated, b being the number of observations in the test set. For each of the guns, r random location-wise database divisions are tested. It is important to remark that since there are only 5 microphone locations per division, we have decided to consider the 6 observations available per gun-location ($\mathbf{o}_{g,l}$) as additional microphone locations. Treating repetitions as different locations is justified by assuming they are recordings of the same gunshot taken at 6 locations very close to each other. In total we consider 30 possible locations (5 locations × 6 repetitions).

The final step involves the generation of p random permutations of the test set without repetition, that is, sorting the observations included of the test set randomly. For each permutation we obtain the m^{th} ensemble error rate by fusing the decisions obtained for the first $m = 1, ..., M$ observations. The final results are computed averaging the errors obtained in the whole $14 \times r \times p$ experiments for each ensemble size. The presented data uses $r = 32$ and $p = 8$ adding up to a total of 3584 iterations.

Fig. 6.6 shows the r^{th} division for the third gun ($n = 3$), the sounds highlighted in light gray are used to train the classifiers while the sounds highlighted in dark gray are arranged in p random permutations in order to test the system. Each cell in the table contains the six observations available per gun-location combination. Notice that on each iteration half of the sounds remain unused.

The covariance matrices of the LS-LDA (full-covariance matrices) tree were obtained from the train set using LOOCV with weapon-location combinations (13 guns × 5 positions repetitions), leaving out all the available sounds of a gun at a single position and training with the rest. We opted to compute the train error using cross validation since the results are less prone to overfitting and so, it yields a closer estimation to the covariance values found when testing the system with new observations. As we have previously mentioned, the test set was never used in the design phase, including the estimation of the covariance matrices.

The results were obtained under the assumption of perfect detection, hence, every observation belongs to one of 3 target classes. The evaluation metric is error rate, or the percentage of incorrect predictions made by the classifier over all the predictions made.

The features are normalized (between 0 and 1 according to the training set) prior to the training. For the experiments, first we compared the performance of the LS-LDA tree and some well

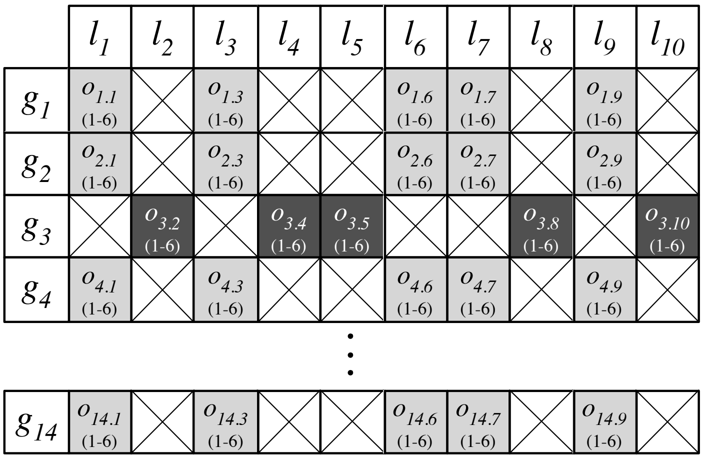

l_1 l_2 l_3 l_4 l_5 l_6 l_7 l_8 l_9 l_{10}
g_1 $o_{1.1}$ (1-6) $o_{1.3}$ (1-6) $o_{1.6}$ (1-6) $o_{1.7}$ (1-6) $o_{1.9}$ (1-6)
g_2 $o_{2.1}$ (1-6) $o_{2.3}$ (1-6) $o_{2.6}$ (1-6) $o_{2.7}$ (1-6) $o_{2.9}$ (1-6)
g_3 $o_{3.2}$ (1-6) $o_{3.4}$ (1-6) $o_{3.5}$ (1-6) $o_{3.8}$ (1-6) $o_{3.10}$ (1-6)
g_4 $o_{4.1}$ (1-6) $o_{4.3}$ (1-6) $o_{4.6}$ (1-6) $o_{4.7}$ (1-6) $o_{4.9}$ (1-6)
g_{14} $o_{14.1}$ (1-6) $o_{14.3}$ (1-6) $o_{14.6}$ (1-6) $o_{14.7}$ (1-6) $o_{14.9}$ (1-6)

Figure 6.6: Representation of the r^{th} division of the database with $n = 3$. Light gray: Train set, dark gray: Test set

Table 6.1: Classification error (and its standard deviation) for different ensembles of Mclassifiers using the average fusion rule.

M	**LS-LDA tree**	**LS-LDA**	**MLP**	**RndF**	**K-NN**
1	**30.2**% ±4.0	35.8% ±3.5	32.9% ±5.7	31.1% ±3.1	36.8% ±2.9
2	22.3% ±3.4	26.9% ±3.7	23.6% ±6.7	**21.2**% ±3.2	29.6% ±3.1
3	16.9% ±2.8	22.1% ±4.0	19.8% ±6.1	**16.4**% ±3.6	24.1% ±3.4
4	13.9% ±2.7	19.3% ±3.3	16.8% ±6.9	**13.8**% ±4.1	21.0% ±3.9
5	**11.7**% ±2.7	17.2% ±3.0	14.9% ±6.8	12.1% ±4.0	19.1% ±3.9
6	**10.2**% ±2.6	15.7% ±3.4	13.3% ±7.1	10.9% ±4.3	17.3% ±4.1
7	**8.9**% ±2.6	14.3% ±3.6	12.2% ±7.0	10.2% ±4.6	14.9% ±4.1
8	**7.9**% ±2.5	13.5% ±3.4	11.3% ±7.3	9.6% ±4.7	13.8% ±4.3

established classifiers using the average fusion rule under equal conditions. Later, we compared the results obtained by the LS-LDA tree using different fusion rules. During the experiments, the exact same data set divisions were used for every classifier and fusion rule tested.

6.5.3 Experimental Results

Table 6.1 shows the results obtained for various classifiers using the average fusion rule. The tested classifiers are:

- LS-LDA tree: The classifier described in section 6.3.1
- LS-LDA: single stage LS-LDA. Trained using Matlab's *fitcdiscr* function.
- MLP: Multi-Layer Perceptron with 2 hidden layers (12 and 6 neurons) and 3 output neurons. Trained using Matlab's *patternnet* function.
- RndF: A random Forest formed by 128 decision trees. Trained using Matlab's *TreeBagger* function.

Table 6.2: Classification error (and its standard deviation) for the LS-LDA tree using different fusion rules.

M	**average**	**ML**	**weighted**	**maj. vote**	**supervised**
1	30.2% ±4.0	30.2% ±4.0	30.2% ±4.0	30.2% ±4.0	31.3% ±3.3
2	22.3% ±3.4	**20.6**% ±3.3	22.7% ±3.6	22.4% ±3.1	22.0% ±2.7
3	16.9% ±2.8	**15.0**% ±2.8	17.5% ±3.2	17.2% ±2.4	16.1% ±2.4
4	13.9% ±2.7	**11.6**% ±2.5	14.5% ±3.1	14.6% ±2.8	13.2% ±2.4
5	11.7% ±2.7	**9.5**% ±2.5	12.4% ±3.2	12.8% ±2.1	10.8% ±2.4
6	10.2% ±2.6	**8.1**% ±2.3	11.0% ±3.0	11.6% ±2.7	9.5% ±2.3
7	8.9% ±2.6	**7.0**% ±2.2	9.7% ±2.9	10.6% ±2.5	8.2% ±2.3
8	7.9% ±2.5	**5.9**% ±2.2	8.7% ±2.9	9.8% ±2.5	7.2% ±2.4

- K-NN: K-nearest neighbors using L1 norm and the 5 closest neighbors. Trained with Matlab's *fitcknn* function.

We are using probabilistic outputs with every classifier and unless otherwise stated, the employed functions and libraries are set to their default configuration.

The results show a similar error decrease for every classifier when more members are added to the classifier ensemble. This clearly points at spatial diversity as a major factor for achieving good accuracies, more relevant than the classifier itself. Nevertheless, among the tested classifiers, the LS-LDA tree obtained the best results for large ensembles while having a lower computational complexity than some of the other tested methods.

Table 6.2 shows the results obtained using the LS-LDA tree when different decision fusion rules are applied:

- Average: Average fusion described in (2.50).
- Maximum Likelihood: fusion rule proposed in (6.16).
- Weighted Average: weighted variation described in (2.51), with e_m computed from the train set using LOOCV in the same way as $\mathbf{C}_k$
- Majority Vote: D is assigned to the class label selected by the most nodes. Ties resolved by absolute maximum output.
- Supervised Average: the weights for each zone and class are obtained using LS-LDA during training. In this configuration $\mathbf{w}_k$ is a 3 element vector out of 4 possible ones. The weight vector of each zone is computed with $M = 8$, for every training iteration the outputs of the nodes that have selected one particular zone are averaged, in case none of the M nodes did, the observation is withdrawn from the training set.

Of all the tested methods, only the supervised average and the proposed rule were able to improve upon average fusion. The proposed ML fusion rule yields the best results, obtaining a similar error rate when 6 nodes are used to that of average fusion with 8 nodes. It is important to highlight that

Table 6.3: Confusion matrix of the proposed classifier with $M{=}1$. Columns: true class, Rows: predicted class.

	$\mathbf{H}_1$	$\mathbf{H}_2$	$\mathbf{H}_3$	$\mathbf{H}_4$	$\mathbf{H}_5$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_5$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$
Handgun	4.5%	5.2%	5.4%	5.7%	2.8%	1.2%	1.0%	1.8%	0.7%	0.9%	0.2%	0.9%	2.2%	0.6%
Rifle	1.2%	1.2%	1.5%	0.9%	3.3%	4.7%	5.5%	4.6%	5.4%	4.8%	1.5%	1.6%	1.0%	1.0%
Shotgun	1.4%	0.8%	0.3%	0.5%	1.0%	1.2%	0.6%	0.8%	1.1%	1.4%	5.5%	4.7%	4.0%	5.6%

Table 6.4: Confusion matrix of the proposed classifier using the ML fusion rule with $M{=}8$. Columns: true class, Rows: predicted class.

	$\mathbf{H}_1$	$\mathbf{H}_2$	$\mathbf{H}_3$	$\mathbf{H}_4$	$\mathbf{H}_5$	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$	$\mathbf{R}_5$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$
Handgun	6.8%	6.9%	7.1%	7.1%	4.0%	0.2%	0.1%	0.4%	0.1%	0.2%	0.0%	0.2%	0.7%	0.1%
Rifle	0.1%	0.1%	0.1%	0.1%	2.3%	7.0%	7.1%	6.8%	7.1%	6.9%	0.1%	0.1%	0.0%	0.1%
Shotgun	0.4%	0.2%	0.0%	0.0%	1.0%	0.1%	0.1%	0.1%	0.1%	0.1%	7.1%	7.0%	6.5%	7.0%

naive weighted fusion can be counterproductive, as shown in the table for one common weighted solution. Communication cost is fairly low regardless of the fusion method employed.

Assuming a network where the outputs are transmitted in single-precision (32 bit) floating-point format, the total amount of data required to transmit the outputs of 8 nodes is just 768 bits per evaluation (without considering protocol overhead). With a weighted scheme, since every node has to have every weight stored, the only transmission needed is an index (per node) to represent the selected weight. Encoding said index as a single byte, the total transmission would increase by 64 bits. Since the system works with framed audio, the required bit rate with the present configuration is 78000 bps, although there is no need to take a decision every frame. In terms of computational complexity, the proposed fusion rule is by far the most demanding, with a number of operations proportional to M^2 while for every other method it is proportional to M. However ML fusion can also be made proportional to M by using a tabulated implementation as suggested in section 6.4.1.

The error rates obtained with the LS-LDA tree for range and alignment estimations are 3.5% and 5.8% respectively, which is in clear contrast with those obtained for weapon classification and it is thanks to this degree of accuracy that the spatial division is able to boost the accuracy of the system.

The obtained results show a strong relationship between the spatial resolution of the classifiers and the obtained error. From them, it is clear that the addition of spatial diversity to the system is of the utmost importance. With the LS-LDA tree, just by adding a second node in conjunction with the proposed fusion method, the classification error falls by almost 10% getting as low as 5.9% when 8 nodes are used. It is worth remembering that the classifiers were trained without using any observations of the tested guns neither the tested locations, so it is safe to say that we are working in the most restrictive conditions that the current database allows. The proposed methodology can be exploited with a larger number of weapon classes. For as long as a given weapon class encompasses weapons with similar and characteristic features, such as barrel length, caliber, propeller load, projectile velocity, etc, it would be possible to differentiate those weapons as belonging to a particular group.

Finally, Table 6.3 shows the confusion matrix (breakdown by weapon) obtained with the proposed classifier using the average fusion rule and M =1, and Table 6.4 shows that obtained with the proposed ML fusion rule and M =8. These results show that the benefit of a multichannel setup for gunshot classification is fairly consistent for all tested weapons. In almost every case, the classification error with the largest ensemble size is reduced significantly in comparison with the single channel. The weapons that cause the largest amount of errors are H$_5$, which coincidentally is the largest caliber handgun in the database, and S$_3$ which is the shotgun with the shorter barrel length, thus, a larger error rate can be explained as a divergence from the common characteristics within their class.

6.6 Known Spatial Relations

After tackling multi-observation classification on a network in which array processing is not feasible due to technical limitations, at this point we are going to assume that the various functionalities described in the previous chapters of this thesis have been successfully implemented. This way, we can consider that both node and source locations are known to a reasonable degree, that the network is properly synchronized, and that it is possible to obtain the local ToAs of an acoustic event (i.e., local detections). In this scenario, it is not longer advisable to have the nodes estimate the relative location of the source since it is already known. Thus, the focus will be how to take advantage of previously obtained information in order to boost the classification.

Since we are considering that the nodes are already implementing various algorithms (at least detection and source localization), and one of the objectives of this thesis is efficiency; the lower the computational complexity of the classification the better for our goal. Thus, in this second scenario we have decided to maintain the features used for the detection system, as to not to excessively overload the nodes. Therefore, the classification system must work with already available information and the fusion scheme is again decision fusion.

6.6.1 Multichannel Classification

The simplest approach to multichannel classification is to combine the decision taken by the whole ensemble at the same time instant t. Considering a network composed of M nodes, where some global detection instant is Γ and each classifier provides C probabilistic outputs $y_{mc}(t)$, the final decision taken by the ensemble is:

$$D = \arg\max_{c} \Big(\sum_{m=1}^{M} y_{mc}(\Gamma) \Big). \tag{6.19}$$

Given the short time duration of impulsive acoustic events this method of fusing decisions can prove problematic for classification. If we were to combine the outputs of an ensemble of classifiers at the same time instant, this would imply that by the time some of the nodes receive the transient of the acoustic event, it is very likely for some other nodes to already be past that point, so that they are receiving reverberations. This situation is far from ideal (specially when short time frames are used) given that the most reliable source of information about the event's source is the direct signal (i.e., the noise transient).

The problem can be lessen by setting a time window for the classification in a similar way to that described on section 4.4.1. The time offset between channels (i.e., TDoA) is not much of a problem for detection since the time duration of the target acoustic events is very likely to be larger than the TDoA between channels, thus local detectors are expected to remain active for a long enough period of time. However, impulsive acoustic event classification is a harder problem than detection, given that the differences between impulsive sources are far smaller than that between impulsive noises and most other types of sound. Therefore, the output of a classifier is less likely to remain 'steady' for a long enough period of time to this direct approach to work. Thus, this time around, it is not advised to combine time instants inside the window together, being preferable instead to have every node compute an auxiliary output (i.e., time-domain decision fusion). Let T_C be a temporal window (in frames) sot that every member of the ensemble obtains a single output for each class as:

$$z_{mc} = \mathcal{F}\big(y_{mc}(\ell)\big), \text{ with: } \ell = \{l \in \mathbb{Z} : \Gamma \leq l < \Gamma + T_C\}, \tag{6.20}$$

where z_{mc} is a representative score (e.g., average value, argument of the maximum, etc) for a given class inside the time window. The final decision then becomes:

$$D = \arg\max_{c} \Big(\sum_{m=1}^{M} \mathcal{F}\left(z_{mc}\right) \Big). \tag{6.21}$$

Note that this scheme is most effective when the signals are being framed with a short hop time, since the spectral content of impulsive events significantly changes between consecutive frames, worsening the effect of having misaligned classifier outputs.

If we consider that every node in the network is capable of obtaining the ToA of a target acoustic event (i.e., detection onset Γ_m), then individual classifier outputs can be aligned according to their ToAs so that the the time offset between channels gets bypassed. This way it is possible to go back to expression (6.19) and combine the outputs of the ensemble at the time instants corresponding to their local ToA. However, in this particular case, we have decided to use the same features employed for detection, and unfortunately, a single frame lacks the information needed for accurate classification. Note that in this case the feature vector includes temporal context, hence, the frame corresponding to the ToA of the acoustic event is not the only frame that contains information about its transient, being possible for some later output $y_{mc}(\Gamma_m + t)$ to be the most reliable for classification.

Therefore, we propose to use a detection-aligned decision fusion scheme in which every node contributes a single value per class computed from the classifier output over a fixed time window. Defining this time window as $\ell_m = \{l \in \mathbb{Z} : \Gamma_m \leq l < \Gamma_m + T_C\}$, then z_{mc} is the minimum value of the classifier output for the c^{th} class during the predefined T_C frames an the final decision obtained with proposed decision rule is:

$$D = \arg\max_{c} \Big(\sum_{m=1}^{M} \arg\min_{\ell_m} \big(y_{mc}(l)\big) \Big). \tag{6.22}$$

Choosing the minimum value over the time window may seem counterintuitive. However, it is worth considering that since the classifier is trained using class labels that encompass a number of frames, the true class is the most likely among all classes to remain active during the whole time window. Additionally, the likelihood of a FP in a single frame is much higher than over the whole window, thus, computing z_{mc} as the mean or the maximum value of each class score during the classification window is more prone to be affected by spurious outputs. Thus, by computing z_{mc} as the minimum value of each class output during the time window we are setting a tradeoff between the ability of the classifier to recognize the class of some event during T_C frames and the likelihood of having a FP for the same amount of time.

6.6.2 Weighted Decision Fusion

We are tackling multi-observation classification of impulsive acoustic events in a spatially diverse scenario, therefore, it is very likely for some of the classifiers to perform better than others. The further away the source is from a microphone, the lower the SNR of the received signal, and the higher the influence of the propagation path. Thus, without additional considerations (e.g., faulty sensors, presence of undesired sources, etc), distance from a given microphone to the source and SNR are good indicatives of the quality of the signal to be classified.

At this point we can consider that node locations ($\mathbf{p}_m$) are known, and that the location of the source ($\mathbf{p}_s$) has been properly estimated, so that every node can simply compute its distance to the source as a L^2-norm: $d_m = ||\mathbf{p}_s - \mathbf{p}_m||$. Then, it is possible to weight the contribution of each classifier to the decision fusion according to its distance to the source (similarly to [DH04]). The standard inverse square law for point sources specifies that the received sound pressure is inversely proportional to the square of the distance from the receiver to the source, thus we propose to use a weighted decision fusion scheme derived from it. Taking expression 6.22 and introducing a weight $w_m = 1/{d_m}^2$ to the contribution of every classifier in the ensemble, we can derive a simple distance-based weighted decision fusion rule as:

$$D = \arg\max_{c} \Big(\sum_{m=1}^{M} w_m \arg\min_{\ell_m} \big(y_{mc}(\ell) \big) \Big). \tag{6.23}$$

Other than d_m, the SNR of the signals can also be used to weight the decisions. SNR is defined as the ratio of the power of the desired signal and the power of background noise. Since we do not know neither of these elements we have to work with an approximation. Let us consider the periodogram-based power spectral estimate of an audio stream as $S(k,t)$, $k = 1,\dots,L$, where k is the frequency bin index, and t is the frame index. If we assume that every node is capable of estimating the ToA of some acoustic event as Γ_m, then its instantaneous SNR can be approximated with the following expression:

$$\mathrm{SNR}_m = \frac{\sum_{k=1}^{L} S(k,\Gamma_m) - \sum_{k=1}^{L} S(k,\Gamma_m - 1)}{\sum_{k=1}^{L} S(k,\Gamma_m)}. \tag{6.24}$$

For the sake of efficiency, we propose to obtain the SNR approximation from the Mel log-band energies instead of the periodogram. Let vector $\mathbf{x}_{mt}$ of length F be the Mel log-band energies computed from the t^{th} frame recorded by the m^{th} microphone, and let T_W be a parameter that controls the number of frames that are to be considered. Then the weight to be applied to the m^{th} classifier can be computed as:

$$w'_m = \frac{\sum_{t=\Gamma_m}^{\Gamma_m+T_W-1} \sum_{j=1}^{F} x_{mtj} - \sum_{t=\Gamma_m-T_W}^{\Gamma_m-1} \sum_{j=1}^{F} x_{mtj}}{\sum_{t=\Gamma_m-T_W}^{\Gamma_m-1} \sum_{j=1}^{F} x_{mtj}}, \tag{6.25}$$

Thus, replacing w_m for w'_m in expression (6.23) yields a SNR-based fusion rule. Note that T_W is included in (6.25) to average the energy of various frames before and after Γ_m in case a wider time window is wanted.

Finally, since the classification is very likely to be affected both by SNR and microphone to source distance, we propose a hybrid weighed decision fusion rule that takes these these two factors into consideration. In a hypothetical scenario, some microphone may be close to the target source but also affected by some undesired noise (i.e., wind) that could make its decisions unreliable, thus, combining both proposed weighted schemes may prove beneficial. The weights for this 'dual' rule are obtained simply by combining normalized versions of w_m and w'_m, as expressed by:

$$w''_m = \frac{w_m}{\sum_{m=1}^{M} w_m} \frac{w'_m}{\sum_{m=1}^{M} w'_m}. \tag{6.26}$$

Again, plugging w''_m into expression (6.23) yields the proposed decision fusion rule. Note that there is no need to normalize w_m or w'_m when used independently because the argument of the maximum function is not affected when all the elements are scaled by the same factor, and also that, the normalization is used here just to equalize the contribution of each component.

6.7 Experimental Work and Results: Known Geometry

This section describes the experiments conducted to evaluate the performance of the proposed multichannel impulsive sound event classification scheme as well as a discussion of the obtained results. The evaluation was done using multichannel recordings of real impulsive acoustic events generated with 5 different sources on a spatially diverse scenario.

6.7.1 Database Description

Evaluation of impulsive sound event classification with a known network geometry was done with the same database used in chapters 4 and 5. As previously mentioned, the database is composed of 12 channel recordings of 5 impulsive acoustic sources triggered at 16 unique locations. The recorded impulsive sounds are: Firecracker explosion, Balloon pop, Compressed air gun shot, Drum hit and Handclap. Recording took place in two sessions in which microphone locations and source locations where maintained, adding up to a total of 160 acoustic events.

Database labeling was done manually. The labels describe the class, 'onset' time and 'offset' time of every acoustic event. In contrast to chapter 4, where the 5 class labels where collapsed into a single 'event activity' label, this time all 5 labels are used as targets for the classification system. Figure 6.7 shows the average normalized Mel log-band energies of all the events (selected from

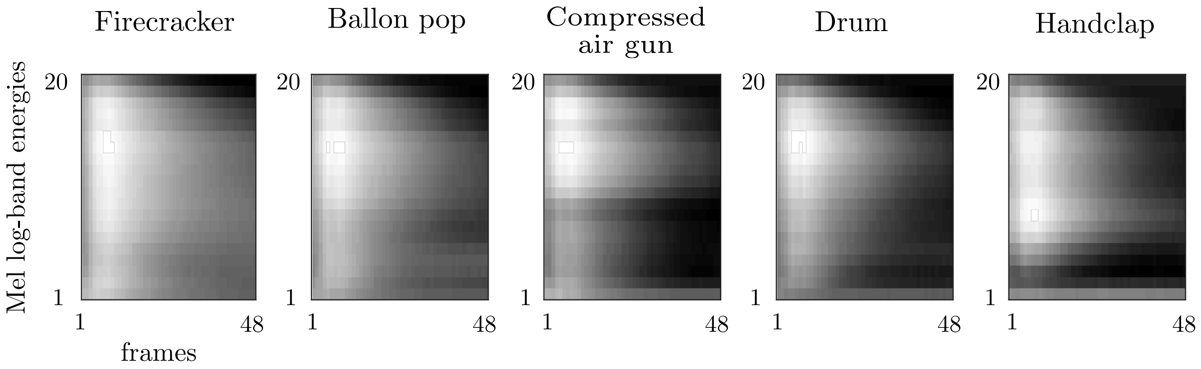


Figure 6.7: Average normalized Mel log-band energies of the 5 classes included in the database with 11.6 ms frames and 5.8 ms hop length.

their onset labels) belonging to each of the 5 classes included in the database with 11.6 ms frames and 5.8 ms hop length. For the figure it is possible to see that differences between the 5 sources are mostly present in the lower half of the spectrum, which is somewhat representative of the 'loudness' of each source.

Regarding the type of impulsive transients of each source: a firecracker explosion is a blast wave, a balloon pooping is a combination of a blast wave and a shock wave[2], the sound of a compressed air gun is a combination of a blast wave and impact noise (i.e. mechanically produced noise), and finally a drum hit and a handclap are purely impact noise.

6.7.2 Description of the Experiments

The features used for impulsive sound classification are the same used for detection that are described on section 4.5.2. The feature vector is composed of 20 log Mel-band energies plus 9-frame compressed context, resulting in $F = 60$. Hop time between consecutive frames is 256 samples, or approximately 5.8 ms. Note that, weights for the SNR-based approach described in section 6.6.2 are obtained without considering the temporal context.

Training is also very similar to that of chapter 4. We consider that for any given event, both the source location and the receiving microphone are very likely to be unknown to the system (i.e., not included within the train samples), thus, the presented results were obtained using LOOCV over microphone source location combinations. In total there are 192 such combinations with 2 events of each class per combination. At every iteration of the LOOCV, the database is divided into a test set that contains the signal captured by the microphone under test (10 events), during the tested source location time window; and a train set that contains most of the combinations: the signals captured with the remaining 11 microphones, during the remaining 15 source location time windows. Then, the results for each microphone are obtained by selecting a number of output frames from the 16 source-location specific classifiers (32 events per class in total). Once again, by using this training scheme, we try to assess how the results of the detection would generalize to a larger dataset, and since each classifier is trained using a specific train set designed to isolate it, the results obtained

[2] The sound of a balloon pop is believed to be a combination of the release of internal pressure and a shock wave produced by the contracting rubber breaking the sound barrier.

Table 6.5: Error rate obtained by various classifiers for different decision fusion rules an different number of microphones (M).

	***K*-LDA** $(K = 6)$		**MLP** $(N_L = [20, 20, 5])$		**RndF** $(N_T = 16)$	
M	local-min	global-mean	local-min	global-mean	local-min	global-mean
1	23.9% ±6.1	23.5% ±5.2	**16.2**% ±5.2	18.9% ±5.8	21.0% ±6.7	17.0% ±6.0
2	18.7% ±5.6	19.8% ±5.0	11.5% ±6.1	13.0% ±5.9	12.3% ±5.9	**11.2**% ±6.2
3	15.6% ±6.2	18.0% ±5.6	10.3% ±6.8	11.3% ±6.1	9.7% ±5.7	**9.4**% ±6.6
4	13.9% ±6.4	17.4% ±5.9	9.8% ±7.1	10.7% ±6.4	**8.7**% ±5.7	9.0% ±6.9
5	12.5% ±6.8	16.4% ±6.2	9.8% ±7.3	10.5% ±6.5	**8.2**% ±5.8	8.7% ±7.1
6	11.6% ±7.0	16.1% ±6.4	9.5% ±7.6	10.1% ±6.7	**7.7**% ±5.8	8.5% ±7.3
7	11.0% ±7.1	15.8% ±6.6	9.3% ±7.9	9.8% ±6.9	**7.3**% ±5.8	8.3% ±7.4
8	10.4% ±7.2	15.6% ±7.3	9.1% ±8.1	9.5% ±7.1	**7.1**% ±5.8	8.2% ±7.6
9	10.1% ±7.4	15.5% ±7.8	9.0% ±8.3	9.3% ±7.3	**6.9**% ±5.9	8.2% ±7.7
10	9.5% ±7.4	15.5% ±8.3	8.9% ±8.4	9.3% ±7.5	**6.7**% ±5.9	8.1% ±7.7
11	8.8% ±7.2	14.9% ±9.4	8.7% ±8.6	9.4% ±7.8	**6.6**% ±5.8	8.1% ±7.7
12	8.6% ±7.2	15.4% ±9.1	8.9% ±8.7	9.5% ±8.4	**6.5**% ±6.5	8.1% ±7.8

on a multichannel setup can be considered independent to one another, therefore, it is fair to use decision fusion to obtain multichannel detection results.

The results were obtained under the assumption of perfect detection, hence, event onset labels were used to select the frames used for the classification. Multichannel results are obtained by averaging all the possible combinations of $M = 1, \ldots, 12$ channels. Multichannel classification is implemented as described on section 6.6.1 with a detection window $T_C = 15$ samples. In contrast to detection, this time around there are no negative observations, that is, every observation belongs to one of 5 target classes. The evaluation metric is again error rate.

For the experiments, first we compared the performance of the K-LDA classifier proposed on chapter 4 and two well established classifiers using two different temporal fusion schemes. The classifiers are a RndF using $N_T = 16$ trees, and a fully connected feedforward ANN, or MLP, with a number of neurons per layer $N_L = [20, 20, 5]$, trained for 300 epochs, with 0.5 parameter regularization, and assigning an error weight $E_w = [1, 100]$ to the train set samples in order to counter class imbalance. Temporal fusion on each channel was performed using the methods described in expressions (6.21) and (6.22). We also compared the performance of various weighted decision fusion schemes with the K-LDA classifier. During the experiments, the exact same data set divisions were used for every classifier tested.

6.7.3 Experimental Results

Table 6.5 shows the error rate obtained by various classifiers for different decision fusion rules an different number of microphones (M). Note that, “local-min” refers to the proposed detection-aligned decision fusion scheme in which every node contributes the minimum value of each of

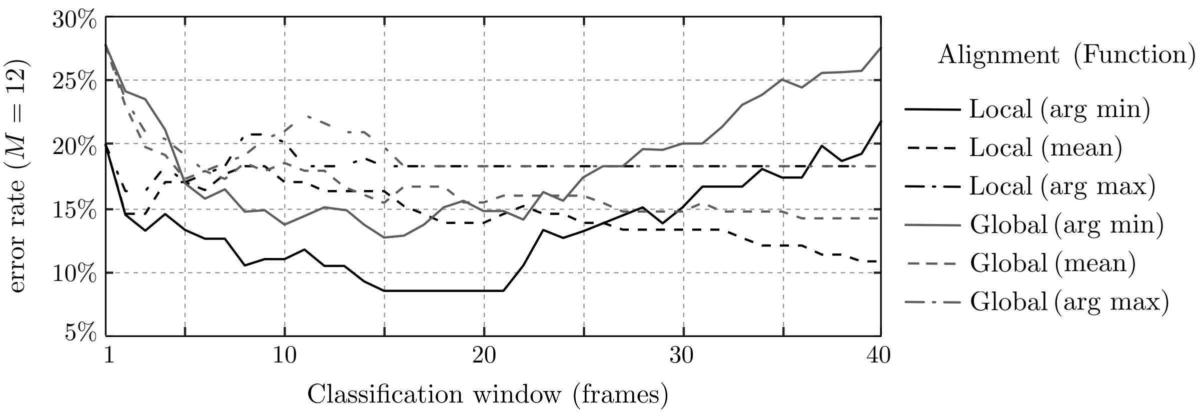


Figure 6.8: Error rate of various time fusion methods as a function of the classification window (T_C) with the K-LDA classifier (K=6) and $M = 12$.

its outputs during the classification window, while "global-mean" refers to a globally-aligned (i.e., same time base) fusion scheme in which every node contributes the mean value of each output during the classification window. For every ensemble size grater than one, best performance is obtained by the RndF. The MLP is the best with a single microphone but it does not obtains the same 'boost' as the other tested classifiers for larger ensembles sizes. In almost every case, the proposed decision fusion scheme outperforms the globally-aligned scheme, except for the RndF with M $\leq$3 where the latter works better. In general the combination of multiple channels via decision fusion proves as a good strategy for impulsive sound event classification. The results with the proposed fusion method with $M = 12$ are around 2 to 3 times better than that with a single classifier, which demonstrates the importance of spatial diversity for this particular task. The accuracy of the K-LDA classifier falls behind the other two tested classifiers, however, is worth remembering that even with a larger number of classes it is still the least computationally expensive algorithm of the three. It is worthwhile to mention that results of simpler classifiers have been dismissed due to a large difference in performance. Just for illustrative purposes, a regular LS-LDA obtained an error rate 38% with all 12 microphones, while the K-LDA with 2 clusters obtained 31%.

Results shown on Table 6.5 were obtained with a classification window length $T_C = 15$ samples. Regarding this decision, due to the outputs of the classifiers being processed before the decision fusion stage, the length of the window (together with its starting instant) and the function used to select the representative value of each output both have a notable effect on the classification accuracy. Figure 6.8 shows the error rate of and ensemble of 12 K-LDA classifiers using different decision fusion schemes. From the figure it is clear that for most classification window lengths, the proposed scheme (local detection alignment plus minimum value) is the most reliable for most values of T_C (i.e., until the window length is larger than the acoustic events). It is interesting that the minimum value is also the best function for time fusion with globally aligned outputs, thus, this points out at misclassifications on a few spurious frames being the most relevant problem. This claim is also justified by the bad performance of the maximum value with either alignment. Considering these results, the value of T_C was chosen as the shorter window length offering the best performance. Also these results prove that, at least under the tested conditions, performing decision fusion with locally aligned signals is superior to doing so with signals sharing a common timebase.

Table 6.6 shows the error rate obtained with the K-LDA classifier (K=6) for various decision

Table 6.6: Error rate of the proposed K-LDA classifier (K=6) for various decision fusion rules an different number of microphones (M).

M	**Average**	**Distance**	**SNR**	**Dual**	**ML**
1	23.9% ±6.1	23.9% ±6.1	23.9% ±6.1	23.9% ±6.1	23.9% ±6.1
2	18.7% ±5.6	17.9% ±5.8	16.3% ±6.2	**15.7**% ±6.2	18.2% ±5.7
3	15.6% ±6.2	14.6% ±6.1	12.8% ±6.8	**12.3**% ±6.8	15.2% ±6.4
4	13.9% ±6.4	12.6% ±6.4	11.6% ±6.6	**10.8**% ±6.7	13.6% ±6.6
5	12.5% ±6.8	11.1% ±6.6	10.3% ±6.8	**9.4**% ±7.1	12.2% ±6.9
6	11.6% ±7.0	10.2% ±6.8	9.7% ±7.2	**8.9**% ±7.4	11.4% ±7.3
7	11.0% ±7.1	9.4% ±6.9	9.2% ±7.2	**8.6**% ±7.5	10.8% ±7.2
8	10.4% ±7.2	8.9% ±7.2	8.9% ±7.2	**8.4**% ±7.5	10.4% ±7.4
9	10.1% ±7.4	**8.4**% ±7.4	8.7% ±7.5	**8.4**% ±7.7	9.9% ±7.7
10	9.5% ±7.4	**8.0**% ±7.1	8.4% ±7.7	8.3% ±7.8	9.3% ±7.7
11	8.8% ±7.2	**7.6**% ±6.8	8.1% ±8.2	8.1% ±8.3	8.4% ±7.4
12	8.6% ±7.2	**7.3**% ±6.8	8.3% ±7.2	8.2% ±7.0	8.6% ±7.2

fusion rules an different number of microphones (M). Note that ‘distance’, ‘SNR’ and ‘dual’ correspond to the weighting schemes described by expressions (6.23),(6.24) and (6.26) respectively, while ‘ML’ is the decision rule proposed in section 6.4. In order to use the ML rule, source to microphone distance was divided in 4 tiers, covariance matrices for every source location-microphone combination were computed excluding the results of the microphone under test as described in section 6.5.2. The distances used for both distance-based methods were obtained using the true source and microphone locations.

The results presented on Table 6.6 show that the best decision fusion rule for most ensemble sizes is the dual weight rule, while for large ensembles $M \geq 9$ distance-based decision fusion is more accurate. The SNR-based rule obtains a notable ER decrease for smaller ensemble sizes but stalls for larger ensemble sizes, while the distance-based rule offers a more even improvement for any ensemble size. As it should be expected, the accuracy of the proposed dual weighted rule is somewhat a combination of the improvement obtained by both rules independently, although it seems like the SNR-based weights become more relevant for large ensemble sizes. The ML fusion rule fails to obtain the same improvement as the simpler weighted decision fusion rules, most likely due to the dissimilarities of the events generated at different locations not being large enough for the classifiers of each zone to be considered as different. Additionally, in this case, segmentation was done based on distance to the source, and given that source to microphone distance differences are less pronounced than in the previous case (gunshot classification), it is possible that discrete segmentation is not the best approach. In any case all of the proposed weighted schemes are able to improve the results of standard average decision fusion.

Weighted decision fusion can also be applied to any other classifier. Figure 6.9 shows the results obtained with the 3 tested classifiers using the average fusion rule and the proposed dual weighted fusion rule. The weighted scheme outperforms the standard one in every case except for the RndF with M >10. Note that error rate decrease is related to the performance of the unweighted classifi-

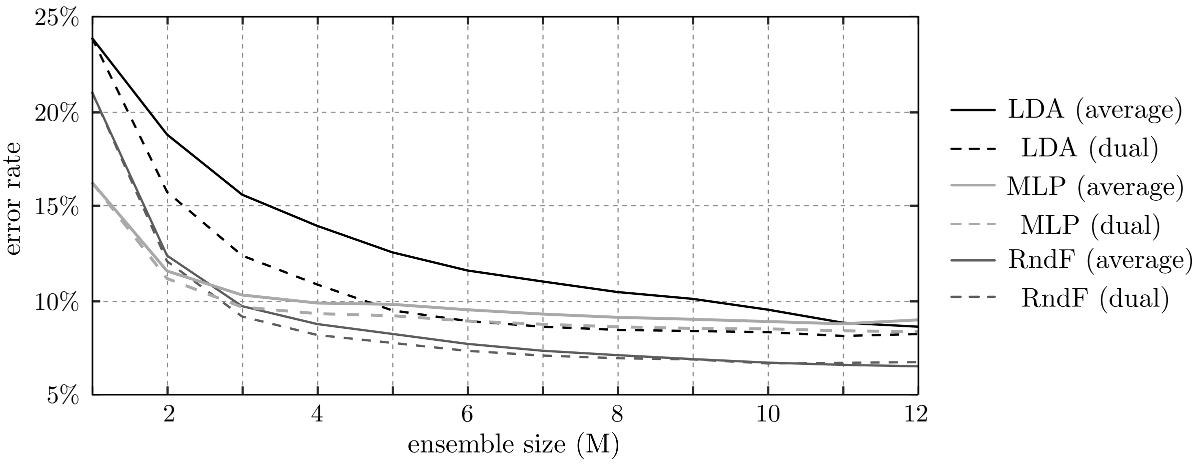


Figure 6.9: Error rate of various classifiers for an increasing ensemble size with two fusion rules

Table 6.7: Confusion matrix of the K-LDA classifier (K=6) with M=1. Columns: true class, Rows: predicted class.

	Firecracker	**Balloon**	**Air gun**	**Drum**	**Handclap**
Firecracker	18.9% ±1.5	3.9% ±2.0	0.6% ±0.7	2.1% ±1.1	0.9% ±0.8
Balloon	0.4% ±0.5	7.7% ±4.0	0.7% ±0.7	1.4% ±1.4	0.1% ±0.3
Air gun	0.3% ±0.3	3.9% ±3.1	16.8% ±1.8	0.6% ±0.6	0.1% ±0.3
Drum	0.2% ±0.2	3.4% ±2.7	1.5% ±2.1	13.8% ±2.0	1.1% ±1.8
Handclap	0.2% ±0.2	0.8% ±1.1	0.3% ±0.4	2.6% ±2.7	17.9% ±4.3

cation, being notably larger for the K-LDA, so far as outperforming the MLP for M >6. The benefit of using a weighted decision fusion scheme holds for any tested classifier. Although not shown in the results, the error rate of a regular LS-LDA and a K-LDA with 2 clusters is reduced by approximately 3% and 7% respectively when the the dual weighted fusion rule is introduced.

Overall, the presented results show that spatial diversity obtained from an ensemble of classifiers working with multiple observation of an acoustic event is a valuable tool for impulsive source classification, and that the performance of a multichannel classification system can be further increased by using suitable weighted decision schemes. Additionally, the proposed system shows that it is advisable to take advantage of previous information about the acoustic events to formulate novel approaches to impulsive sound event classification.

Finally, Tables 6.7 and 6.8 show the confusion matrices obtained with the K-LDA classifier (K=6) with M =1, and with M =9 using the proposed dual weighted fusion rule. The results clearly show that some of the classes are better identified than others, specially 'fireckacker explosion' and 'air gun gunshot', which are perfectly classified with M =9. The most problematic class is 'balloon pop'; even with a large ensemble it still produces a large amount of misclassifications (almost 40% of the events). In contrast to this, 'drum hits' also produce a large amount of misclassifications with M =1 but performance is significantly increased with M =9 (note the difference in 'air gun-drum hit' confusions). This is most likely due to balloons producing a very short noise transient when popped, making it harder to correctly classify. It is worthwhile mentioning that balloons are often used for

Table 6.8: Confusion matrix of the K-LDA classifier (K=6) using the proposed dual weighted fusion rule with M=9. Columns: true class, Rows: predicted class.

	Firecracker	**Balloon**	**Air gun**	**Drum**	**Handclap**
Firecracker	20.0% ±0.4	1.9% ±3.4	0.0% ±0.0	0.0% ±0.0	0.2% ±0.9
Balloon	0.0% ±0.0	12.4% ±6.2	0.0% ±0.0	0.2% ±0.1	0.0% ±0.0
Air gun	0.0% ±0.0	2.6% ±3.9	20.0% ±0.3	0.0% ±0.0	0.0% ±0.0
Drum	0.0% ±0.0	2.7% ±3.9	0.0% ±0.0	18.9% ±2.0	0.1% ±0.6
Handclap	0.0% ±0.0	0.5% ±1.9	0.0% ±0.0	0.6% ±1.2	19.7% ±4.0

impulse response measurements [PKL11]. These results show again that, in general, multichannel classification based on decision fusion is a suitable strategy for impulsive sound event classification.

6.8 Conclussions

In this chapter two efficient multichannel classification schemes for impulsive sound source classification that take advantage of spatial information have been proposed. The first method is intended for networks lacking array processing capabilities, thus, nodes have to estimate their spatial relation towards the source in order to lessen the impact of spatial uncertainty. The second methods takes full advantage of some of the algorithms described throughout this thesis, therefore, every node knows its own location as well as the source's location, and is capable of estimating the ToA of the acoustic event.

The main novelty resides in the use of spatial information to weight the decision taken by every classifier in the ensemble based on different strategies. Regardless of the type of impulsive source or the methodology used to exploit spatial diversity, the presented results show that weighted decision fusion is a valuable tool for impulsive sound event classification on WASNs. The proposed multichannel classification approaches are simple to implement, do not imply a significant computational complexity increase and data transmission requirements are fairly low. Since the proposed classification methods rely on previous acoustic event detection, the whole system can be viewed as a hierarchical multi-class detection system.

When the network is assumed not to be able to estimate the location of the source, spatial diversity is exploited with a novel ML decision fusion rule that considers diversity in the classifier ensemble. Classifier diversity comes from a modification of the K-LDA classifier proposed in chapter 4, in the form of a classification tree that uses the outcome of two binary classifiers to segment space and select one of various spatially-specialized weapon classifiers. Then, during the decision fusion, differences in performance between these specialized classifiers are exploited to further 'boost' the system.

In the second scenario, spatial information is assumed to be known, thus, node to source distance can be directly used to weight the decisions taken by the ensemble. Additionally, a combination of distance-based and SNR-based weights is proposed. Adaptation of the ML rule to this scenario failed to achieve the same performance as simpler weighted methods, thus, further research is needed in order to better capture how the accuracy of the classifiers changes in relation to the relative location of the source.

It is important to remember that most work on impulsive ‘noise’ detection and classification does not take the spatial aspect of the problem into consideration, with many proposals being developed around examples extracted from sound libraries that are intended as audio effects and do not resemble real recordings of the impulsive sources they portrait. Coincidentally, the sounds used for gunshot classification were indeed obtained from a sound library, however, in this particular case, it is intended to serve as a source for gunshot sound design. The vendor markets this sound library as a source of raw recordings to be processed and mixed together in order to provide a wide range of possibilities when designing original sound effects. Sounds used in the second part of this chapter come from the same database used on chapters 4 and 5. The database was purposely recorded for the experiments described in this thesis, hence, we can ensure that the sound events contained on it come from raw multichannel recordings.

Likewise in previous chapters, the possibility of having transmission errors or faulty sensors was not considered. Additionally, for the most part, target acoustic events, have high SNR and direct line of sight to the sensors. Therefore, future work should address less favorable conditions and consider a wider spatial diversity, which could in part be approached by recording a more extensive database. Muzzle blast directivity was a major point in the first part of this chapter and was not considered on the second part since every source, excluding the compressed air gun, can be considered close to omnidirectional. Then, a larger database should include not only more spatial diversity but also more types of sources with different characteristics such as glass breaking, various explosive sources, collisions, proper gunshots, etc. Some of this sources are more relevant than other for automated acoustic surveillance, thus future work will be focused on combining detection and classification as to produce a robust multi-class detection system capable of dealing with multiple types of sources.

6.9 Summary

The main contributions described in the present chapter of the thesis are the following:

1. An efficient classification algorithm based on the previously proposed K-LDA classifier, that substitutes the clustering stage for a spatial estimation stage and is capable of obtaining a competitive performance with limited computational complexity.
2. A ML decision fusion rule that considers diversity in the classifier ensemble.
3. An efficient solution to the data fusion problem for an ensemble of classifiers that uses local detections to select the outputs employed for the multichannel fusion.
4. Various weighted decision fusion methods that take advantage of different sources of information to improve the accuracy of the classification.
5. A multichannel impulsive sound event classification scheme capable of obtaining an adequate performance on a spatially diverse scenario, suitable for a WASN with restricted technical capabilities.

The contributions described in the first half of this chapter have originated publication [P2] and those described in the second half are being prepared for submission to a peer reviewed journal [P6]. Further details can be found in section: Publications.

CHAPTER 7 Conclusions

7.1 Introduction

This chapter is divided in two parts. The first part summarizes the main contributions of the thesis and the results presented throughout it. The second part offers a brief discussion about possible future research derived from the presented work.

7.2 Summary of Conclusions

This thesis has presented a number of novel multichannel algorithms for impulsive sound analysis in WASNs that address four distinct but not necessarily independent problems, namely: acoustic event detection and classification, sound source localization and node self-localization.

This section serves as a recapitulation of the contributions derived from the research carried out to fulfill each of the goals of this thesis. Each successive subsection bellow is related to one of the goals of the thesis and offers a summary of contributions and some conclusions.

7.2.1 Efficient Self-localization and Synchronization

Accurate knowledge about the locations of the microphones that form an array is a critical step for most spatial signal processing algorithms. Unfortunately, the process of determining the coordinates of multiple microphones manually is tedious and prone to human error. Therefore, self-localization has been an active research topic for a good number of years. Recently, focus has shifted from traditional microphone array calibration to ad hoc acoustic sensor networks, mainly due to the high availability and low price of portable devices that can be used to implement a WASN.

Acoustic-based self-localization methods work by combining one or more time measures taken by every microphone (or by pairs of microphones) using some optimization algorithm, the objective of which is to find the most probable node location given a set of observations (measurements). The computational load associated with the solution of this problem is usually significative, although, the complexity of the problem can have large differences depending on the particular set of conditions set by a given method. That is, what kind of measurements are available and how many unknowns have to be accounted for. In any case, most self-localization methods are very sensitive to outliers, hence, without a proper strategy to asses the quality of the measurements, the convergence of the estimation can fail due to a few large errors.

In chapter 3 we presented a closed-form active self-localization method. The algorithm assumes that the nodes are equipped with at least two microphones and one speaker. The entire localization process is based on the combination of DoA and range estimates taken between node pairs and obtained from acoustic signals without requiring prior synchronization of the nodes. The main contribution is the ML estimator of node locations that works by treating pairwise angle and range estimates as polar coordinates pointing towards the possible relative location of the node.

The proposed algorithm is capable of solving the self-localization problem even without prior knowledge about the orientation of the nodes and in the presence of DoA ambiguity. However, the presented solution to the DoA ambiguity problem takes away the benefit of having a closed-form self-localization method. Therefore, more efficient solutions that exploit the relations between the DoA estimates of various nodes should be explored. Although it is worth mentioning that this is only relevant for the implementation of the proposed method with readily available devices equipped with two microphones. The best solution, considering ad hoc hardware, is to equip the nodes with 3 or more microphones, which would maximize efficiency by eliminating DoA ambiguity and also increase DoA resolution. Another problem of the proposed method that can be effortlessly eliminated using ad hoc hardware is the error introduced by the speaker not being located at the center of the array.

The presented results show that the localization error is largely dependent on cross-correlations. All tested algorithms obtained a mean localization error between 10 cm and 20 cm with the longest cross-correlations. Localization errors were almost doubled in most cases with the shortest cross-correlations due to erroneous ToA estimates. This clearly points out at time lag measurement being a more relevant matter than the self-localization algorithm in terms of error.

In brief, the performance of the proposed algorithm is on the same scale as (or better than) other self-localization algorithms without requiring ad-hoc hardware or synchronization between nodes, while having a computational complexity that is assumable for current mobile processors (specially without DoA ambiguity).

A byproduct of the proposed self-localization method is the obtention of synchronization within the network. With a simple modification, the method used for obtaining the range estimates can be used to estimate the clock offset between the nodes. Although, the obtained synchronization is not without problems (most notably clock drift), thus, the use of more robust RF-based methods is advised.

7.2.2 Multichannel Impulsive Acoustic Event Detection

The most prevalent applications of WASNs in the literature are related to automated acoustic surveillance and almost every such application uses acoustic event detection in some form or another. This type of acoustic detection systems are typically focused on one or more hazardous acoustic events such as gunshots, explosions, and/or other potential distress indicators such as human screams.

In general, the kind of acoustic events that automated audio surveillance systems are set to detect, can be catalogued as "rare" acoustic events, which implies that every node in a WASN-based detection system needs to process their input audio stream continuously as to maximize probability of detection. Most often than not, detection systems are based on supervised machine learning techniques. It is important to mention that most previous work on impulsive acoustic event detection, does not take into consideration the spatial aspect of the problem, being commonplace to replace real recordings of target acoustic events with sound effects extracted from sound libraries.

In chapter 4 we presented an efficient multichannel acoustic event detection scheme that takes advantage of spatial diversity via decision fusion. The proposal is rooted on a novel classification algorithm based on LS-LDA that manages to improve the results of regular LS-LDA by taking advantage of K-means clustering to segment the feature space so that a specialized classifier can be trained for each of K clusters. Additionally, a ‘compressed’ feature temporal context scheme was proposed, intended to increase the temporal information available to the classifier without excessively increasing the dimensionality of the problem. The proposed temporal context scheme, progressively reduces the length of past frames by computing a linear combination of their features.

The results obtained with the proposed classifier are close to that of more complex algorithms while having a lower computational complexity. Furthermore, the proposed compressed temporal context approach outperforms the classic ‘full-frame’ approach with all the tested detectors when same feature vector length is used. The main advantage of the presented multichannel detection system is not the improvement on the number of true detections but the improvement on false positive rejection. The FPR of the proposed detector (K-LDA with 6 clusters and compressed temporal context) went from 21.3% when only one detector was used, to a perfect 0% when all 12 detectors were combined.

In light of the presented results, it seems like impulsive event detection (with high SNR and at a moderate distance) is not a difficult task, however, the problem gets increasingly difficult when propagation path diversity is introduced. While an isolated detector may not obtain an adequate performance in a spatially diverse scenario, the presented results clearly show that combining the decisions of a group of spatially distributed detectors is an efficient method to improve the global detection results. In general, the proposed impulsive sound event detection system offers a good performance while keeping the complexity of the algorithms involved as low possible, which makes it a good choice for automated surveillance on low-cost WASN.

7.2.3 Detection-based Sound Source Localization

Sound source localization is a classic problem of spatial signal processing. The most common approach, when the location of the microphones is known and they share a common timebase, is to estimate the TDoA of every available microphone pair using GCC. Localization systems are very likely to be specialized in some type(s) of source(s), hence, source localization is typically governed by some acoustic event detection system. However, the influence of the detection system in the localization process is typically limited to taking the decision on when to locate a source.

In chapter 5 two methods for sound source localization based on previous impulsive event detection were proposed. The first method takes advantage of the presence of a proper multichannel sound event detector in order to use shorter signal segments that that dictated by the sound propagation model for the GCC. The second method takes a different approach, it exploits the detection capabilities of the system as to bypass the need for cross-correlations and works instead with a set ToA estimates obtained with a double onset estimation scheme (local detection onset and event onset).

Both proposed methods tackle some of the problems associated with performing classical source localization in WASN where large inter-microphone distances call for long cross-correlations that are heavily affected by coherence loss between channels. Although, the main novelty resides in the second method, specially in the ML estimator that works with ToA estimates. It demonstrates how the solution to a seemingly independent problem (i.e., sound event detection) can be exploited to formulate new solutions to a classic multichannel signal processing problem. Furthermore, the presented results show that the ToA-based method is more reliable for the kind of signals expected from impulsive noise sources recorded outdoors. When all 12 microphones were used, the ToA-based method obtained an average localization of 1m while the best result of the GCC-based methods was 1.7m.

Regarding the effect of synchronization, the presented results prove that it is a main concern for sound source localization in WASNs. Clock offsets of a few milliseconds can have a devastating effect on the localization error, thus, tight synchronization within the network together with accurate knowledge about the microphone locations are part of the basic requirements for most source localization algorithms.

7.2.4 Efficient Sound Source Localization

Sound source localization computational complexity on large WASNs (both in physical size and number of nodes) quickly escalates given that a sizable number of long cross-correlations needs to be computed. Yet, the results presented in chapter 5 show that even long cross-correlations fail to correctly capture TDoAs due to the large differences between propagation paths that make signals captured at distant locations no longer resemble a time lagged version of one another.

The first of the source localization method presented in chapter 5 addresses this problem by reducing the length of the signals being cross-correlated. This simple modification not only induces an improvement on the localization results but also reduces the computational complexity and the bandwidth requirements significantly.

However the ToA-based proposal is again the superior method. The computational complexity of obtaining the ToA estimates is lower than that of computing the GCC used by most source localization algorithms, specially if the scalability of the algorithms is taken into consideration (M ToAs vs $M(M-1)/2$ TDoAs). Yet, it is worth mentioning that the minimization required to estimate the source location is very likely the most computationally expensive part of sound source localization algorithms, thus, development of a closed-form version of the proposed ML estimator would be very beneficial towards efficiency.

7.2.5 Multichannel Impulsive Acoustic Event Classification

The objective of classification as part of an automated acoustic surveillance system is to tell apart sound events signaling a potentially dangerous situation from similar but irrelevant sounds. Impulsive sound event classification poses a harder problem than detection, given that differences between impulsive sources are much less pronounced than differences between impulsive noise and other sounds. Furthermore, the influence of the propagation path is a dominant factor, so much so that the recorded events can be completely dissimilar due to differences on the recording environment and the relative locations of sources and receivers. Albeit, this problem can be lessen by using a spatially diverse receiver such as a WASN. Such a device can easily provide multiple observations of an acoustic event taken at different locations, making it possible to obtain a more robust classification by fusing the available data.

In chapter 6 two efficient multichannel classification schemes for impulsive sound source classification in two scenarios were proposed. The first method tackles the problem of gunshot classification on a WASN without array processing capabilities, while the second method takes full advantage of some of the algorithms presented in this thesis and deals with more generalist impulsive sources.

For the first scenario, since gunshot recordings present large differences depending on the propagation path, we proposed a classification tree based on the K-LDA classifier that uses the outcome of two binary classifiers to segment the space instead of clustering, in conjunction with some specialized features based on the acoustic model of gunshots. For the second scenario, multichannel classification was carried out using the same basic elements described on 4. In order to lessen the effect of the use of short signal frames, decision fusion was performed by first having every node compute an auxiliary output representative of each class output during a 'classification time window'.

Overall, the presented results show that spatial diversity obtained from an ensemble of classifiers working with multiple observation of an acoustic event is a valuable tool for impulsive source classification that can be easily exploited using decision fusion. In both tested scenarios the results show a similar error decrease for every classifier when more members are added to the classifier ensemble, which clearly points at spatial diversity as a major factor for achieving good classification accuracies.

7.2.6 Classification Aided by Spatial Information

Recordings of impulsive acoustic events depend heavily on the propagation path, therefore, in case the spatial relation between the source and the receivers is known, this information can be used to aid the classification process by giving the decisions taken by an ensemble of classifiers a certain degree of priority or weight.

Regarding the two methods presented in chapter 6; the first method is intended for networks lacking array processing capabilities, thus, nodes have to estimate their spatial relation towards the source in order to lessen the impact of spatial uncertainty. The second methods takes full advantage of some of the algorithms proposed in this thesis, thus, node to source distances can be used directly to weight the decisions taken by the ensemble.

The main novelty resides in the use of spatial information to weight the decision taken by every classifier in the ensemble based on different strategies. When the network is assumed not to be able to estimate the location of the source, spatial diversity is exploited with a novel ML decision fusion rule that considers diversity in the classifier ensemble. Classifier diversity comes from the proposed classification tree that uses the outcome of two binary classifiers to segment space and select one of various spatially-specialized weapon classifiers. Then, during the decision fusion, differences in performance between these specialized classifiers are exploited to further 'boost' the system. In the second scenario, spatial information is assumed to be known, thus, node to source distance is directly used to weight the decisions taken by the ensemble. Additionally, a combination of distance-based and SNR-based weights was proposed.

Regardless of the type of impulsive source or the methodology used to exploit spatial diversity, the presented results show that weighted decision fusion is a valuable tool for impulsive sound event classification on WASNs. In the first scenario, the classification error of the LS-LDA tree, fell by almost 10% just by adding a second node in conjunction with the ML fusion method, getting as low as 5.9% when 8 nodes are used. In the second scenario, error rate dropped from 23.9% with a single K-LDA classifier to 7.3% with an ensemble of 12 classifiers and the distance-based weights.

7.3 Future Research

The research carried out during this thesis is not intended as a definitive solution to any of the presented problems, but as a mere attempt at expanding the knowledge in the field of multichannel audio signal processing and its application to engineering constrained problems such as WASNs. Some of the ideas proposed in this thesis do indeed show potential for further research that could eventually lead to new research lines. Considering the theoretical studies and the results derived in this thesis, the next future research lines are proposed:

- Assessing the quality of estimates. Results presented in chapters 3 and 5 show that the influence of spurious estimates is the main source of error for spatial signal processing algorithms. Some outlier detection techniques deal with this problem by doing multiple evaluations each time using a different subset of observations. However, this method is overkill, and it inherently increases the computational load of an already data intensive problem. The ideal solution would be to 'measure' the confidence level of individual estimates based on some metric, so that weighted optimization could be applied. By doing so the quality of the results would be improved without a significative impact to computational complexity. Thus, further research is necessary in order to develop new solutions to this problem.

- Obtaining ToA estimates using machine learning. In light of the good performance obtained by the ToA-based sound source localization algorithm proposed in chapter 5, it should be possible to derive a similar methodology to work with ToA estimation during the self-localization process. This way, long cross-correlations may be substituted by some method based on the detection of a sequence of predetermined patterns, with the objective of lowering the computational complexity and improving the robustness of ToA estimation. Albeit this would imply a large divergence from the classic methods used to estimate time lag, therefore, a thorough study should take place before claiming the feasibility of machine learning based ToA estimation.

- Synchronization. As we have seen during this thesis, tight synchronization (ideally tens of microseconds) is a crucial part of array processing in WASNs. It is worth mentioning that the synchronization method presented on chapter 3 has two main weakness: clock drift and long distances. Since nodes are subjected to clock drift, synchronization must be 'refreshed' periodically, and that can prove as a nuisance to end users due to the repeated emission of reference acoustic signals. Additionally, sound-based methods are not suitable for implementation when large inter-microphone distances are involved, thus, RF-based solutions are favored. RF signals take a little over 3νs to travel 1km, so assuming an ad-hoc network, they could be easily exploited to obtain synchronization. There are already a number of networking synchronization protocols available, although, new solutions tailored to the particularities of WASN can be a valuable contribution.

- Recursive detection. As we mentioned in chapter 4, sound event detection intended for acoustic surveillance typically deals with "rare" sound events, thus, most of the power spent on continuously monitoring the environment is going to waste. Given that hierarchical detection schemes are fairly popular in the literature, it is worth taking the time to study the possibility of using a recursive detection strategy as to minimize energy consumption. Such a system would start with a very simple detector (e.g., energy based as to avoid DFT computation) and use a number of increasingly complex stages to obtain the final decision. Furthermore, since acoustic event detection is the starting point for multiple algorithms, minimizing detection energy consumption would imply an overall efficiency improvement for the system.

- Closed-form ToA-based sound source localization. Typically the most computationally expensive part of sound source localization algorithms is the minimization, therefore obtaining a closed-form solution would be a great aid towards efficiency. There are already various closed-form solutions based of TDoA estimates, however it is worth developing a closed-form version of the ToA-based algorithm presented on chapter 5.

- Influence of network topology. All the work developed in this thesis was done considering a fully connected network topology. It is important to notice that long distance transmission requires more power, specially at high data-rates, thus, a fully connected topology is far from ideal for (physically) large networks. Given that none of the algorithms proposed in this thesis requires high transmissions rates, topology selection and routing have been overlooked, however they can be critical for other applications such as: synchronization algorithms, real-time source-localization, beamforming, etc.

- Microphone subset selection. Similarly to the previous point, all the work in this thesis was done without considering the possibility of some nodes providing spurious information either caused by transmission errors, faulty sensors or even spatial effects (i.e., very long distance to the source, lack of direct line of sight, etc). Given that these problems are very likely to occur in a real scenario, future research should address methods to detect their presence in order to avoid their negative impact on the performance of the system. Additionally, selecting a subset of the best signals can prove beneficial towards efficiency by means of reducing the amount of information that needs to be processed (and/or transmitted), which is specially relevant for algorithms with poor scalability.

- Sub-array preprocessing for acoustic event detection and classification. The proposed solutions for multichannel detection and classification were developed using a single microphone per node. In case the nodes were equipped with a sub-array, it should be possible to use signal enhancement techniques (e.g., beamforming, deconvolution, etc) locally without the need for high transmission data rates or precise inter-node synchronization. In that scenario each node could obtain a ‘cleaner’ version of a target acoustic event which would potentially enhance the accuracy of the system. By having a WASN composed of a group of distributed arrays instead of single microphones a whole new group of techniques could be developed, thus further research is greatly encouraged.

- Source tracking and projectile trajectory estimation. Perhaps the only classic functionalities of acoustic surveillance systems that were not addressed on this thesis are source tracking and projectile trajectory estimation. These problems are closely related to sound source localization, albeit, both of them have their own particularities. Source tracking requires continuous (or at least periodical) sound emission such as that of a moving vehicle, thus, it is not feasible to track most impulsive sources. Projectile trajectory estimation is a specialized technique; its area of application is very narrow, yet it is a valuable tool nonetheless. In any case, it would be interesting to research new efficient methods to implement these techniques on a WASN.

Publications

During the development of this thesis some of the contributions have yielded several of publications on international journals and conferences alike. The work accepted and published either in journals or conference proceedings, as well as that currently under review, that serves to support the main contributions of this thesis is listed below.

Published International Journals

[P1] Héctor A Sánchez-Hevia, David Ayllón, Roberto Gil-Pita, and Manuel Rosa-Zurera. Indoor Self-Localization and Orientation Estimation of Smartphones Using Acoustic Signals. *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 3534829, 11 pages, 2017.

[P2] Héctor A Sánchez-Hevia, David Ayllón, Roberto Gil-Pita, and Manuel Rosa-Zurera. Maximum likelihood decision fusion for weapon classification in wireless acoustic sensor networks. *IEEE/ACM transactions on audio, speech, and language processing*, 25(6):1172-1182, 2017.

[P3] David Ayllón, Héctor A Sánchez-Hevia, Roberto Gil-Pita, Manuel Utrilla Manso and Manuel Rosa-Zurera. Indoor blind localization of smartphones by means of sensor data fusion. *IEEE Transactions on Instrumentation and Measurement*, 65(4), 783-794., 2016.

Submitted International Journals

[P4] Héctor A Sánchez-Hevia, Roberto Gil-Pita, and Manuel Rosa-Zurera. Efficient multi-channel impulsive sound event detection. *Engineering Applications of Artificial Intelligence*, Elsevier, (*submitted*).

Publications in preparation

[P5] Héctor A Sánchez-Hevia, Roberto Gil-Pita, and Manuel Rosa-Zurera. Impulsive sound source localization based on time domain onset estimation using a trained detector. *In preparation.*

[P6] Héctor A Sánchez-Hevia, Roberto Gil-Pita, and Manuel Rosa-Zurera. Weighted decision fusion for impulsive sound classification with a distributed microphone array. *In preparation.*

Published International Conference Papers

[cP1] Héctor A Sánchez-Hevia, David Ayllón, Roberto Gil-Pita, and Manuel Rosa-Zurera. Gunshot classification from single-channel audio recordings using a divide and conquer approach, In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, vol. 2: 233-240, 2015.

[cP2] David Ayllón, Héctor A Sánchez-Hevia, Roberto Gil-Pita, Manuel Utrilla Manso and Manuel Rosa-Zurera. Indoor blind localization of smartphones by means of sensor data fusion. In *2015 IEEE Sensors Applications Symposium (SAS)*, 6 pages, 2015.

Some additional conference papers indirectly related to the content of this thesis are the following:

[cP3] Roberto Gil-Pita, Héctor Sánchez-Hevia, Cosme Llerena-Aguilar, Inma Mohino-Herranz, Manuel Utrilla-Manso, and Manuel Rosa-Zurera. Distributed and collaborative sound environment information extraction in binaural hearing aids. In *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016*, pages 1-5. IEEE, 2016.

[cP4] Héctor A Sánchez-Hevia, Cosme Llerena-Aguilar, Guillermo Ramos-Auñón, and Roberto Gil-Pita. Automatic vocal percussion transcription aimed at mobile music production. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[cP5] Hector A Sánchez-Hevia, Roberto Gil-Pita, and Manuel Rosa-Zurera. Fpga-based real- time acoustic camera using pdm mems microphones with a custom demodulation filter. In *8th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014*, pages 181-184. IEEE, 2014.

[cP6] Héctor A Sánchez-Hevia, Inma Mohino-Herranz, Roberto Gil-Pita, and Manuel Rosa- Zurera. Memory requirements reduction technique for delay storage in real time acoustic cameras. In *Audio Engineering Society Convention 136*. Audio Engineering Society, 2014.

Bibliography

[AB79] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[AEQGC06] Ahmad Rami Abu-El-Quran, Rafik A Goubran, and Adrian DC Chan. Security monitoring using microphone arrays and audio classification. *IEEE Transactions on Instrumentation and Measurement*, 55(4):1025–1032, 2006.

[AH84] Mordechai Azaria and David Hertz. Time delay estimation by generalized cross correlation methods. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):280–285, 1984.

[AH01] Ali N Akansu and Richard A Haddad. *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic Press, 2001.

[AHM$^+$14] Malik Aqeel Anwar, Hammad Hassan, Hasan Maqbool, Akif Rehman, and Muhammad Tahir. Acoustic sensor network relative self-calibration using joint tdoa and doa with unknown beacon positions. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 3064–3069. IEEE, 2014.

[Aka78] Adnan Akay. A review of impact noise. *The Journal of the Acoustical Society of America*, 64(4):977–987, 1978.

[ANR74] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[APH14] Xavier Alameda-Pineda and Radu Horaud. A geometric approach to sound source localization from time-delay estimates. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(6):1082–1095, 2014.

[APP$^+$17] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, and Tuomas Virtanen. Sound event detection in multichannel audio using spatial and harmonic features. *arXiv preprint arXiv:1706.02293*, 2017.

[AR77] Jont B Allen and Lawrence R Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, 1977.

[ARM10] ARM Limited. *Cortex-M4 Technical Reference Manual*, March 2010. Rev. r0p0.

[ASJS07] MR Azimi-Sadjadi, Y Jiang, and S Srinivasan. Acoustic classification of battlefield transient events using wavelet sub-band features. In *Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, volume 6562, page 656215. International Society for Optics and Photonics, 2007.

[ASMM11] Shoko Araki, Hiroshi Sawada, Ryo Mukai, and Shoji Makino. Doa estimation for multiple sparse sources with arbitrarily arranged multiple sensors. *Journal of Signal Processing Systems*, 63(3):265–275, 2011.

[AUM13] Talal Ahmed, Momin Uppal, and Abubakr Muhammad. Improving efficiency and reliability of gunshot detection systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 513–517. IEEE, 2013.

[BCH08] Jacob Benesty, Jingdong Chen, and Yiteng Huang. *Microphone array signal processing*, volume 1. Springer Science & Business Media, 2008.

[BDA$^+$05] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on speech and audio processing*, 13(5):1035–1047, 2005.

[Ber11] Alexander Bertrand. Applications and trends in wireless acoustic sensor networks: A signal processing perspective. In *Communications and Vehicular Technology in the Benelux (SCVT), 2011 18th IEEE Symposium on*, pages 1–6. IEEE, 2011.

[BGBBJ03] MC Bérengier, B Gauvreau, Ph Blanc-Benon, and D Juvé. Outdoor sound propagation: A short review on analytical and numerical approaches. *Acta Acustica united with Acustica*, 89(6):980–991, 2003.

[BNM11] Steven D Beck, Hirotaka Nakasone, and Kenneth W Marr. Variations in recorded acoustic gunshot waveforms generated by small firearms. *The Journal of the Acoustical Society of America*, 129(4):1748–1759, 2011.

[BOO11] BOOM Library GbR. *"GUNS - Construction Kit"*, 2011.

[BP92] Judith C Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[BS97] Michael S Brandstein and Harvey F Silverman. A robust method for speech signal time-delay estimation in reverberant rooms. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 375–378. IEEE, 1997.

[CAA$^+$17] Maximo Cobos, Fabio Antonacci, Anastasios Alexandridis, Athanasios Mouchtaris, and Bowon Lee. A survey of sound source localization methods in wireless acoustic sensor networks. *Wireless Communications and Mobile Computing*, 2017, 2017.

[CAST13] Antonio Canclini, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro. Acoustic source localization with distributed asynchronous microphone networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(2):439–443, 2013.

[CCTM16] Marco Crocco, Marco Cristani, Andrea Trucco, and Vittorio Murino. Audio surveillance: A systematic review. *ACM Computing Surveys (CSUR)*, 48(4):52, 2016.

[CDBBM12] Marco Crocco, Alessio Del Bue, Matteo Bustreo, and Vittorio Murino. A closed form solution to the microphone position self-calibration problem. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2597–2600. IEEE, 2012.

[CER05] Chloé Clavel, Thibaut Ehrette, and Gaël Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, 2005. ICME 2005. IEEE International conference on*, pages 1306–1309. IEEE, 2005.

[CHHV15] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–7. IEEE, 2015.

[CHOS95] I Céspedes, Y Huang, J Ophir, and S Spratt. Methods for estimation of subsample time delays of digitized echo signals. *Ultrasonic imaging*, 17(2):142–171, 1995.

[CK14] Sachin Chachada and C-C Jay Kuo. Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

[CLW69] James W Cooley, Peter AW Lewis, and Peter D Welch. The fast fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34, 1969.

[CML11] Maximo Cobos, Amparo Marti, and Jose J Lopez. A modified srp-phat functional for robust real-time sound source localization with scalable spatial sampling. *IEEE Signal Processing Letters*, 18(1):71–74, 2011.

[CPSB$^+$16] Maximo Cobos, Juan J Perez-Solano, Óscar Belmonte, German Ramos, and Ana M Torres. Simultaneous ranging and self-positioning in unsynchronized wireless acoustic sensor networks. *IEEE Transactions on Signal Processing*, 64(22):5993–6004, 2016.

[CRHK12] Woohyun Choi, Jinsang Rho, David K Han, and Hanseok Ko. Selective background adaptation based abnormal acoustic event recognition for audio surveillance. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 118–123. IEEE, 2012.

[CRJC$^+$11] Alfonso Chacon-Rodriguez, Pedro Julian, Liliana Castro, Pablo Alvarado, and Néstor Hernández. Evaluation of gunshot detection algorithms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(2):363–373, 2011.

[CS03] Michael Cowling and Renate Sitte. Comparison of techniques for environmental sound recognition. *Pattern recognition letters*, 24(15):2895–2907, 2003.

[CV86] Z Chair and PK Varshney. Optimal data fusion in multiple sensor detection systems. *IEEE Transactions on Aerospace and Electronic Systems*, (1):98–101, 1986.

[Dan06] E Danicki. The shock wave-based acoustic sniper localization. *Nonlinear analysis: theory, methods & applications*, 65(5):956–962, 2006.

[DBAP99] Alain Dufaux, Laurent Besacier, Michael Ansorge, and Fausto Pellandini. Automatic detection and classification of wide-band acoustic signals. In *Proc. of ASA 99, 137 th Meeting of the Acoustical Society of America and Forum Acusticum*, volume 99, pages 14–19, 1999.

[DGA$^+$05] Prabal Dutta, Mike Grimmer, Anish Arora, Steven Bibyk, and David Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 70. IEEE Press, 2005.

[DH04] Marco Duarte and Yu-Hen Hu. Distance-based decision fusion in a distributed wireless sensor network. *Telecommunication Systems*, 26(2-4):339–350, 2004.

[Dij83] Theo Dijkstra. Some comments on maximum likelihood and partial least squares methods. *Journal of Econometrics*, 22(1-2):67–90, 1983.

[DKT17] Dua Dheeru and Efi Karra Taniskidou. Uci machine learning repository, 2017.

[DKW10] Thyagaraju Damarla, Lance M Kaplan, and Gene T Whipps. Sniper localization using acoustic asynchronous sensors. *IEEE Sensors Journal*, 10(9):1469–1478, 2010.

[DSY07] Hoang Do, Harvey F Silverman, and Ying Yu. A real-time srp-phat source location implementation using stochastic region contraction (src) on a large-aperture microphone array. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–121. IEEE, 2007.

[DT13] Mustapha Djeddou and Tayeb Touhami. Classification and modeling of acoustic gunshot signatures. *Arabian journal for science and engineering*, 38(12):3399–3406, 2013.

[Efr79] Bradley Efron. Bootstrap methods: another look at the jackknife. *The annals of Statistics*, pages 1–26, 1979.

[EGE02] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[Emb96] Tony FW Embleton. Tutorial on sound propagation outdoors. *The Journal of the Acoustical Society of America*, 100(1):31–48, 1996.

[FAJ10] Izabela L Freire and Jose A Apolinario Jr. Gunshot detection in noisy environments. In *International Telecommunications Symposium*, 2010.

[FB87] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

[FBP06] John C Freytaga, Durand R Begaultb, and Christopher A Peltierc. The acoustics of gunfire. In *INTER-NOISE*, 2006.

[FCL02] Brian G Ferguson, Lionel G Criswick, and Kam W Lo. Locating far-field impulsive sound sources in air by triangulation. *The Journal of the Acoustical Society of America*, 111(1):104–116, 2002.

[FDON06] Mário AT Figueiredo, J Bioucas Dias, João P Oliveira, and Robert D Nowak. On total variation denoising: A new majorization-minimization algorithm and an experimental comparisonwith wavalet denoising. In *Image Processing, 2006 IEEE International Conference on*, pages 2633–2636. IEEE, 2006.

[Fis36] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

[FTCP93] Kevin S Fansler, William P Thompson, John S Carnahan, and Brendan J Patton. A parametric investigation of muzzle blast. Technical report, ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD, 1993.

[GH15] Jürgen T Geiger and Karim Helwani. Improving event detection for audio surveillance using gabor filterbank features. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 714–718. IEEE, 2015.

[Gis90] Herbert Gish. A probabilistic approach to the understanding and training of neural network classifiers. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 1361–1364. IEEE, 1990.

[GKH13] Nikolay D Gaubitch, W Bastiaan Kleijn, and Richard Heusdens. Auto-localization in ad-hoc microphone arrays. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 106–110. IEEE, 2013.

[GLB15] Mario Guggenberger, Mathias Lux, and Laszlo Böszörmenyi. An analysis of time drift in hand-held recording devices. In *International Conference on Multimedia Modeling*, pages 203–213. Springer, 2015.

[GNM15] Sebastian Gergen, Anil Nagathil, and Rainer Martin. Classification of reverberant audio signals using clustered ad hoc distributed microphones. *Signal Processing*, 107:21–32, 2015.

[GPAR$^+$15] Roberto Gil-Pita, David Ayllón, José Ranilla, Cosme Llerena-Aguilar, and Irene Díaz. A computationally efficient sound environment classifier for hearing aids. *IEEE Transactions on Biomedical Engineering*, 62(10):2358–2368, 2015.

[GPKM14] Panagiotis Giannoulis, Gerasimos Potamianos, Athanasios Katsamanis, and Petros Maragos. Multi-microphone fusion for detection of speech and acoustic events in smart spaces. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 2375–2379. IEEE, 2014.

[GS08] Matthew D Gillette and Harvey F Silverman. A linear closed-form algorithm for source localization from time-differences of arrival. *IEEE Signal Processing Letters*, 15:1–4, 2008.

[GVT$^+$07] Luigi Gerosa, Giuseppe Valenzise, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. Scream and gunshot detection in noisy environments. In *Signal Processing Conference, 2007 15th European*, pages 1216–1220. IEEE, 2007.

[Har61] Herman O Hartley. The modified gauss-newton method for the fitting of non-linear regression functions by least squares. *Technometrics*, 3(2):269–280, 1961.

[Har78] Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[HH86] D Henderson and RP Hamernik. Impulse noise: critical review. *The Journal of the Acoustical Society of America*, 80(2):569–584, 1986.

[HH91] Roger P Hamernik and Keng D Hsueh. Impulse noise: some definitions, physical acoustics and other considerations. *The Journal of the Acoustical Society of America*, 90(1):189–196, 1991.

[HLM80] M Hunt, Matthew Lennig, and Paul Mermelstein. Experiments in syllable-based recognition of continuous speech. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, volume 5, pages 880–883. IEEE, 1980.

[HMdC$^+$16] Diego B Haddad, Wallace A Martins, Mauricio do VM da Costa, Luiz WP Biscainho, Leonardo O Nunes, and Bowon Lee. Robust acoustic self-localization of mobile devices. *IEEE Transactions on Mobile Computing*, 15(4):982–995, 2016.

[HRZZ09] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

[HVRVO98] Paul M Hofman, Jos GA Van Riswick, and A John Van Opstal. Relearning sound localization with new ears. *Nature neuroscience*, 1(5):417, 1998.

[IC06] ISO-CEN.17201-2. Acoustics. noise from shooting ranges. part 2: Estimation of muzzle blast and projectile sound by calculation, 2006.

[Jai10] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[JSHU12] Florian Jacob, Joerg Schmalenstroeer, and Reinhold Haeb-Umbach. Microphone array position self-calibration from reverberant speech input. In *Acoustic Signal Enhancement; Proceedings of IWAENC 2012; International Workshop on*, pages 1–4. VDE, 2012.

[JW00] Prasad Jogalekar and Murray Woodside. Evaluating the scalability of distributed systems. *IEEE Transactions on parallel and distributed systems*, 11(6):589–603, 2000.

[KC76] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976.

[Kie77] Susan Werner Kieffer. Sound speed in liquid-gas mixtures: Water-air and water-steam. *Journal of Geophysical research*, 82(20):2895–2904, 1977.

[KLC14] Jozef Kotus, Kuba Lopatka, and Andrzej Czyzewski. Detection and localization of selected acoustic events in acoustic field for smart surveillance applications. *Multimedia Tools and Applications*, 68(1):5–21, 2014.

[KLP$^+$13] Eva Kiktova, Martin Lojka, Matus Pleva, Jozef Juhar, and Anton Cizmar. Comparison of different feature types for acoustic event detection system. In *International Conference on Multimedia Communications, Services and Security*, pages 288–297. Springer, 2013.

[Knu76] Donald E Knuth. Big omicron and big omega and big theta. *ACM Sigact News*, 8(2):18–24, 1976.

[KO11] Bekir Karlik and A Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122, 2011.

[KPD06] A Kawalec, J Pietrasinski, and E Danicki. Selected problems of sniper acoustic localization. Technical report, DTIC Document, 2006.

[Kru64] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[KS06] Amin Karbasi and Akihiko Sugiyama. A doa estimation method for an arbitrary triangular microphone arrangement. In *Signal Processing Conference, 2006 14th European*, pages 1–5. IEEE, 2006.

[KWH13] Werner Bertels Karl Wilhem Hirsch. Estimation of the directivity pattern of muzzle blasts. In *AIA-DAGA*, 2013.

[LD11] L Ladha and T Deepa. Feature selection methods and algorithms. *International journal on computer science and engineering*, 3(5):1787–1797, 2011.

[Ler12] Alexander Lerch. *An introduction to audio content analysis: Applications in signal processing and music informatics*. John Wiley & Sons, 2012.

[LLW09] Zhi Li, Xiang Li, and Yongjun Wang. A calibration method for magnetic sensors and accelerometer in tilt-compensated digital compass. In *Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on*, pages 2–868. IEEE, 2009.

[LS97] Louisa Lam and Ching Y Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 27(5):553–568, 1997.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[LWHS02] Dan Li, Kerry D Wong, Yu Hen Hu, and Akbar M Sayeed. Detection, classification, and tracking of targets. *IEEE signal processing magazine*, 19(2):17–29, 2002.

[Lyo10] Richard F Lyon. Machine hearing: An emerging field. *Ieee signal processing magazine*, 27(5):131–139, 2010.

[Mah07] Robert C Maher. Acoustical characterization of gunshots. In *Signal Processing Applications for Public Security and Forensics, 2007. SAFE'07. IEEE Workshop on*, pages 1–5. IEEE, 2007.

[McC01] Iain McCowan. Microphone arrays: A tutorial. *Queensland University, Australia*, pages 1–38, 2001.

[ML65] F Edward Mc Lean. Some nonasymptotic effects on the sonic boom of large airplanes. , National Aeronautics and Space Administration (NASA), Washington, D.C., USA, 1965.

[MNH08] Baljeet Malhotra, Ioanis Nikolaidis, and Janelle Harms. Distributed classification of acoustic targets in wireless audio-sensor networks. *Computer Networks*, 52(13):2582–2593, 2008.

[Mor78] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[MP10] Toni Mäkinen and Pasi Pertilä. Shooter localization and bullet trajectory, caliber, and speed estimation based on detected firing sounds. *Applied acoustics*, 71(10):902–913, 2010.

[MS08] Robert C Maher and Steven R Shaw. Deciphering gunshot recordings. In *Audio Engineering Society Conference: 33rd International Conference: Audio Forensics-Theory and Practice*. Audio Engineering Society, 2008.

[MS10] Robert C Maher and Steven R Shaw. Directional aspects of forensic gunshot recordings. In *Audio Engineering Society Conference: 39th International Conference: Audio Forensics: Practices and Challenges*. Audio Engineering Society, 2010.

[MSDC10] Utthara Gosa Mangai, Suranjana Samanta, Sukhendu Das, and Pinaki Roy Chowdhury. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical review*, 27(4):293–307, 2010.

[MSK$^+$09] Timo Machmer, Alexej Swerdlow, Kristian Kroschel, Jorge Moragues, Luis Vergara, and Jorge Gosálbez. Robust impulsive sound source localization by means of an energy detector for temporal alignment and pre-classification. In *Signal Processing Conference, 2009 17th European*, pages 1409–1412. IEEE, 2009.

[MSVG11] Jorge Moragues, Arturo Serrano, Luis Vergara, and Jorge Gosálbez. Acoustic detection and classification using temporal and frequency multiple energy detector features. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1940–1943. IEEE, 2011.

[NPF09] Stavros Ntalampiras, Ilyas Potamitis, and Nikos Fakotakis. On acoustic surveillance of hazardous situations. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 165–168. IEEE, 2009.

[NS96] Boon Chong Ng and Chong Meng Samson See. Sensor-array calibration using a maximum-likelihood approach. *IEEE Transactions on Antennas and Propagation*, 44(6):827–835, 1996.

[OM99] David W Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.(JAIR)*, 11:169–198, 1999.

[PAG95] Roy D Patterson, Mike H Allerhand, and Christian Giguere. Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform. *The Journal of the Acoustical Society of America*, 98(4):1890–1894, 1995.

[PAK$^+$05] Neal Patwari, Joshua N Ash, Spyros Kyperountas, Alfred O Hero, Randolph L Moses, and Neiyer S Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal processing magazine*, 22(4):54–69, 2005.

[Par08] J-M Parot. Localizing impulse sources in an open space by time reversal with very few transducers. *Applied Acoustics*, 69(4):311–324, 2008.

[PF14] Axel Plinge and Gernot A Fink. Geometry calibration of multiple microphone arrays in highly reverberant environments. In *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on*, pages 243–247. IEEE, 2014.

[PJHUF16] Axel Plinge, Florian Jacob, Reinhold Haeb-Umbach, and Gernot A Fink. Acoustic microphone geometry calibration: An overview and experimental evaluation of state-of-the-art algorithms. *IEEE Signal Processing Magazine*, 33(4):14–29, 2016.

[PKL11] Jukka Pätynen, Brian FG Katz, and Tapio Lokki. Investigations on the balloon as an impulse source. *The Journal of the Acoustical Society of America*, 129(1):EL27–EL33, 2011.

[PMH$^+$15] Huy Phan, Marco Maass, Lars Hertel, Radoslaw Mazur, and Alfred Mertins. A multichannel fusion framework for audio event detection. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*, pages 1–5. IEEE, 2015.

[PP17] Mikko Parviainen and Pasi Pertilä. Self-localization of dynamic user-worn microphones from observed speech. *Applied Acoustics*, 117:76–85, 2017.

[PPH14] Mikko Parviainen, Pasi Pertila, and Matti S Hamalainen. Self-localization of wireless acoustic sensors in meeting rooms. In *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2014 4th Joint Workshop on*, pages 152–156. IEEE, 2014.

[PR12] Ioannis Paraskevas and Maria Rangoussi. Feature extraction for audio classification of gunshots using the hartley transform. *Open Journal of Acoustics*, 2(03):131, 2012.

[PSZ12] Chunyi Peng, Guobin Shen, and Yongguang Zhang. Beepbeep: A high-accuracy acoustic-based system for ranging and localization using cots devices. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(1):4, 2012.

[Ras86] KB Rasmussen. Outdoor sound propagation under the influence of wind and temperature gradients. *Journal of sound and vibration*, 104(2):321–335, 1986.

[Rin00] Jens Holger Rindel. The use of computer modeling in room acoustics. *Journal of vibroengineering*, 3(4):219–224, 2000.

[RM04] Kay Romer and Friedemann Mattern. The design space of wireless sensor networks. *IEEE wireless communications*, 11(6):54–61, 2004.

[SCK14] Dae-Hoon Seo, Jung-Woo Choi, and Yang-Hann Kim. Impulsive sound source localization using peak and rms estimation of the time-domain beamformer output. *Mechanical Systems and Signal Processing*, 49(1-2):95–105, 2014.

[SEA02] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio Engineering Society*, 50(4):249–262, 2002.

[SH05] Xiaohong Sheng and Yu-Hen Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1):44–53, 2005.

[SH15] Etto L Salomons and Paul JM Havinga. A survey on the feasibility of sound classification on wireless sensor nodes. *Sensors*, 15(4):7462–7498, 2015.

[SHV$^+$11] Janos Sallai, Will Hedgecock, Peter Volgyesi, Andras Nadas, Gyorgy Balogh, and Akos Ledeczi. Weapon classification and shooter localization using distributed multichannel acoustic sensors. *Journal of Systems Architecture*, 57(10):869–885, 2011.

[SJHU$^+$11] Joerg Schmalenstroeer, Florian Jacob, Reinhold Haeb-Umbach, Marius H Hennecke, and Gernot A Fink. Unsupervised geometry calibration of acoustic sensor networks using source correspondences. In *Interspeech*, pages 597–600, 2011.

[SK16] Anil Sharma and Sanjit Kaul. Two-stage supervised learning-based method to detect screams and cries in urban environments. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(2):290–299, 2016.

[SNM13] Umamahesh Srinivas, Nasser M Nasrabadi, and Vishal Monga. Graph-based multi-sensor fusion for acoustic signal classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 261–265. IEEE, 2013.

[SPS98] Brian M Sadler, Tien Pham, and Laurel C Sadler. Optimal and wavelet-based shock wave detection and estimation. *The Journal of the Acoustical Society of America*, 104(2):955–963, 1998.

[SR87] HC Schau and AZ Robinson. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(8):1223–1225, 1987.

[SSP02] Joshua M Sachar, Harvey F Silverman, and William R Patterson. Position calibration of large-aperture microphone arrays. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–1797. IEEE, 2002.

[SSSS13] Alexander Sutin, Hady Salloum, Alexander Sedunov, and Nikolay Sedunov. Acoustic detection, tracking and classification of low flying aircraft. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, pages 141–146. IEEE, 2013.

[TDVB97] David MJ Tax, Robert PW Duin, and Martijn Van Breukelen. Comparison between product and mean classifier combination rules. In *Proc. Workshop on Statistical Pattern Recognition, Prague, Czech*, 1997.

[Thr06] Sebastian Thrun. Affine structure from sound. In *Advances in Neural Information Processing Systems*, pages 1353–1360, 2006.

[TN09] Andrey Temko and Climent Nadeu. Acoustic event detection in meeting-room environments. *Pattern Recognition Letters*, 30(14):1281–1288, 2009.

[VA12] Xavier Valero and Francesc Alías. Gammatone wavelet features for sound classification in surveillance applications. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1658–1662. IEEE, 2012.

[VDMPVdH09] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

[Wer90] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[Whi52] Gerald Beresford Whitham. The flow pattern of a supersonic projectile. *Communications on pure and applied mathematics*, 5(3):301–348, 1952.

[WHRC16] Lin Wang, Tsz-Kin Hon, Joshua D Reiss, and Andrea Cavallaro. Self-localization of ad-hoc arrays using time difference of arrivals. *IEEE Transactions on Signal Processing*, 64(4):1018–1033, 2016.

[WK74a] Pearl G Weissler and Michael T Kobal. Noise of police firearms. *The Journal of the Acoustical Society of America*, 56(5):1515–1522, 1974.

[WK74b] Pearl G Weissler and Michael T Kobal. Noise of police firearms. *The Journal of the Acoustical Society of America*, 56(5):1515–1522, 1974.

[Ye07] Jieping Ye. Least squares linear discriminant analysis. In *Proceedings of the 24th international conference on Machine learning*, pages 1087–1093. ACM, 2007.

[YS06] Bin Yu and Katia Sycara. Learning the quality of sensor data in distributed decision fusion. In *Information Fusion, 2006 9th International Conference on*, pages 1–8. IEEE, 2006.

[ZHNZ11] Haichao Zhang, Thomas S Huang, Nasser M Nasrabadi, and Yanning Zhang. Heterogeneous multi-metric learning for multi-sensor fusion. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[ZML10] Dong Zhao, Huadong Ma, and Liang Liu. Event classification for living environment surveillance using audio sensor networks. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 528–533. IEEE, 2010.

[ZZHJH10] Xiaodan Zhuang, Xi Zhou, Mark A Hasegawa-Johnson, and Thomas S Huang. Real-world acoustic event detection. *Pattern Recognition Letters*, 31(12):1543–1551, 2010.