Sparse separation of sources in 3D soundscapes

Oliver Bunting

Ph.D.

The University of York
Department of Electronics

September 2010

Sparse separation of sources in 3D soundscapes

Submitted in accordance to the requirements of the University of York for the degree of Doctor of Philosophy.

Oliver Bunting

September 2010

The University of York
Department of Electronics

Abstract

A novel blind source separation algorithm applicable to extracting sources from within 3D soundscapes is presented. The algorithm is based on constructing a binary mask based on directional information. The validity of filtering using binary masked based on the ω-disjoint assumption is examined for several typical scenarios. Results for these test environments show an improvement by an order of magnitude when compared to similar work using speech mixtures.

Also presented is the novel application of a dual-tree complex wavelet transform to sparse source separation, providing an alternative transformation to the short-time Fourier transform often used in this area. Results are presented showing comparable signal-to-interference performance, and significantly improved signal-to-distortion performance when compared against the short time Fourier transform.

Results presented for the separation algorithm include quantitative measures of the separation performance for robust comparison against other separation algorithms.

Consideration is given to the related problem of localising sources within 3D soundscapes. Two novel methods are presented, the first using a peak estimation on a spherical histogram constructed using a geodesic grid, the second by adapting a self learning plastic self-organising map to operate on the surface of a unit sphere.

It is concluded that the separation algorithm presented is effective for soundscapes comprising ecological or zoological sources. Specific areas for further work are recognised, both in terms of isolated technologies and towards the integration of this work into an instrument for soundscape recognition, evaluation and identification.

Contents

List of Tables

List of Figures

Glossary of Terms

BS	British standard
CWT	continuous wavelet transform
DFT	discrete Fourier transform
DirAC	directional audio coding
DOA	direction of arrival
DTCWT	dual-tree complex wavelet transform
DUET	degenerate unmixing and estimation technique
DWT	discrete wavelet transform
FFT	fast Fourier transform
FIR	finite impluse response
GNU GPL	GNU general public license
HRTF	head-related transfer function
ICA	independant component analysis
IDTCWT	inverse dual-tree complex wavelet transform
IIR	infinite impluse response
ILD	inter-aural level difference
ISRIE	Instrument for soundscape recognition, evaluation and identification

ISRIE	instrument for soundscape recognition, evaluation and identification
ITD	inter-aural time difference
IWT	inverse wavelet transform
ML	maximum likelihood
PCA	principle component analysis
PPG	planning and policy guidence
PSOM	plastic self-organising map
PSR	preserved signal ratio
SDR	signal-to-distortion ratio
SIR	signal-to-interference ratio
SNR	signal-to-noise ratio
SOM	self-organising map
STFT	short-time Fourier transform
TIFROM	time-frequency ratio of mixtures algorithm
WT	wavelet transform

Chapter 1

Introduction

Contents

1.1 The ISRIE Project

1.1.1 Inception

The ISRIE (instrument for soundscape recognition, identification and evaluation) project [6] was born following an EPSRC sandpit event in 2006. ISRIE is a joint project with collaborators from the University of York, the University of Huddersfield (now Newcastle), along with the Institute of Sound and Vibration Research (ISVR), based at the University of Southampton. The instrument is envisioned to be capable of separating out sound components from within a soundfield and automatically classifying them.

1.1.2 Project division

The project is divided into three broad categories. The impact ISRIE could have on existing and future legislation is covered by Christos Karatsovis and Stuart Dyne at ISVR based in Southampton University. Colleagues Prof. Gui Yun Tian and Omar Bouzid at Huddersfield / Newcastle University performed some sound propagation modelling and prototyping of wireless monitoring systems. The research into methods of source separation and classification was undertaken at the University of York, overseen by Dr David Chesmore. The work package undertaken in York is divided in to classification algorithms, researched by Jon Stammers, and the area of source separation which is the subject of this thesis.

1.1.3 Rational

Monitoring of soundscapes is routine in urban planning for residential and industrial buildings, as well as for assessing the validity of noise complaints. Soundscapes may also be monitored for research in other areas, such as health, wildlife and ecological

studies.

Currently, monitored soundscapes are typically expressed as A-weighted sound pressure levels, averaged over some time period, often the hours of day or night. The ISRIE project goal is to work towards the development of instrumentation to characterise a soundfield by localising the constituent sources within the source field both spatially and temporally. Temporal localisation would allow the automated identification of infrequent loud events, such as military aircraft, pneumatic drills or rail services. These loud sources are a potential source of irritation in a soundscape, but contribute little to A-weighted long-term averaged levels.

Being able to decompose a soundscape enables more automated soundscape monitoring to existing standards such as PPG 24 [4] and BS 4142 [2]. It would also pave the way for a review of existing legislation. For rural and ecological soundscape monitoring, spatial and temporal localisation of sound sources enables the development of automatic species recognition and bio-diversity monitoring.

A project with similar goals, but limited in scope to specific urban environments is discussed in [3]. Defreville et al. note that the challenge of classifying sounds using their acoustic features is by far the biggest technical challenge, complicated by simultaneously active sources in a recording. ISRIE aims to ease this burden by first separating sounds to improve the signal to interference ratio of the recording presented to the classification algorithm.

1.2 Noise Metrics

Depending on the measurement being undertaken, differing sound metrics are currently employed to describe a soundfield. Table 1.1 gives definitions of some of the most commonly used [14].

Also very common is the use of A-weighted sound pressure levels. This is a frequency-dependent weighting applied to the sound pressure level that roughly follows the human

L_{max}	The maximum instantaneous sound pressure level during a specified period
L_{min}	The minimum instantaneous sound pressure level during a specified period
L_{eq}	The average sound pressure level over a specified time
L_{90}	The minimum sound pressure level observed for 90% of the time
$L_{A,90}$	The minimum A-weighted sound pressure level observed for 90% of the time
$L_{Amax,s}$	The maximum instantaneous A-weigthed sound pressure level during a period of one second duration
SEL	The sound exposure level is a unit for the equivalent noise level of the total sound energy during an event scaled to a 1 second time scale
T_A	The total time the instantaneous sound level exceeds a specified threshold during a time period

Table 1.1: Commonly used sound metrics

ear's frequency response. A-weighted metrics are generally used when the sound pressure level is applicable to humans.

Whilst A-weighted sound pressure levels are the current preferred metric for numerous studies, their applicability to soundscapes where the perpuse of the study is the soundscape's effect on non-human subjects much be questioned. As a minimum, to preserve the applicability of data sets to the widest possible set of applications, A-weighting should not be applied to the underlying data.

1.3 Potential areas of application

1.3.1 Noise monitoring

Mapping tranquil rural areas

A study with the aim of establishing a baseline for soundscapes in Ireland that are considered 'relatively quiet areas' has been conducted by the environmental protection agency (EPA)[18]. A relatively quiet area is defined in this paper as

> "An area, delimited by national or regional competent authority that

> is undisturbed by noise from traffic, industry or recreational activities and where natural quiet can be enjoyed"

This definition of a relatively quiet area is also described as the absence of anthropogenic noise, or extreme natural noise.

The metric used to characterise this environment is $L_{A,90} < 30$ dB for a total of at least one hour during day-time or evening, and for a total of at least 3 hours in any given night-time period. These metrics are combined with the requirement for the area to be distanced from sources of anthropogenic noise such as urban centres, transport links and industry.

These distancing requirements are in place to remove the effect of external anthropogenic sources on very quiet areas. An automated tool for classifying the noise sources would remove the need for this distancing requirement, and simplify the classification of quiet areas.

Long term wilderness studies

Soundscapes in national parks are of interest to researchers, not only for monitoring intrusive anthropogenic noise sources, but also for ecological and biodiversity studies [11].

Requirements for a long term study of soundscapes in national parks is presented in [12]. Maher argues that such long term studies are required to provide statistically valid research. The task of detecting and classifying sounds present within the long term recording of the soundscape is recognised as forming the mostly challenging aspect of such a proposal. Maher calls for the development of automated technologies to perform the classification and measurement of sources contributing to the recorded soundscape.

This is precisely the kind of application that ISRIE could be deployed in, allowing such long term studies to be viable, and allowing statistically significant audio work to be more easily performed.

In a long term study of the soundscape within Yellowstone National Park, remote acoustic sensors were deployed at several sites [1]. The recordings were analysed to determine and classify all identifiable sounds within an audio clip, and the duration of each source recorded.

This analysis was performed using a team of researchers, of nominally equivalent hearing abilities. To reduce the amount of data processing required, ten second samples were taken from each minute of recording to reduce the data by a factor of six.

ISRIE could be applied to such long term audio processing with great benefit, allowing continuous audio to be analysed, as well as reducing the workload of repetitive tasks by researchers in sound classification and subsequent soundscape analysis.

Nuisance noise

The link between adverse health effects and sound exposure is considered in [17]. Skånberg and Öhrström note that the perceived annoyance caused by sound is not purely a function of sound level, but also subject to the source of the sound, its duration, as well as its time-varying characteristics.

Skånberg and Öhrström propose the creation of environments exposed to soundscapes that are perceived to be annoying, and those considered tranquil, and assessing the impact they have on observable health indicators. ISRIE would allow the soundscapes to be better characterised, allowing a more detailed analysis of the noise factors within soundscapes against recorded levels of annoyance.

1.3.2 Animal studies

Ascertaining the effects of noise on wildlife is an active area of research. A typical review of literature on this subject [16] highlights some of the soundscape environments of interest:

- Assessment of behavioural change in deer populations caused by snow mobile

noise in an artic environment [5].

- The soundscapes of recreational open water marine activities, and their impact on indigenous bald eagle population[10].
- A study on airport noise and its effects on a population of cotton rats[9].
- The effects of the disturbance caused by mining noise on elf calf behaviour[7].

These studies have in common the necessity to prove the link between the noise source of interest and any change in behavioral patterns observed. To achieve this, the noise contribution of the noise source of interest must be measured and considered against the overall soundscape levels, particularly any other anthropogenic sources, in addition to isolation noise effects from other pressures on animal behavior and psychology. It is in this area that ISRIE could provide most benefit to this set of applications

Radle [16] also considers the effect of anthropogenic noise on aquatic environments. Sources of interest include shipping [13], but also non-marine anthropogenic sources such as aircraft and transport infrastructure [15].

Whilst the marine environments provide a recording environment in which isolating anthropogenic noise can be simpler, Radle [16] notes that the other factors in proving a causal link between anthropogenic noise sources and a change in animal behavior are complicated by poorer understanding of marine creatures behavioral patterns.

1.3.3 Areas of current legislation

Planning and Policy Guidance: 24

PPG 24 [4] is applied to new developments to evaluate noise exposure. Four noise exposure categories (NECs) are defined for local authorities when evaluating applications for residential development near existing noise sources. These are termed bands A, B, C, and D, defined by a range of free-field noise levels, dependent on the category of the noise source source; road traffic, rail traffic, air traffic and mixed noise sources.

As the NEC categories have different noise level ranges depending on the type of the assessed noise environment, the soundscape must be characterised by the contribution of the different noise sources present in the soundscape.

At the moment, the contributing sources to a soundscape are not easily quantifiable with existing technology. Under current practice[8], if the soundscape type cannot be satisfactorily placed into categories, A to C, category D is used as a catch-all.

However, the mixed noise source category D should only be used if there are no individual dominant noise sources, considered to be if its level lies within 2 dB(A) of the average value.

ISRIE would allow acoustic consultants to objectively select the correct noise source category to asses the soundscape against.

Other factors complicating the selection of categories is the perceived quality of the soundscape. According to PPG 24 guidance, events that exceed 82 dB $L_{Amax,s}$ several times in any hour place the soundscape in category C, regardless of the overall sound levels. Planning permission is usually denied to proposals within category C. However, the morning chorus of birdsong can frequently exceed this threshold, yet many find this a positive aspect of soundscapes, and not a factor which should lead to the denial of planning permission.

ISRIE could allow these threshold exceeding events to be logged and classified, freeing the consultant from having to perform this step manually, enabling them to make a decision based on metrics alone.

British Standard 4142

BS 4142 [2] is the standard for assessing whether commercial and industrial noise emissions are likely to cause complaint from adjacent residential dwellings.

The noise level of the source under examination is calculated by measuring the specific noise level at the dwelling location and subtracting the background noise sources.

Correction constants and on-time factors may also be applied to arrive at a rating.

The magnitude of the difference between the rating level and the background noise level is related to the likelihood of complaint.

In practice, the measurement of the specific noise level can be difficult[8]. Recordings of the specific noise can be affected by other sounds within the soundscape, for instance the passing of traffic. This is commonly avoided by pausing the recording to avoid such interference.

The ability to differentiate mechanical plant noise from transport noise would benefit consultants as it would allow a continuous accurate sound level to be measured. Correction factors for the on-time to off-time ratio could also be calculated automatically.

1.4 Thesis aims and objectives

1.4.1 Overall aim

This thesis aims to demonstrate that the sources present in typical rural and urban soundscapes can be separated and localised in three dimensions, and that this can be achieved using a compact coincident microphone array. This aim will be met by meeting a series of objectives outlined below.

1.4.2 Objectives

Applications are to be explored where a viable system for performing blind source separation of signals could have a positive benefit on current practice. Particularly of interest are those soundscapes typical of the applications previously discussed, such as nuisance noise or ecological studies.

A review of current source separation methodologies is to be analysed to determine an appropriate direction for research in this area. Also to be reviewed are methodologies

that may improve or extend the performance of these current methods.

As current source separation methodologies tend to be focused on speech or musical separation, the applicability of a chosen methodology for typical test cases of the applications mentioned earlier in this chapter must be proven. This must be achieved using standard performance metrics, such that the performance of the separation is easily comparable to other work.

The development of a methodology for the practical separation of signals within a 3D soundscape is the main thrust of the research in this thesis. A methodology must be demonstrated that is capable of discriminating between potentially many sources at different 3D locations around the sensor array. These positions are not necessarily known *a-priori*.

1.5 Overview of thesis organisation

The beginning of chapter 1 discusses the potential application areas for ISRIE. These are split into regulatory and biodiversity areas. Current methodologies are discussed where applicable, with particular focus on shortcomings and areas with potential for improvement. The benefits of deploying ISRIE in these environments is then examined to identify areas where ISRIE has potential to offer significant improvements.

Chapter 2 provides a background review of literature in the field of source separation. Several key methodologies for the separation of acoustic signals are identified, and considered in greater detail. Consideration is then given to the methodology's applicability to the diverse requirements imposed by the potential application areas for ISRIE covered in this chapter.

The wavelet transform is used extensively in this work, as alternative to the Fourier transform. This is because no underlying periodicity is assumed in the time domain signal, and therefore no *a-priori* information is required to optimise a windowing functino. Chapter 3 provides the reader with a brief background in the theory of the wavelet

transform. This is then built upon with an introduction to the dual-tree complex wavelet transform (DTCWT), which is the specific incarnation of the wavelet transform implemented in this thesis.

Following the brief review of the underlying theory, development work on implementing the DTCWT in MATLAB, both using finite impulse response (FIR) filtering approach, and via the lifting wavelet transform, is presented.

Chapter 4 begins by identifying a suitable framework to develop a methodology for blind source separation of soundscapes, and examines the underlying assumptions of sparse source separation. A metric is chosen to measure the conformance of mixtures to these assumptions. Test case mixtures are then generated to represent typical applications discussed in this chapter. The test cases are then examined to test the applicability of sparse source separation to these mixtures. Separation is performed using an ideal binary mask filter, and the results are presented. These results are compared to existing published work on speech separation for comparison. The applicability of this sparse source separation is discussed for specific examples of applications.

Work on the development of an audio source separation algorithm suitable for deployment in three dimensional soundscape environments is presented in chapter 5. This is the primary novel contribution in this thesis.

Performance metrics are defined to prove the validity of the separation performance, and to allow comparison of the method developed in this chapter with other works.

The development draws principally upon three main areas. The first of these areas is the somewhat contritely named DUET (degenerate unmixing and estimation technique) algorithm for the separation of ω-disjoint audio signals using a stereo microphone array. This is coupled with an algorithm for directional audio coding (DirAC), which provides a mechanism for extracting directional information for ω-disjoint audio mixtures in a three dimensional environment using an ambisonicorthogonal coincident microphone array.

An algorithm combining these concepts is developed, which provides for a means of

separating three dimensional soundscapes by the application of a directional binary mask.

By combining this algorithm with a shift invariant DTCWT, which allows signal phase information within the wavelet domain to be accessed using the same mathematical tools as the short-time Fourier transform (STFT), separation with improved performance of a signal-to-distortion metric compared to the STFT is achieved.

Results are presented for sources recorded under anechoic conditions. This recording environment is used to provide definitive metrics for the performance of the algorithm. The sensor array's effect on the directional sensitivity is considered, and results are compared to the ideal model.

Chapter 6 details work on estimating the direction of arrival of sources contained within the soundscape. The concept of using a peak detection algorithm applied to a histogram used in other works is extended into three dimensions. An approach using spherical coordinates to define a regular histogram is considered, and its merits and drawbacks discussed. A geodesic histogram describing the surface of a sphere is developed to overcome the main shortcomings of the previous approach, and program code to generate an arbitrary resolution histogram of this form is included. An alternative clustering approach using a dynamic self learning plastic self organising map (PSOM) of neurons is considered. This is is improved for this application by transforming the algorithm from a Euclidean space to a spherical surface. Further improvements to this model are suggested.

The final chapter, 7, summarises the achievements made for each section of this work, and considers the proposed system. Suggestions are made to identify areas where further research would be beneficial in realising ISRIE.

1.6 A summary of areas of novel research

A brief description of the main points of novelty contained in this work are contained in the list below:

- The review and identification of sound monitoring applications where improvements on current methodologies can be achieved. *Chapter 1*
- The analysis of ω-disjoint sparseness in typical soundscapes. *Chapter 4.*
- Demonstrating the applicability of separation based on time-frequency binary masking for non-speech signals. *Chapter 4.*
- Development of a separation algorithm for three dimensional soundscapes based on directional binary masking. *Chapter 5.*
- The application of the dual-tree complex wavelet transform (DTCWT) to audio signal processing *Chapters 3, 4 and 5.*
- The development of a spherical histogram using a geodesic grid. *Chapter 6.*
- Research into using clustering using a PSOM. *Chapter 6.*
- Extension of the plastic self organising map (PSOM) from Euclidean space to a spherical surface. *Chapter 6.*

Publications and conference proceedings arising through the course of this research are listed in Appendix D, which also contains copies of the published papers.

1.7 Chapter Bibliography

[1] S. Bison. Natural soundscape monitoring in yellowstone national park, december 2005 - march 2006. Grand Teton National Park soundscape program Report no: 200601, September 2006.

[2] E. committee committee. *BS 4142 Method for rating industrial noise affecting mixed residential and industrial areas.* BSI Standards, 1997. ISBN 0 580 28300 3.

[3] B. Defreville, F. Pachet, C. Rosin, and P. Roy. Automatic recognition of urban sound sources. In *Audio Engineering Society 120th Convention*, number 6827, May 2006.

[4] Dept. of Environment. *Planning Policy Guidance 24: Planning and Noise*. Stationery Office Books, October 1994. ISBN 9780117529243.

[5] M. Dorrence. Effects of snowmobiles on white-tailed deer. *Journal of wildlife management*, 39(3):563–569, 1975.

[6] S. Dyne, G. Y. Tian, and D. Chesmore. Isrie - instrument for soundscape recognition, identification and evaluation. EPSRC funding application, March 2006.

[7] G. Hompland. Elk responses to simulated mine disturbances in south-east idaho. *Journal of wildlife management*, 49(3):751–757, 1985.

[8] C. Karatsovis and S. Dyne. Instrument for soundscape recognition, identification and evaluation: An overview and potential use in legislative applications. In *Institute of Acoustics Spring Conference: Widening Horizons in Acoustics*, volume 30, page 602608, April 2008.

[9] L. Kavaler. *Noise: The new menace*. The John Day Company, New York, 1975.

[10] R. Knight. Responses of wintering bald eagles to boating activity. *Journal of wildlife management*, 48(3):999–1004, 1984.

[11] B. L. Krause and S. Gage. Seki natural soundscape vital signs program report. Technical report, Michigan State University (MSU), MI, 2003.

[12] R. C. Maher. White paper: Obtaining long-term soundscape, 2004.

[13] T. Norris. The effects of boat noise on the acoustic behaviour of humpback whales. In *Proc. of the Acoustic Society of America*, volume 96, page 3251, 1994.

[14] *The analysis and protection of natural soundscapes in national parks*, 2002. NRSS, WASO.

[15] J. Quinn. Whale trapped in firth of forth by traffic noise. Home News, March 1997.

[16] A. L. Radle. The effect of noise on wildlife: A literature review. World Forum for Acoustic Ecology Online Reader, 1998.

[17] A. Skånberg and E. Öhrström. Adverse health effects in relation to urban residential soundscapes. *Journal of Sound and Vibration*, 250(1):151–155, 2002. ISSN 0022-460X. URL `http://www.sciencedirect.com/science/article/B6WM3-4576DP8-21/2/eef9d3248819a943ab50b490ed7e094a`.

[18] D. Waugh. Environmental quality objectives for noise in relatively quiet areas and its potential impact on the mining and quarrying industry. *Extractive Industry Ireland Journal*, 2002.

Chapter 2

Review of signal separation literature

Contents

Separating mixtures of signals, audio or otherwise, is a long-standing problem. The task of separating audio sources was first described as the *cocktail party problem* [7]. Signal separation tasks covers a great variation in recording, environmental, and signal characteristics. This chapter aims to provide a review of the terminology and models used to frame the separation problem, and to identify metrics used to quantify performance of separation algorithms.

A review of literature identifying existing separation algorithms follows, with the merits of each considered based on published performance metrics. The chapter concludes with an evaluation of the separation requirements of ISRIE, and the selection of a method on which to base the development of a separation algorithm capable of source separation of a 3D soundscape.

2.1 Terminology overview

2.1.1 Separation type

Separation with *a-priori* knowledge

This type of separation assumes detailed prior knowledge of the signal of interest or environmental mixing parameters are known in advance. A typical application would be the separation of a musical ensemble playing from a known score.

Semi-blind source separation

Semi-blind separation assumes some knowledge of signal features in advance, which can be exploited to aid in separation. An example of a typical application would be the removal of wind noise from a recording using knowledge of the low frequency nature of the wind noise to help the separation.

Blind source separation

In theory, no prior knowledge of the signals or mixing environment is assumed. In practice, this is unattainable, as all the methods discussed in this chapter impose some assumptions implicitly or explicitly, either on the number of sources present, statistical properties of the sources, or the mixing environment. In general, a weak assumption is imposed on the sources. This is usually an assumption that is likely to be met by a particular application, or where occasional non-compliance will still lead to an acceptable separation.

2.1.2 The mixing model

The mixing environment is the medium in which the signals from the audio sources propagate between source and sensor. This is usually described as one of three linear models, each model incorporating an increasing degree of sophistication, realism, and complexity.

The mixing model is the mathematical framework used to formally describe the mixture of the sources observed at the sensors through the mixing environment. The models here are those that are introduced in [16], and are described below.

A set of T observations, $\mathbf{S}$, are made of M sensors at time intervals τ:

$$\mathbf{S} = [(s_0), \ldots, (s_T)] = \begin{bmatrix} s_1(0) & s_1(\tau) & s_1(2\tau) & \cdots & s_1(T\tau) \\ s_2(0) & s_2(\tau) & s_2(2\tau) & \cdots & s_2(T\tau) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_M(0) & s_M(\tau) & s_M(2\tau) & \cdots & s_M(T\tau) \end{bmatrix} \tag{2.1}$$

is made of a linear mixture of N source signals:

$$\mathbf{X} = [(x_0), \ldots, (x_T)] = \begin{bmatrix} x_1(0) & x_1(\tau) & x_1(2\tau) & \cdots & x_1(T) \\ x_2(0) & x_2(\tau) & x_2(2\tau) & \cdots & x_2(T) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N(0) & x_N(\tau) & x_N(2\tau) & \cdots & x_N(T) \end{bmatrix} \tag{2.2}$$

These sources $\mathbf{X}$ are subject to a linear mixing environment, $\mathbf{A}$ between the sources and the sensors:

$$\mathbf{A} = [(a_1), \ldots, (a_T)] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \tag{2.3}$$

where a_{ij} is the environmental mixing parameter between source j and sensor i. This leads to:

$$\mathbf{S}(t) = \mathbf{A}(t) \star \mathbf{X}(t) + \epsilon(t) \tag{2.4}$$

where ϵ represents a noise term, and $\star$ represents a linear operator specific to the mixing environment being modelled. The mixing environments commonly modelled are discussed in the following section.

2.1.3 The mixing environment

Instantaneous propagation

The instantaneous model is the simplest of the three models. The propagation path from source to sensor is assumed to be direct, and the propagation delay is ignored, with the environmental mixing parameter for each path being modeled as an attenuation. The operator $\star$ in equation 2.4 is matrix multiplication.

Equation 2.5 describes an instantaneous mixing model for N sources x_n observed at M sensors s_m. A two-source, two-sensor setup is shown in figure 2.1(a):

$$\begin{bmatrix} s_1(t) \\ \vdots \\ s_M(t) \end{bmatrix} = \begin{bmatrix} a_{(1,1)}(t) & \cdots & a_{(1,N)}(t) \\ \vdots & \ddots & \vdots \\ a_{(M,1)}(t) & \cdots & a_{(M,N)}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} \tag{2.5}$$

Anechoic propagation

The anechoic model builds on the instantaneous model, incorporating the propagation delay between each source and sensor. The operator $\star$ from equation 2.4 for this model is matrix convolution.

This is shown in figure 2.1(b), and is described by equation 2.6:

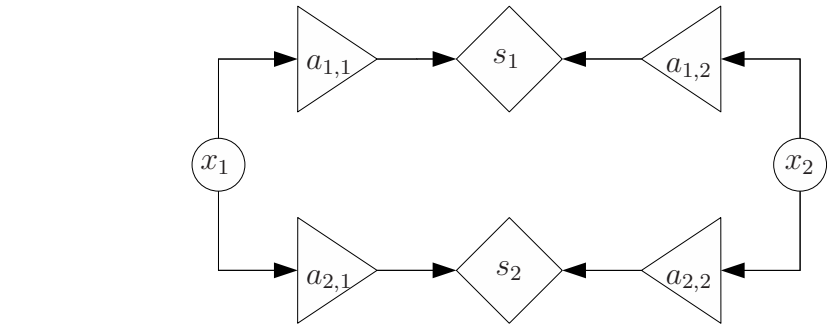
$$\begin{bmatrix} s_1(t) \\ \vdots \\ s_M(t) \end{bmatrix} = \begin{bmatrix} a_{(1,1)}(t-\delta_{(1,1)}) & \cdots & a_{(1,N)}(t-\delta_{(1,N)}) \\ \vdots & \ddots & \vdots \\ a_{(M,1)}(t-\delta_{(M,1)}) & \cdots & a_{(M,N)}(t-\delta_{(M,N)}) \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} \tag{2.6}$$

where δ is the propagation delay.

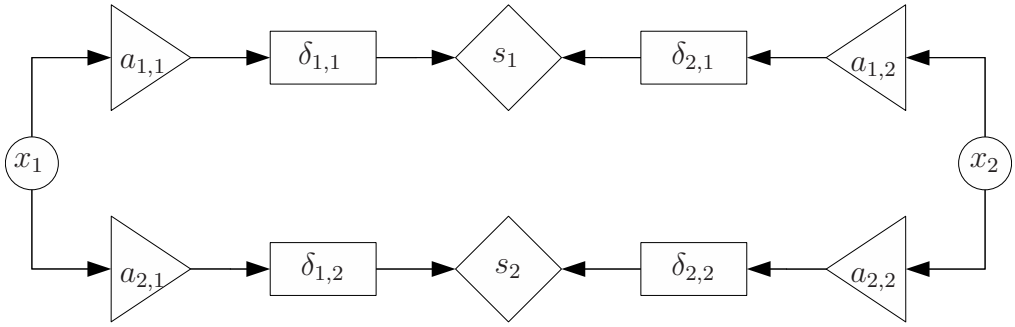
Echoic propagation

The most complex linear model, echoic propagation allows multi-path propagation from each source to each sensor. Each path $P_p(m,n)$, equation 2.7 between source n and sensor n is modelled as an attenuation and a delay, as the convolutive model. The contribution of a source n to sensor m is a function of all paths between them. The operator $\star$ from equation 2.4 for this model is multi path convolution. This is shown in figure 2.1(c), and is described by:

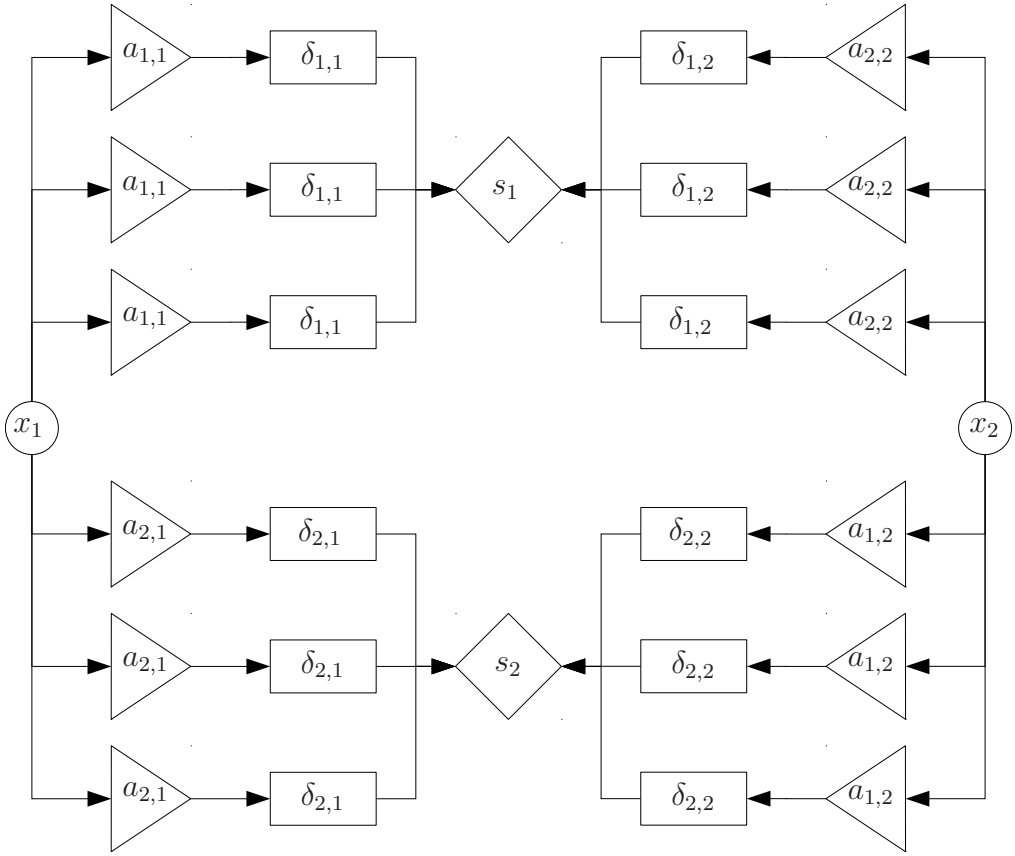
$$P_p(m,n) = a_{(m,n)}(t-\delta_{(m,n)}) \tag{2.7}$$



(a) Instantaneous mixing model



(b) Anechoic mixing model



(c) Echoic mixing model

Figure 2.1: Modeling the propagation environment

Where $n = 1 \rightarrow N$, $m = 1 \rightarrow M$, and N is the number of sources, M the number of sensors. $P_p(n, m)$ is path p between source n and sensor m.

$$\begin{bmatrix} s_1(t) \\ \vdots \\ s_M(t) \end{bmatrix} = \begin{bmatrix} F\{P(1,1)\} & \cdots & F\{P(1,N)\} \\ \vdots & \ddots & \vdots \\ F\{P(M,1)\} & \cdots & F\{P(M,N)\} \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} \tag{2.8}$$

where FP is a function of P.

2.1.4 Sources to sensors ratio

Equations 2.5, 2.6, and 2.8 in the previous section describe the mixture of N sources observed at M sensors mixed in differing environments. The ratio of sources N to sensors M is a constraint on many algorithms used for source separation. The ratio is referred to as:

Under-determined: Number of sensors M greater than the number of sources N.

Even-determined: Number of sensors M equal to the number of sources N.

Over-determined: Number of sensors M less than the number of sources N.

2.1.5 Separation tasks

An attempt to classify the tasks involved in a complex source separation scenario is considered in [22]. Here Vincent and and [22] notes that typical blind source separation applications may involve greatly differing mixing environments, which will affect the performance of the each algorithm differently, leading to different algorithms performing optimally in a given environment. Also noted is that aside from performing blind source separation, the algorithm may also be required to estimate the number of sources, their locations, or even adapt to changing numbers and location of sound sources. Vincent and and [22] suggests that the task of blind source separation can be

classified in terms of objectives, of which a summary of the main tasks are shown in figure 2.2.

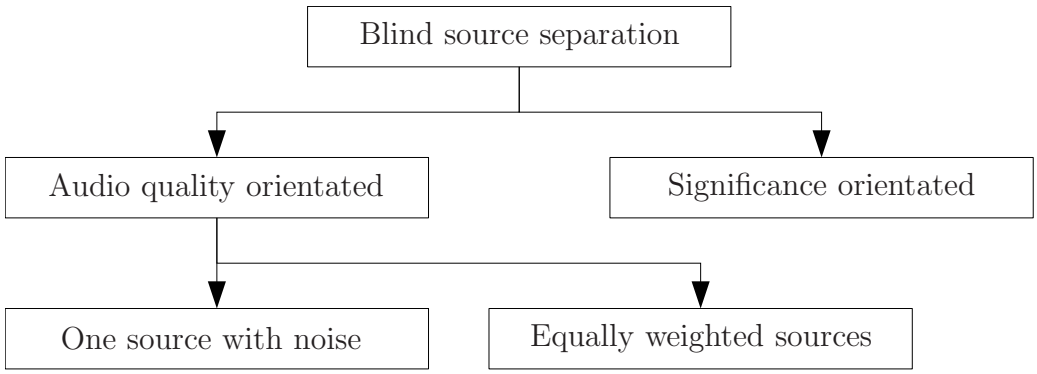


Figure 2.2: Topology of blind source separation tasks

The aim of audio quality oriented tasks is to extract the original audio without degradation of the source quality. This is further divided into two subclasses; the aim is either to extract one source and treat all other sources as noise, or to treat all sources as equally important and extract all sources without degradation to any.

Significance oriented blind source separation aims to extract features of the sources to give high or low level descriptions of each source. Some examples of this approach are: matching signal features against a signal dictionary, identifying the tones in a musical composition for automatic score rendering, or text transcription of speech signals. Provided that the features necessary for the task are extracted, the quality of the extracted sources is irrelevant; indeed, the extraction of an estimate of the original sources is unnecessary.

2.2 Performance metrics

Metrics are provided in [10] that allow the performance of blind source separation algorithms to be compared objectively, rather than relying on subjective human perception to determine the quality of estimated audio sources.

Gribonval et al. [10] note that the most commonly used benchmark for assessing the performance of a source separation algorithm is the signal to interference (or noise)

ratio (SIR / SNR) gain. Equation 2.9 is the definition of SIR used in this work.

$$SIR(s_i) = 10\log_{10}\left(\frac{\|x_i\|^2}{\sum_j \|x_j\|^2}\right) \qquad \forall i \neq j \tag{2.9}$$

This leads to the notion of SIR_{gain}, which is the improvement in SIR as a result of the separation process. Gribonval et al. note that SNR does not fully describe the performance of a particular algorithm. Algorithms can introduce distortions into the estimated sources, either as interference from other sources, as additive noise, or as non-linear artifacts from the algorithm itself. Gribonval et al. propose a measure for the impact of any particular algorithm, the signal to distortion ratio (SDR), equation 2.10

$$SDR(x_i) = 10\log_{10}\left(\frac{|\langle \hat{x}_i, x_i\rangle|^2}{\|\hat{x}_i\|^2 - |\langle \hat{x}_i, x_i\rangle|^2}\right) \tag{2.10}$$

where $\hat{x}$ is the estimate for x separated from the mixture.

Gribonval et al. also derive an upper performance limit applicable to any under-determined separation problem solved using a linear demixing approach. For at least one source estimate, the maximum SIR achievable for a mixture of normalised sources is given as equation 2.11

$$SIR \leq 10\log_{10}\left(\frac{M}{N-M}\right) \tag{2.11}$$

where N is the number of sources and M is the number of sensors.

2.3 Independent component analysis

Independent component analysis (ICA) is a method for finding a linear representation of non-Gaussian data which maximises the statistical independence of its outputs. The following introduction is based on two reviews of ICA algorithms [8, 11].

2.3.1 Model

Taking the mixing model described in equation 2.4, ICA aims to find an estimate for the mixing environment **A**. In the even-determined case, the inverse is then calculated, leading to an estimate $\hat{\mathbf{X}}$ of the original sources by applying a linear transform **W** to the observations **S**:

$$\begin{aligned} \mathbf{W} &= \mathbf{A}^{-1} \\ \hat{\mathbf{X}} &= \mathbf{WS} \end{aligned} \tag{2.12}$$

The noise term $\epsilon(t)$ in the model described in equation 2.4 is ignored here.

Minimising Gaussianinity

ICA can be performed using the observation that the sum of two non-Gaussian sources has a distribution closer to Gaussian than the two original sources. By finding a value for each column $\mathbf{w_i}$ in **W** that maximises the non-Gaussianinity of $\mathbf{w^T s}$, an estimate for the original component x can be found.

By using some measure of Gaussianinity, such as Kurtosis or neg-entropy, a search can be performed to find maxima representing the optimal solution.

Minimising mutual information

This approach to ICA uses a measure of the mutual information contained within estimates of the sources, and aims to minimise this. For statistically independent variables, the mutual information contained within two signals is zero.

Maximum likelihood estimation

This approach is based on maximising the function in equation 2.13. Hyvärinen and Erkki [11] note that this has been shown to be directly connected to the infomax

algorithm, which aims to maximise the information contained in the estimated sources.

$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} \log f_i \left(\mathbf{w}_i^T \mathbf{s}(t) \left(\right) \right) + T \log |\det \mathbf{W}| \tag{2.13}$$

2.3.2 Limitations of ICA

Whilst ICA is a well established technique, it is not well placed for application to ISRIE. Whilst there are a several variations on ICA that are applicable to echoic mixtures, these still rely on finding and applying a linear transform to the observed mixtures, and so the performance limitation of equation 2.11 applies. In the case of ISRIE, where there is the potential for many more sources than sensors, this performance penalty may be quite severe.

2.4 Sparse source separation

2.4.1 Two sensor model

DUET

The degenerate unmixing and estimation technique (DUET) algorithm [12] performs separation of an arbitrary number of sources from two mixtures using a spaced linear array of omni-directional sensors, and applying binary masking in the time-frequency transform domain. The assumption placed upon the sources made by this technique is that for every discrete point within the time-frequency domain, energy from at most one source is present. This condition is termed ω-disjoint orthogonality, and is formally expressed as:

$$S_i(\omega, \tau) S_j(\omega, \tau) = 0 \qquad \forall \omega, \tau, i \neq j \tag{2.14}$$

A real-time implementation of this algorithm is presented in [19]. In this implementation, for each time-frequency point, an estimation of the amplitude and delay mixing model parameters is made using the ratios of the two mixtures for every point in the time-frequency domain. In the original DUET algorithm, these parameters are used to form a 2D histogram, where the number of peaks determines the number of sources, and the peaks' location determines the mixing parameters for that source. Under ideal, no-noise conditions, this histogram would only have content in the same number of bins as sources, allowing perfect reconstruction of each source. In the implementation in [19], a maximum likelihood (ML) gradient search algorithm is used to determine the location of the peaks for a known number of audio sources.

The results presented with this implementation show a SNR improvement of 15dB in the case of an anechoic mixing environment, and a 5 dB SNR improvement in the echoic mixing case, for mixtures of 2 sources. This paper also presents some results to support the ω-disjoint orthogonality assumption in the case of mixtures of speech signals.

This work is continued in [18], where objective measures for the ω-disjoint orthogonality between signals are defined. Extensive results examining the validity of the assumption are then shown for voice recordings for mixtures of up to ten sources. The effect on the parameter estimation caused by violation of the ω-disjoint assumption are also considered, and can be seen in a spreading of the peaks in the previously described 2D histogram of delay and in the attenuation mixing parameters as a result of interference between signals at a time-frequency point.

Application of the DUET algorithm to direction-of-arrival (DOA) estimation is explored in [17], using the known geometry of the linear array to resolve a 2D half plane angle of arrival ($-\pi/2 \rightarrow \pi/2$). This is subject to the array geometry being suitably closely spaced, such that the relative delay between the sensors can be expressed as a phase shift, i.e. the distance between the two sensors is less than half a wavelength at the frequencies of interest.

Extensive further results for the separation of signals using DUET, for both voice from

voice, and voice from noise, are presented in [24]. Results are provided for combinations of up to ten sources using two mixtures. These results verify the the effectiveness of using the DUET algorithm for the separation of speech mixtures. Performance metrics for the technique used in the paper are also provided.

DUET with statistical assumptions

The assumptions made by DUET of ω-disjoint orthogonality is scrutinised in [3]. This notes that whilst DUET is successful in the separation of sources and the suppression of other interfering sources for large numbers of speech signals, artefacts are introduced into the recovered signals. Balan and Rosca [3] suggest analysing the ratio of mixtures using a statistical technique to better estimate the mixing parameters. The proposed approach removes the need for the sources to be ω-disjoint orthogonal, but imposes two additional constraints upon the signals.

1. Sources must be stationary in the short term, but must vary in frequency content over the long term.

2. For a given sampling window, signals are permitted to have gaps in their frequency content, but over the long term, each source must have energy in each frequency band.

DUET with harmonic assumptions

Another approach which can be considered to be based upon DUET for the special case of musical recordings is detailed in [23]. The approach analyses the ratio of mixtures calculated using the DUET algorithm to find area of the time-frequency domain that have a high probability of containing only one source. These areas are used as a bit mask in the time frequency domain, together with a set of harmonic masks based on the identified signal source areas. Results are presented for the performance of the algorithm, but as the focus of the paper is on distortion of recovered musical signals,

Woodruff and Pardo have used the signal to distortion ratio SDR instead of the SNR, thus a direct comparison of results in the literature is not possible.

TIFROM

The Time-Frequency Ratio of mixtures algorithm (TIFROM) [1] is another approach that exploits time-frequency sparseness. Unlike DUET, TIFROM requires only that the sources are ω-disjoint at a subset of points in the time-frequency domain, allowing the sources to overlap in the rest of the plane. The approach TIFROM takes is to estimate the mixing parameters for a particular source and use this to calculate an estimate of the source. This source is then removed from the mixture and the algorithm is recursively applied. In the over or evenly determined case, complete source separation is achieved. In the under-determined case, only partial blind source separation is achieved.

DESPRIT

The DESPRIT algorithm [14], extends DUET using the ESPRIT DOA estimation algorithm. DESPRIT separates an arbitrary number of sparsely echoic sources from two or more sensors arranged in a linear array. DESPRIT, like other extensions to DUET, relaxes the ω-disjoint orthogonality condition, and allows sources to overlap in some portions of the time-frequency domain.

Echoic separation is achievable with DESPRIT, although the number of echoic paths must be less than equal to half the number of sensors. This limits its application in echoic environments to special cases of either very large sensor arrays, or low numbers of echoic paths.

Beamforming

Building on the concept of separation in the time-frequency domain, [6] introduces an extension to simple binary masking to overcome the issues of tonal artefacts being introduced as a result of imperfect estimation, caused by either inaccurate esti-

mation of the mask, or the sources not being completely sparse in the transformed domain. The approach presented first performs a Fourier transform, and then calculates a time-frequency binary mask, for example by using the DUET algorithm. Rather than actually perform masking, the binary mask, together with estimates of the mixing parameters used to create the mask, as well as the time-frequency domain mixtures are fed into a beamforming stage. This is applied to the mixtures using the estimates of the mixing parameters to guide the separation using the beamformer.

A suggested further stage of enhancement on the separated signals is achieved by applying the previously calculated mask to the recovered source estimates. This is either in the form of a binary mask or a soft mask. In the former case, what has been achieved can be seen as an extension to binary masking where source components overlapping within the time-frequency domain are also filtered. In the case of the latter, tonal artefacts may be reduced by filtering the discontinuities in the filtering mask.

Results for both SIR and SDR are presented, using three microphones in a 2D linear array. The effects of varying the soft masking are briefly discussed in [6].

Other clustering approaches

An approach for the separation of under-determined sources using clustering in the time-frequency domain is presented in [4]. This approach assumes sparsity in the time-frequency domain. The algorithm first clusters by magnitude peaks in the time-frequency domain of each mixture, which provides an estimation of the direction of arrival for each source. An estimation is made for the delay of each source. Results are provided for both the echoic and anechoic cases for mixtures of speech and music, with results in each case comparable with one another.

Introduced in [5] is a method for under-determined BSS from two sensors, exploiting sparseness in the time-frequency domain. The approach segments the time domain observation into frames, on which a short-time Fourier transform (STFT) is performed. Bofill and Zibulevsky use a clustering algorithm to estimate the number of sources present in the mixtures, and also the mixing parameters. A linear programming oper-

ation is used to estimate the original sources.

Results using the algorithm for mixtures of up to six flute sources are given. Bofill and Zibulevsky note good separation in simple cases, but note that the performance of the separation achieved by the algorithm is dependent on the window size used for the STFT. The separation quality is affected by the window size as this alters the sparsity of the signals in the time-frequency domain. Bofill and Zibulevsky conclude that the application of this algorithm is limited by the complexity of the sources in the mixture and the sparsity achievable by optimising the STFT window used, rather than by the number of sources.

Hough transform

The approach presented in [13] partitions the input vectors into frames which are used to create a histogram. Image analysis using the Hough transform is then performed on the histogram to identify edges of dominant features in the histogram. This is used to estimate the mixing parameters and is used as the basis for source separation. This method is applicable for even-determined mixtures, although Lin et al. do note that it can be applied to achieve partial separation of under-determined mixtures.

2.4.2 Many-sensor model

Three-sensor binary mask

An extension to the two sensor model to allow use of three or more sensors is developed in [2]. The proposed method is able to perform separation of ω-disjoint orthogonal sources spaced in 3D rather than in the 2D half plane that the approaches using a two-sensor linear array are limited to. The algorithm presented here also removes the constraint of knowing *a-priori* details of the sensor array geometry. Instead, the maximum distance from a sensor to other sensors in the array is required for each member of the array.

The time-frequency domain observations for each sensor are normalised with reference to one of the sensors, designated the reference sensor, factoring into the normalisation process the distance from each sensor to the reference sensor, as well as the attenuation of the recording medium. The normalised mixture's signal power at each time-frequency point is now representative of the relative distance between the source and each sensor in the array. All the observation are now combined to form a vector. These vectors are clustered based on the squared distance from the reference sensor. All points belonging to a cluster are taken to belong to a source and a binary mask is applied to the reference sensor to provide estimates for each source.

B-format microphone

A system of blind source separation using four signals is introduced in [21]. The signals are the output of a B-format microphone system, and comprises orthogonal measurements of particle velocities, and sound pressure. Teramoto et al. note that, as these are measured at a coincident point, this method has the advantage of simplifying anechoic convoluted blind source separation problems to the instantaneous model, as it removes the measurement delay between sources. Teramoto et al. use the particle velocity as the basis for independent analysis, with the standard assumptions of independent and non-Gaussian sources. The algorithm performs over-determined separation and is capable of the separation of three sources from four observations.

Coincident array

Another solution that allows separation of sources, is presented in [15]. This technique also calculates an estimate of direction of arrival for each source. Mukai et al. propose using an eight microphone array, and use a frequency based ICA method for source separation. The mixing parameters estimated from the ICA algorithm are representative of a direction of arrival for a signal component. These are clustered to find estimates of the source locations, which are used to guide the ICA algorithm, providing one method of solving the permutation problem inherent in frequency domain ICA.

This method allows for sources to be moved during the separation, and provides results for the effect on the SNR such movement causes. The SNR gain achieved is reduced whilst sources are in motion, but the SNR gain returns to its former level once the sources are stationary again. No indication is given of the angular velocity of the source movement about the microphone array.

2.4.3 Binaural model

Frequency domain

An approach for under-determined separation of anechoic speech using sensors to replicate the binaural model is presented in [9] - i.e, by replicating the human auditory system. This approach assumes sparsity in the time-frequency domain. The inter-aural time difference (ITD), and the inter-aural level difference (ILD) are calculated. These are the equivalents of the delay and attenuation between sensors subject to a head-related transfer function (HRTF).

The ITD and ILD are used to calculate an estimate for the direction of arrival for each source in 2D. The HRTF allows the model to distinguish between the fore and aft half planes, giving 2D resolution for each. These directional estimates are used as the basis to distinguish between sources, and allows binary time-frequency masking to be performed on the mixture observed at one of the two sensors. The proposed method then performs a post-separation processing step by applying a filter weighted to better match the human auditory system to improve the processed sound quality. No numerical metrics are presented to allow comparison of the performance of this algorithm.

Time domain

A similar technique to the previous paper is presented in [20], applicable to under-determined mixtures of sources. The technique relies on the localisation of sources using a binaural model. The requirement for sparsity in the time-frequency domain

representation of sources is implied.

This method uses 128 band-pass filterbanks to achieve frequency separated time domain signals, rather than the transformed time-frequency coefficient of the Fourier transform. The ITD and ILD are then calculated for these time domain filtered signals. Direction of arrival estimates are then found using cross-correlation of the ITD. The location of the sources is assumed to be fixed. A binary mask is then created and applied to the time domain data.

The results shown for the case of a speech against a noise channel (a telephone ringing) show that in this simple case a binary mask is found close to the ideal. No indication is given of how the method performs in applications that contain sources with similar frequency components.

2.5 Chapter synopsis

This chapter contains an overview of separation terminology, metrics, and a review of a range of the separation algorithms. Many of the methods examined are optimised to specific tasks. Even those methods that are suitable to a range of application are not reported using a standard set of tests or metrics and may use differing microphone configurations, so a direct comparison between them is not in general possible. Therefore, to form a conclusion as to which method is best to use as the basis of development for separation within ISRIE, the methods must be considered in the light of the requirements of the ISRIE project, which were considered in the chapter 1.

2.5.1 ISRIE separation model

Model and environment

Due to the range of applications ISRIE may be applied to, there is no mixing model that will be consistently met. In an open field recording environment, the convolutive

model is applicable. However, if ISRIE is used in an urban canyon, the strongly echoic model may be applicable.

The mixing environment ISRIE may experience is also likely to vary, both between applications and during each one.

ISRIE Separation goal

The primary aim of the separation required by ISRIE is to preserve significant features of each source, whilst suppressing interfering sources, to aid the automatic recognition and identification of each source by subsequent stages. ISRIE can therefore be classified as a significance orientated separation task. However, as playback of the estimated sources by audio consultants may be required for verification or review, decomposition of the signal into a high level signal directory is undesirable. What is required is an estimation of each source of interest, balancing the suppression of interfering sources against preserving significant features of the original sources to guide recognition. This differs slightly from common applications that aim to faithfully estimate the original sources.

Depending on the specific use of ISRIE, both models of audio quality separation may be required; namely the preservation of one source with others regarded as noise, as well considering all noise sources equally.

2.5.2 ISRIE separation method

Considering the goals of ISRIE (see chapter 1), the formation of a binary mask based on a sparse representation of the sensor observations seems well suited to the task. Provided a suitable transform can be found for the sources, the preservation of significant features in the source estimation is likely to occur.

The DUET method appears to be successful, but its limitation to the 2D half plane is problematic. Its extension to 3D does overcome this, although the requirement

for a more complex microphone array is a drawback. Indeed, the concept of using a coincident array is attractive, as its compact form leads to easier deployment in the field. The commercially available ambisonics B-format microphones, such as the Soundfield ST-350, are highly compact whilst providing four channels of audio. The separation algorithms discussed in this chapter use ICA as the basis of separation. Developing a 3D binary masking approach based on a coincident array appears to offer the best solution for ease of deployment, whilst allowing separation based on the well proven premise of binary masking.

2.6 Chapter Bibliography

[1] F. Abrard and Y. Deville. A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources. *Signal Processing*, 85: 1389–1403, 2005. doi: 10.1016/j.sigpro.2005.02.010.

[2] S. Araki, H. Sawada, R. Mukai, S. Makino, et al. Underdetermined sparse source separation of convolutive mixtures with observation vector clustering. In *IEEE International symposium on circuits and systems*, page 4, May 2006.

[3] R. Balan and J. Rosca. Statistical properties of stft ratios for two channel systems and applications to blind source separation. In *2nd ICA and BSS Conference*, June 2000.

[4] P. Bofill. Underdetermined blind separation of delayed sound sources in the frequency domain. *Neurocomputing*, 55:627–641, 2003.

[5] P. Bofill and M. Zibulevsky. Blind separation of more sources than mixtures using sparsity of their shot-time fourier transfrom. In *International Workshop on independant component analysis and blind signal separation*, June 2000.

[6] J. Cermak, S. Araki, H. Sawada, and S. Makino. Blind source separation based on a beamformer array and time frequency binary masking. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[7] C. E. Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of Acoustical Society of America*, 25(5):975–979, 1953.

[8] S. Choi, A. Cich, H.-M. Park, and S.-Y. Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing - Letters and Reviews*, 6(1):1–57, January 2005.

[9] A. Favrot, M. Erne, and C. Faller. Improved cocktail-party processing. In *9th International conference on digital Audio Effects*, September 2006.

[10] R. Gribonval, E. Vincent, C. Fevotte, L. Benaroya, et al. Proposals for performance measurement in source separation. In *4th International symposium on independant component analysis and blind signal separation (ICA2003)*, April 2003.

[11] A. Hyvärinen and O. Erkki. Independent component analysis: Algorithms and applications. *Neaural Networks*, 13(4-5):441–430, 2000.

[12] A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '00*, volume 5, pages 2985–2988, 5–9 June 2000. doi: 10.1109/ICASSP.2000.861162.

[13] J. K. Lin, D. G. Grier, and J. D. Cowan. Feature extraction approach to blind source separation. In *IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, pages 398–405. IEEE Press, 1997.

[14] T. Melia. Underdetermined blind source separation in echoic environments using desprit. *EURASIP Journal on Advances in Signal Processing*, 2007(1):1–19, 2007. doi: 10.1155/2007/86484.

[15] R. Mukai, H. Sawada, S. Araki, and S. Makino. Real-time blind source separation and DOA estimation using small 3-d microphone array. In *International Workshop on Acoustic Echo and Noise Control*, pages 45–48, September 2005.

[16] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology*, 15:18–33, 2005.

[17] S. Rickard and F. Dietrich. Doa estimation of many w-disjoint orthogonal sources from two mixtures using duet. In *Tenth IEEE Workshop on Statistical Signal and Array Processing*, pages 311–314, 2000. doi: 10.1109/SSAP.2000.870134.

[18] S. Rickard and Z. Yilmaz. On the approximate w-disjoint orthogonality of speech. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, volume 1, pages I–529–I–532, 2002. doi: 10.1109/ICASSP.2002.1005793.

[19] S. Rickard, R. Balan, and J. Rosca. Real-time time-frequency based blind source separation. In *in Proc. of Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 651–656, 2001.

[20] N. Roman, D. Wang, and G. Brown. Speech segregation based on sound localization. *Journal of the Acoustic Society of America*, (114):2236–2252, 2003.

[21] K. Teramoto, T. Khan, and S. I. Torisu. Acoustical blind source separation based on linear advection. In *SICE annual conference*, pages 1–118, 2007.

[22] E. Vincent and X. R. and. A tentative typology of audio source separation tasks. In *4th International symposium on independent component analysis and blind signal separation(ICA2003)*, 2003.

[23] J. Woodruff and B. Pardo. Using pitch, amplitude modulation, and spatial cues for separation of harmonic instruments from stereo music recordings. *EURASIP Journal on Advances in Signal Processing*, 2007(1):1–10, 2007. doi: 10.1155/2007/86369.

[24] O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE journal on signal processing*, 52(7):1830–1847, July 2004. doi: 10.1109/TSP.2004.828896.

Chapter 3

Time-Frequency Transformations: Fourier and Wavelets

Contents

3.1 Introduction

This chapter provides an introduction to the concept of Fourier and wavelet transformations. An overview of the Fourier transform is provided, although familiarity is assumed. A synopsis of its strengths and limitations for analysis of time-variant signals is provided. An overview of the continuous, discrete and dual-tree complex wavelet transforms, being a relatively recent development, are provided in more detail. Important results are noted, although fully rigorous mathematical proofs are not provided and are beyond the scope of this chapter.

3.2 Fourier transform

The Fourier transform, equation 3.1 has long been used by engineers, scientists and mathematicians as a tool to examine the frequency spectra of signals.

$$F(\omega) = F\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{(-i\omega t)}dt \tag{3.1}$$

where $f(t)$ is a function of time.

$F(\omega)$ is an unbounded continuous function of frequency, but provides no time resolution. For time-variant signals, or signals containing transients, it is desirable to examine a signal over a time-interval of interest. For discretely sampled signals of length N, this is achieved using the discrete Fourier transform (DFT), equation 3.2.

$$F[k] = F\{f[k]\} = \sum_{n=0}^{N-1} f[n]e^{\left(-i2\pi k\frac{n}{N}\right)} \quad \forall \quad k = 0, \ldots, N-1 \tag{3.2}$$

Given the sampling frequency F_s, $F[k]$ now comprises k components representing uniformly distributed frequency bands between 0 to $\frac{Fs}{2}$ Hz, localised in time over the N samples. For a constant sampling frequency, to increase time resolution, fewer samples may be used in the calculation of $F[k]$, at the expense of frequency resolution.

Likewise, frequency resolution may be increased at the expense of time resolution.

The DFT implies a periodicity in the signal under examination, and wideband artefacts are introduced in $F[k]$ if this assumption is broken. In time-variant signals, where in general the signal does not display periodicity, the signal is combined with a windowing function $\nu[n]$, such as the Hamming window, to reduce the artefacts due to the forced periodicity.

For long duration input signals where good time resolution is required, the short-time Fourier transform (STFT), equation 3.3, can be applied. This calculates many DFT frames of length N at time intervals m. Depending on the window used, perfect reconstruction via the inverse transform can be achieved provided that sufficient overlap between frames exists $m \leq \frac{N}{2}$. For the hamming window, this is achieved where $m = \frac{N}{2}$

$$F[k,m] = STFT\{f[k]\} = \sum_{n=0}^{N-1} f[n-m]\nu[n-m]e^{\left(-i2\pi k\frac{n}{N}\right)} \quad \forall \quad k = 0,\ldots,N-1 \tag{3.3}$$

Using $F[k,m]$, it is possible to examine the frequency spectra at a chosen resolution. Choosing the correct length for m is critical in obtaining an appropriate representation of the signal. For high-frequency transient signals, a small m is best, whilst for constant tones a large m would be sufficient. Therefore, selecting m either requires *a-priori* knowledge of the signal's properties, or in cases where this is infeasible, selecting a value of m to provide a compromise.

3.3 The Continuous Wavelet Transform - CWT

The continuous wavelet transform (CWT) of a time domain function $f(t)$ is defined in equation 3.4.

$$\gamma(a,b) = \int_{-\infty}^{\infty} f(t)\psi^*_{(a,b)}(t)dt \tag{3.4}$$

where $\psi_{(a,b)}$ is derived from the basis wavelet $\psi(t)$ using equation 3.5

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.5}$$

where a is a scaling factor and b is a translation factor.

To be considered a wavelet function, $\psi(t)$ must conform to several conditions. One of these defined here, the admissibility condition, is discussed due to the properties it implies a wavelet basis function must display.

3.3.1 Admissibility condition

It has been shown [11] that where $\psi(t)$ is a square integral function, i.e. $\psi(t) \in \ell^2$, if the admissibility condition, equation 3.6, is met then $\psi(t)$ can be used to analyse (equation 3.4) and reconstruct (equation 3.9) signal $f(t)$ without loss of information.

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{3.6}$$

where $\Psi(\omega)$ denotes the Fourier transform of $\psi(t)$

This implies two important properties of the wavelet basis function [15]. Equations 3.7 and 3.8 show conditions that are implicitly met if the admissibility condition is met.

$$\Psi(0) = 0 \tag{3.7}$$

$$\int \psi(t)dt = 0 \tag{3.8}$$

Equation 3.7 implies that the spectrum of the wavelet basis is similar to that of a band-pass filter, whilst equation 3.8 implies the wavelet basis is oscillatory.

3.3.2 The Inverse Continuous Wavelet Transform

It is possible to achieve perfect reconstruction and obtain the original time domain signal $f(t)$ from $\gamma(a, b)$

$$f(t) = \int\int \gamma(a,b)\psi_{(a,b)}(t)db\ da \tag{3.9}$$

3.4 The Discrete Wavelet Transform - DWT

For the wavelet transform to be of any practical benefit, it must be implementable efficiently in the discrete world of digital computing. The continuous wavelet transform described in the previous section is a continuous basis function, translated and scaled by continuous functions convolved with a continuous signal. This leads to the requirement to calculate an infinite number of wavelet transforms.

Rather than perform numerical solutions to an analogue problem, instead a form of wavelet transform for discrete data is preferred, the discrete wavelet transform (DWT). Of the innumerable introductions to the subject, [5, 15] offer excellent introductions to the discrete wavelet transforms from an engineering perspective. A brief description of the derivation of the DWT, based on these sources is presented, as it offers some important insights into the usefulness of the DWT.

To provide a practical implementation for the DWT, several issues need addressing:

- Reduction of the number of computations to a finite level.
- Discretising the wavelet basis function whilst maintaining perfect reconstruction.
- Discretising the scaling and translation factors applied to the wavelet basis.

3.4.1 Discretising scaling and translational factors

In [3], a discrete form of the wavelet basis, equation 3.5 is introduced.

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{2^a}}\psi\left(\frac{t - 2^a b\tau}{2^a}\right) \tag{3.10}$$

where scaling factor a and translation factor b are integers ($a, b \in \mathbb{Z}$), and τ is the sampling period.

This representation gives dyadic sampling of the time and frequency axis, see figure 3.1.

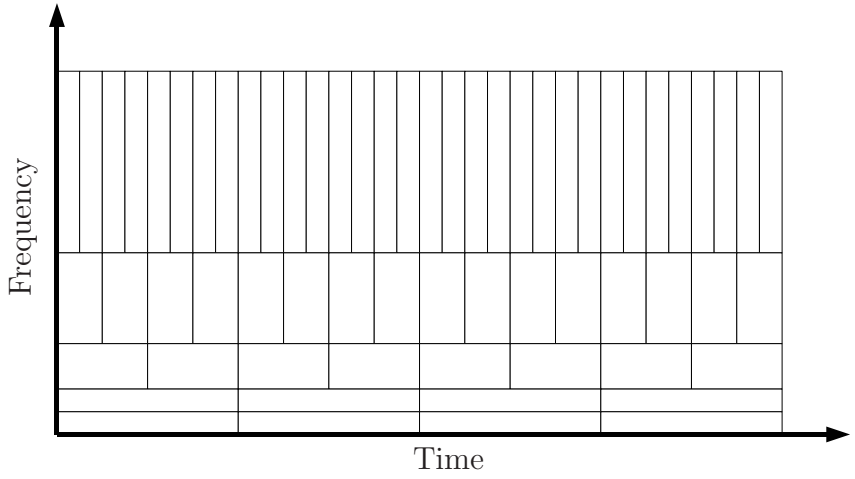


Figure 3.1: Time frequency sampling achieved using dyadic sampling. The time-frequency area for each sample is constant

This is a useful property of the DWT. Dyadic sampling provides an inherently logical time-frequency resolution across the spectrum, i.e, good time domain localisation with poor frequency resolution for high frequencies, and conversely poor time resolution with good frequency resolution for low frequencies. This contrasts with the STFT, where the time and frequency resolution is constant and defined by the length of a windowing function that must be applied to non-periodic data prior to the FFT.

The discrete wavelet transform is now given by

$$\gamma_{(a,b)} = \sum_{a,b} f(t)\psi_{(a,b)}(t) \tag{3.11}$$

[3] shows that this achieves perfect reconstruction providing

$$C \left\| f(t) \right\|^2 \leq \sum_{a,b} \left| \langle f, \psi_{(a,b)} \rangle \right| \leq D \left\| f(t) \right\|^2 \tag{3.12}$$

where $C > 0$, $D < \infty$ and C and D are independent of $f(t)$

If $C = D$, the wavelets are orthonormal, and the inverse wavelet transform can be given by:

$$f(t) = \sum_{a,b} \gamma_{(a,b)}(t) \psi_{(a,b)}(t) \tag{3.13}$$

It is noted that an orthonormal wavelet basis is not necessary for reconstruction, but allows decomposition and reconstruction to be performed with the same wavelet basis. Wavelets are orthonormal if the condition in equation 3.14 is met

$$\int \psi_{(a,b)} \psi^*_{(m,n)} \begin{cases} 1 & | \quad a = m, \quad b = n \\ 0 & | \quad otherwise \end{cases} \tag{3.14}$$

Equation 3.11 and 3.13 now provide a transform requiring the scaling and translation of the wavelet basis ψ at discrete intervals. However, note that ψ and $f(t)$ remain continuous functions, and a is unbounded. In practice, b will be bound by the length of the signal $f(t)$

3.4.2 Bounding the scaling factor a

The admissibility condition, equation 3.7, implies that wavelets have a spectrum similar to that of a bandpass filter. The scaling factor a can be seen in equation 3.10 to be a stretching of the wavelet basis in the time domain. The effect of this is the compression of the wavelets spectrum in the frequency domain, each increment in a corresponding to a halving in the frequency bandwidth.

It is possible to view the DWT as expressed in equation 3.11 as the summation of the outputs of a *constant-q* filter bank, made up of an infinite number of filters, each

having half the bandwidth of the previous filter.

As $a \mapsto \infty$, the bandwidth covered by $\psi_{(a,b)} \mapsto 0$ Hz, and the information contained in each band is less. By combining all the information contained in the wavelets from some value of a to ∞, a bound can be imposed on a. The value of a is chosen as a matter of design to give an acceptable level of detail in the time-frequency domain.

A constant-Q filterbank implementation with the scale a bound to 3 is shown in figure 3.2(a).

This is effectively a lowpass filter applied to the original signal, and can also be expressed as the inverse DWT up to scale a, equation 3.15

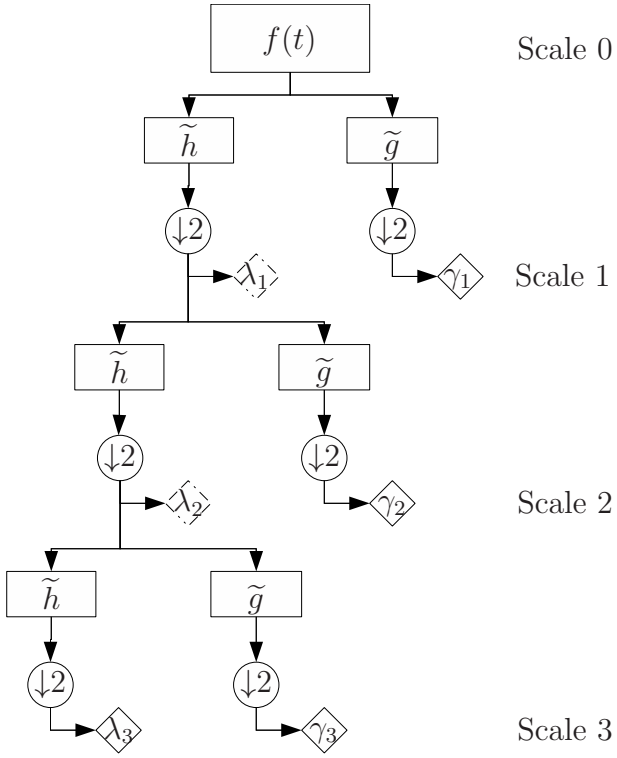
$$\varphi(t) = \sum_{a,b}^{\infty,b} \gamma_{(a,b)}(t), \psi_{(a,b)}(t) \tag{3.15}$$

3.5 The Dual-Tree Complex Wavelet Transform - DTCWT

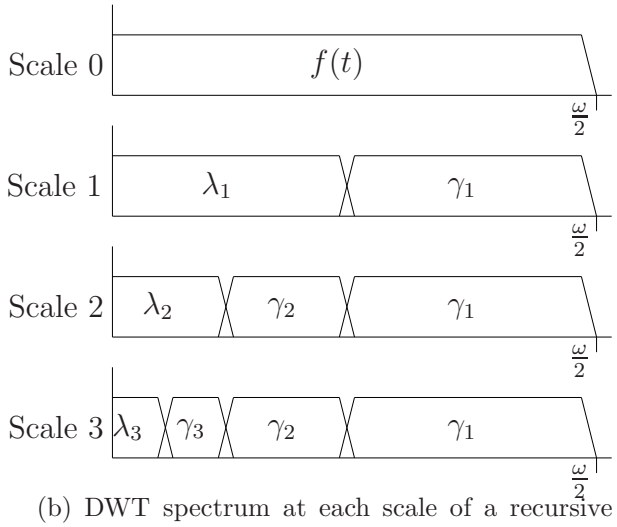
The previous section provides an efficient implementation for the DWT. However, for use with the application discussed in chapters 4 and 5, there are some issues to be resolved.

These issues all have their routes in the shift variance of the wavelet transform. Shift variance causes a radical perturbation in the ratio between adjacent wavelet coefficients following a time-shift in the input signal. This can be seen in figure 3.3. This leads to two problems if the DTCWT is to be used as a direct replacement for the Fourier transform. These are:

- Inaccessible phase information. It is not apparent how to extract phase information from the DWT if the wavelet representation of the input signal changes as a result of a time shift in the input signal.



(a) DWT implemented using a recursive constant-Q filterbank



(b) DWT spectrum at each scale of a recursive constant-Q filterbank

Figure 3.2: Spectral properties of the discrete wavelet transform

- Aliasing. The DWT causes aliasing, which is cancelled by the inverse DWT. This cancellation only occurs provided the wavelet coefficients are unaltered, i.e. without filtering in the wavelet domain.

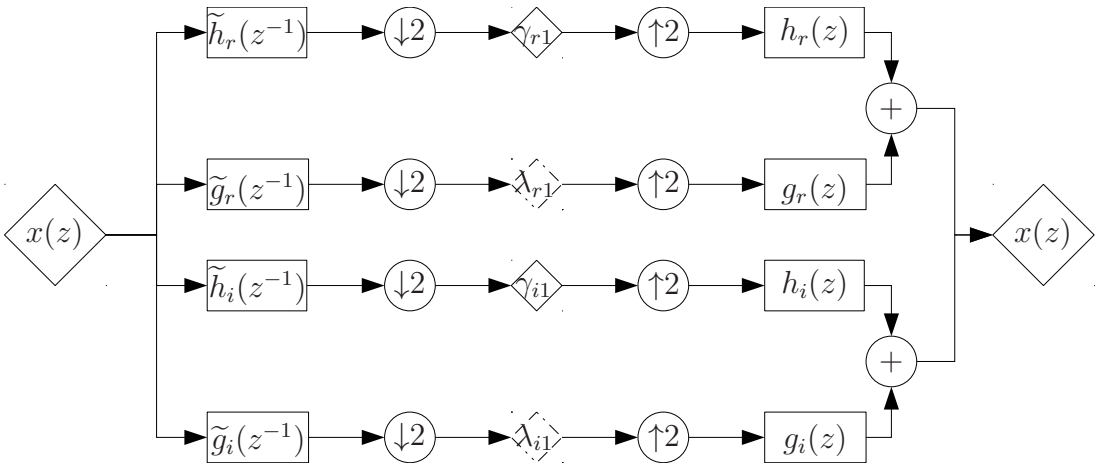


Figure 3.3: The Dual-Tree complex wavelet transform (DTCWT)

A solution to these problems is to extend the wavelet transform to the DTCWT, proposed in [8] as a tool for image processing. The underlying premise is to perform two sets, or trees, of the DWT, with 90° phase difference between the two trees, providing one real set of coefficients and one set of imaginary coefficients, which combined provide a complex DWT.

At the expense of twice the computational overhead, the DTCWT provides a transform with approximate shift invariance and provides a solution to the aforementioned problems. The shift invariant properties of the DTCWT introduced in [8] are investigated further in [9], providing an analysis of the DTCWT's shift-invariant performance.

In [6], a new form of the filters required for the DTCWT is introduced, based on orthogonal rather than bi-orthogonal filters, allowing shorter filter lengths to be used for equivalent performance.

In [7], this concept of designing orthogonal filters for the DTCWT is enhanced further. This implementation is based around a 2-times oversampled linear-phase symmetric low-pass filter (a quadrature mirror filter, QMF) of length $4n$ $H_{L2}(z)$, and sub-sample by a factor of 2 to give a filter with a delay of 1/2 sample. $\widetilde{h}$ and h are then defined as in equation 3.16, i.e, even and odd powers of $H_{L2}(z)$ form $\widetilde{h}$ and h respectively.

$$H_{L2}(z) = \widetilde{h}(z^{-2}) + z^{-1}h(z^2) \tag{3.16}$$

This gives filter $\widetilde{h}(z^{-1})$ a delay of $\frac{1}{4}$ of a sample. As H_{L2} is an even length filter, the time reversal of $\widetilde{h}(z^{-1})$, $\widetilde{h}(z)$ has a delay of $\frac{3}{4}$ of a sample, and can be used for the imaginary tree, giving a delay between the two trees of $\frac{1}{2}$ a sample, and providing the requisite 90° phase shift. It is the filter implementation given in [7] that is used for the DTCWT in this work.

Finally, an overview of the subject of the DTCWT including alternative implementations, as well as applications is provided in [10]

3.5.1 Q-shift filter relationships

The Q-shift filter implementation of the DTCWT is named after the quadrature mirror filter from which they derive. The required filters, $\widetilde{h}_r(z^{-1})$, $\widetilde{g}_r(z^{-1})$, $h_r(z)$, $g_r(z)$, $\widetilde{h}_i(z^{-1})$, $\widetilde{g}_i(z^{-1})$, $h_i(z)$, and $g_i(z)$, see figure 3.3, are derived from one another [7]. These relationships are described formally in this section. The notation used here differs from the original, as this thesis follows the convention set by [4], describing the analysing filter as a function of z^{-1} rather than of z.

The conditions for perfect reconstruction are the same as for the DWT. The conditions apply to both real and imaginary trees independently. The conditions for perfect reconstruction are shown in equation 3.17

$$\begin{aligned}
\widetilde{h}_r(z^{-1})h_r(z) + \widetilde{g}_r(z^{-1})g_r(z) &= 2 \\
\widetilde{h}_r(-z^{-1})h_r(z) + \widetilde{g}_r(-z^{-1})g_r(z) &= 0 \\
\\
\widetilde{h}_i(z^{-1})h_iR(z) + \widetilde{g}_i(z^{-1})g_i(z) &= 2 \\
\widetilde{h}_i(-z^{-1})h_i(z) + \widetilde{g}_i(-z^{-1})g_i(z) &= 0
\end{aligned} \tag{3.17}$$

The low pass and high pass filters are also inter-related. The relations for both trees

are described in equation 3.18

$$\begin{aligned} \widetilde{g}_r(z^{-1}) &= z^{-1}h_r(-z) & \widetilde{g}_i(z^{-1}) &= z^{-1}h_i(-z) \\ g_r(z) &= z\widetilde{h}_r(z^{-1}) & g_i(z) &= z\widetilde{h}_i(z^{-1}) \end{aligned} \tag{3.18}$$

Finally, as the Q-shift filter design method uses orthonormal filters, the analysing and synthesis filter taps are reversals of one another as described in equation 3.19

$$h_r(z) = \widetilde{h}_r(z) \qquad h_i(z) = \widetilde{h}_i(z) \tag{3.19}$$

3.6 The Lifting Scheme

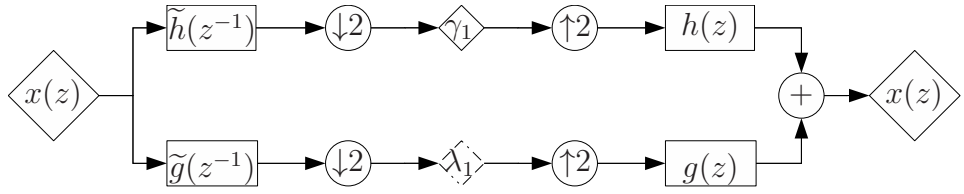
The lifting scheme introduced in [12] is a method for implementing the DWT that has benefits over the recursive finite impulse response filter bank method discussed in section 3.4.

Lifting provides a reduction in computational expense by a factor approaching 2 for long filter lengths [4]. In addition, it is possible to adopt a lifting scheme that maps integer input signals to integer wavelets coefficients even for non-integer filter coefficients, by applying a rounding function, whilst preserving the perfect reconstruction property of the wavelet transform [1, 2, 13]. This makes the lifting scheme extremely attractive for hardware implementation.

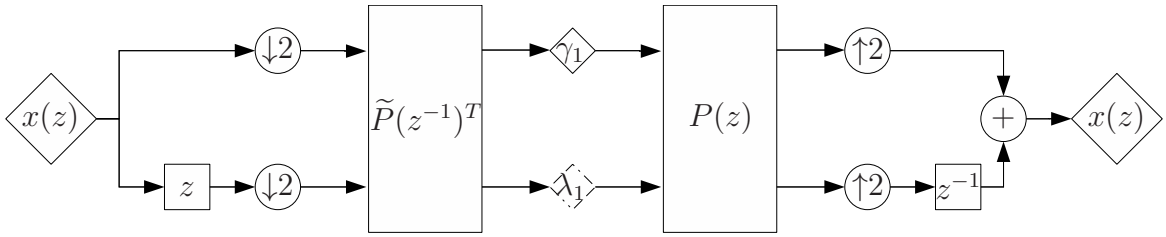
3.6.1 Polyphase Representation

In section 3.4, an implementation of the DWT using a recursive FIR filter bank was detailed. Figure 3.4(a) shows a representation of one stage of such an implementation.

This can be seen to be an inefficient implementation, as half the wavelet and scaling coefficients calculated are immediately discarding by decimation in the analysing stage, whilst the interpolating stage causes multiplications by zeros to be performed.



(a) FIR filterbank implementing a DWT



(b) Polyphase representation of the DWT

Figure 3.4: FIR filterbank and polyphase representations of the DWT

The lifting scheme instead uses as its base the polyphase representation of the filters, which can be arrived at by applying the Noble identities [14]. The Noble identities for the general case are shown in figure 3.5.

(a) Decimation

(b) Interpolation

Figure 3.5: The Noble Identities

The Noble identities allow the recursive FIR structure for the DWT shown in figure 3.4(a) to be restructured into the polyphase represented in figure 3.4(b). The delay operation to the lower path, followed by decimation separates the input signal $x(z)$ into odd and even samples, x_e and x_o. Using matrix notation, the operation shown in figure 3.4(b) is described mathematically by equations 3.20 and 3.21:

$$\begin{bmatrix} \lambda(z) \\ \gamma(z) \end{bmatrix} = \widetilde{P}(z^{-1})^T \begin{bmatrix} x_e \\ zx_o \end{bmatrix} \tag{3.20}$$

$$\begin{bmatrix} x_e \\ z^{-1}x_o \end{bmatrix} = P(z) \begin{bmatrix} \lambda(z) \\ \gamma(z) \end{bmatrix} \tag{3.21}$$

Where $\widetilde{P}(z^{-1})^T$ and $P(z)$ are the polyphase analysis and synthesis matrices, respectively. The polyphase matrices are constructed from the polyphase representations of $\widetilde{h}(z)$, $h(z)$, $\widetilde{g}(z)$ and $g(z)$. These are given in equation 3.22

$$\begin{aligned} \widetilde{h}(z^{-1}) &= \widetilde{h}_e(z^{-2}) + z\widetilde{h}_o(z^{-2}) & \qquad h(z) &= h_e(z^2) + z^{-1}h_o(z^2) \\ \widetilde{g}(z^{-1}) &= \widetilde{g}_e(z^{-2}) + z\widetilde{g}_o(z^{-2}) & \qquad g(z) &= g_e(z^2) + z^{-1}g_o(z^2) \end{aligned} \tag{3.22}$$

Where $\widetilde{h}_e$, $\widetilde{h}_o$, $\widetilde{g}_e$, $\widetilde{g}_o$, h_e, h_o, g_e and g_o are the odd and even coefficients of the analysis and synthesis filter. They are given in equation 3.23:

$$\begin{aligned} \widetilde{h}_e(z^{-1}) &= \sum_k z^{-k}\widetilde{h}_{(2k)}(z^{-1}) & \qquad \widetilde{h}_o(z^{-1}) &= \sum_k z^{-k}\widetilde{h}_{(2k+1)}(z^{-1}) \\ h_e(z) &= \sum_k z^{-k}h_{(2k)}(z) & \qquad h_o(z) &= \sum_k z^{-k}h_{(2k+1)}(z) \\ \widetilde{g}_e(z^{-1}) &= \sum_k z^{-k}\widetilde{g}_{(2k)}(z^{-1}) & \qquad \widetilde{g}_o(z^{-1}) &= \sum_k z^{-k}\widetilde{g}_{(2k+1)}(z^{-1}) \\ g_e(z) &= \sum_k z^{-k}g_{(2k)}(z) & \qquad g_o(z) &= \sum_k z^{-k}g_{(2k+1)}(z) \end{aligned} \tag{3.23}$$

The polyphase matrices can now be defined as equation 3.24:

$$\widetilde{P}(z^{-1})^T = \begin{bmatrix} \widetilde{h}_e(z^{-1}) & \widetilde{h}_o(z^{-1}) \\ \widetilde{g}_e(z^{-1}) & \widetilde{g}_o(z^{-1}) \end{bmatrix} \qquad P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \tag{3.24}$$

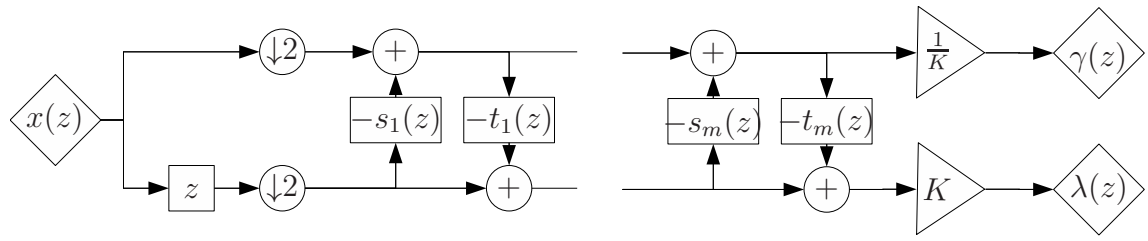
Equations 3.20 and 3.21 imply that $\widetilde{P}(z^{-1})^T = P(z)^{-1}$, and that the perfect reconstruction property can now be expressed as equation 3.25. This can be verified by substituting the equalities from equations 3.18 and 3.19 into the polyphase matrices in equation 3.24

$$P(z)\widetilde{P}(z^{-1})^T = \mathbf{I} \tag{3.25}$$

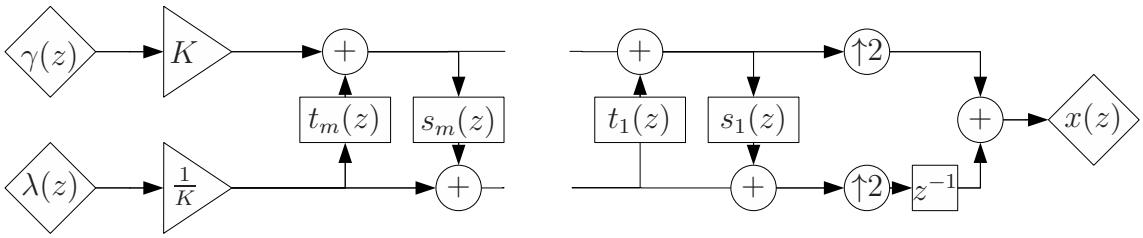
It has been shown [4] that this is only possible if $\det\{P(z)\}$ is a monomial in z, i.e. $\det\{P(z)\} = Cz^p$. It can be assumed that $\det\{P(z)\} = 1$ by dividing $g_e(z)$ and $g_o(z)$ by $\det\{P(z)\}$.

3.6.2 Lifting Transform

The lifting transform comprises alternating primal lifting and dual lifting steps, followed by a scaling factor. Primal lifting is the lifting of low pass coefficients by a function $s(z)$ of the high pass coefficients, whilst dual lifting is the lifting of the high pass coefficients by a function $t(z)$ of the low pass coefficients. This is shown in figure 3.6



(a) Analysis



(b) Synthesis

Figure 3.6: The Lifting Scheme. The synthesis transform is calculated as the reverse of the analysis lifting transform

Examining figure 3.6 with regard to figure 3.4(b), it is apparent that the polyphase matrices $P(z)$ and $\widetilde{P}(z^-1)^T$ have been decomposed into a series of lifting steps, which are described by equations 3.26, 3.27 and 3.28.

$$P(z) = \prod_{i=1}^{m} \left\{ \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \right\} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \tag{3.26}$$

$$\widetilde{P}(z) = \prod_{i=1}^{m} \left\{ \begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix} \right\} \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix} \tag{3.27}$$

$$\widetilde{P}(z^{-1})^T = \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix} \prod_{i=m}^{1} \left\{ \begin{bmatrix} 1 & 0 \\ -t_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -s_i(z) \\ 0 & 1 \end{bmatrix} \right\} \tag{3.28}$$

3.6.3 Factoring FIR into Lifting Steps

In general, providing that $\det\{P\} = 1$, any DWT described by a FIR filterbank can also be described as a series of lifting steps [4]. Starting with $P(z)$, the lifting steps can be factored using an iterative approach. As equations 3.26 and 3.28 show, once a solution has been found for $P(Z)$, $\widetilde{P}(z^{-1})$ can be found by executing the lifting steps in reverse order. If the filters are orthogonal, as in the case of the DTCWT in [7], it can be seen from equations 3.26 and 3.27 that the factorisation is not unique as $P(z) = \widetilde{P}(z)$.

Start by extracting a primal-lifting step, equation 3.29:

$$P(z) = \begin{bmatrix} {}^0h_e(z) & {}^0g_e(z) \\ {}^0h_o(z) & {}^0g_o(z) \end{bmatrix} = \begin{bmatrix} 1 & s_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1h_e(z) & {}^1g_e(z) \\ {}^0h_o(z) & {}^0g_o(z) \end{bmatrix} \tag{3.29}$$

where the factoring iteration is denoted by the leading superscript.

If filter $h(z)$ is defined as a Laurent series $h(z) = \sum_{k=p}^{q} h_k z^{-k}$, the degree of a Laurent series is defined as $|h| = q - p$ and $|0| = -\infty$.

Using the Euclidean algorithm[16], it is possible to perform long division on Laurent polynomials. In general, for $\frac{a}{b}$ where $|a| \geq |b|$, the division will not be exact, and so division with a remainder is achieved, $\frac{a}{b} = c + r$. The degree of the factor c is equal to

the difference in degree of a and b, $|c| = |a| - |b|$.

Equation 3.18 shows that the FIR filters are of the form of Laurent polynomials, and so from equation 3.29:

$$\begin{aligned} {}^0h_e &= {}^1h_e +^0 h_o s_1(z) \\ {}^0g_e &= {}^1g_e +^0 g_o s_1(z) \end{aligned} \tag{3.30}$$

Rearranging for ${}^0g_e$ in equation 3.30, s_1 and ${}^1g_e$ are calculated as the factor of Laurent polynomial division and the remainder respectively:

$$\frac{{}^0g_e}{{}^0g_o} = s_1(z) +^1 g_e \tag{3.31}$$

By substituting $s_1(z)$ from equation 3.31 into equation 3.30, ${}^1h_e$ can be calculated using

$${}^1h_e =^0 h_e -^0 h_e s_1(z) \tag{3.32}$$

Once the prime lifting step has been extracted, the dual-lifting step can be extracted

$$\begin{aligned} P(z) = \begin{bmatrix} {}^0h_e(z) & {}^0g_e(z) \\ {}^0h_o(z) & {}^0g_o(z) \end{bmatrix} &= \begin{bmatrix} 1 & s_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1h_e(z) & {}^1g_e(z) \\ {}^0h_o(z) & {}^0g_o(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & s_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_1(z) & 1 \end{bmatrix} \begin{bmatrix} {}^1h_e(z) & {}^1g_e(z) \\ {}^1h_o(z) & {}^1g_o(z) \end{bmatrix} \end{aligned} \tag{3.33}$$

Similarly, this gives:

$$\begin{aligned} {}^0h_o &= {}^1h_o +^1 h_e t_1(z) \\ {}^0g_o &= {}^1g_o +^1 g_e t_1(z) \end{aligned} \tag{3.34}$$

which, with the use of the Euclidean algorithm, allows the calculation of ${}^1h_o$,${}^1g_o$ and $t_1(z)$.

In general, for the extraction of each block i comprising a primal and dual lifting step, the filters ${}^ih_e$, ${}^ih_o$, ${}^ig_e$, ${}^ig_o$ and the lifting factors can be calculated using equations 3.35.

$$\begin{aligned} {}^{(i-1)}h_e &= {}^ih_e +^{(i-1)} h_o t_i(z) \\ {}^{(i-1)}g_e &= {}^ig_e +^{(i-1)} g_o t_i(z) \\ {}^{(i-1)}h_o &= {}^ih_o +^{(i-1)} h_e t_i(z) \\ {}^{(i-1)}g_o &= {}^ig_o +^{(i-1)} g_e t_i(z) \end{aligned} \tag{3.35}$$

Lifting steps may now be extracted iteratively as equation 3.26 until the result leaves ${}^{(m+1)}h_e = K$, ${}^{(m+1)}g_o = \frac{1}{K}$, and ${}^{(m+1)}h_o =^{(m+1)} g_e = 0$.

3.7 Software Implementation

3.7.1 Generating Filter Coefficients

FIR Filter Coefficients

The filters $h(z)$, $g(z)$, $\widetilde{h}(z^{-1})$, and $\widetilde{g}(z^{-1})$ are calculated using frequency domain energy minimisation as in [7]. MATLAB code for the generation of even length q-shift filters is provided by Kingsbury. An 80-tap filter generated using this algorithm is used throughout this thesis. For reference, the coefficients are those generated using the published code with the MATLAB syntax:

$$[\widetilde{h}(z^{-1}), h(z), \widetilde{g}(z^{-1}), g(z)] = \text{qshiftgen}([\ 80\ \tfrac{1}{3}\ 1\ 1\ 1\ ])$$

The filters used for the first stage of the DTCWT are given in appendix A.1. The remaining stages use the filters as described above.

Converting From FIR Filter Coefficients To Polyphase Representation

To calculate the polyphase matrix $P(z)$ shown in equation 3.24, $h(z)$ and $g(z)$ are split into their odd and even components using equation 3.23. MATLAB code to generate $h_e(z)$, $h_o(z)$, $g_e(z)$ and $g_o(z)$ given $h(z)$ and $g(z)$ is included in appendix A.2.

When using $h(z)$ and $g(z)$ using coefficients generated by the method in [7], $\det\{P(z)\} = z\sum_k h(k)^2$ for a filter of length k. A unit determinant for $P(z)$ is required for the perfect reconstruction condition in equation 3.25. This can be achieved by dividing $h(z)$ by $\sqrt{\sum_k h(z)^2}$ and $g(z)$ by $z\sqrt{\sum_k h^2(z)}$ prior to performing the odd and even component separation.

Factorising The Polyphase Filter Into Lifting Stages

To factorise $P(z)$ into lifting stages, the methodology based on the Euclidean algorithm for Laurent polynomial long division described in section 3.6.3 is followed. The MATLAB implementation for the algorithm, provided in appendix A.3, deviates from the given method by terminating the Euclidean Laurent polynomial division step when a quotient of degree 1 is reached. This has the beneficial effect of enabling the lifting transform for a given sample to be calculated from adjacent samples [4].

The MATLAB code in appendix A.3 matches the largest and smallest powers of z in the non-unique factorisation process. The algorithm in appendix A.3 also differs from the section 3.6.3 by factorising the scaling factors k and $\frac{1}{k}$ into a series of four additional lifting stages [4], allowing the lifting transform to be calculated using only lifting stages. This has the significant benefit of allowing development of fully parallel implementations of the lifting transform.

3.7.2 DWT - Filterbank implementation

The software used to perform the DWT and DTCWT via a FIR filterbank is based on the WT and IWT MATLAB routines of the Uvi_Wave package released under the

GNU general public licence. These routines were subject to a scaling modification to ensure that transforms are energy invariant.

3.7.3 DWT - Lifting implementation

Once the polyphase matrix $P(z)$ has been factored into a series of primal and dual lifting stages, one level of the wavelet transform can be implemented in an almost trivial fashion as operations using adjacent samples for both the primal and lifting stages $s(z)$ and $t(z)$. This is illustrated in figure 3.7

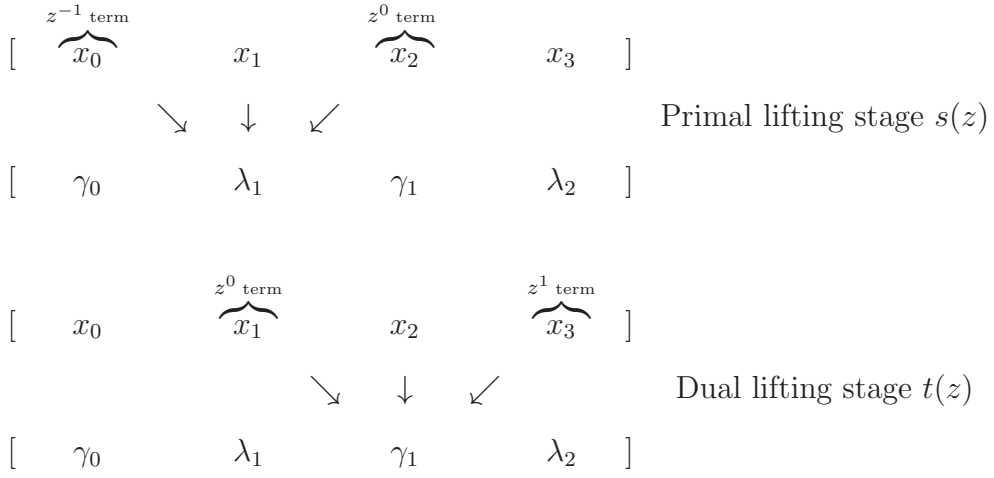


Figure 3.7: Calculating dual and primal lifting stages using adjacent samples

For signals with a finite length, the implementation requires attention to the boundaries, i.e. the first and last samples require non-existent data points to calculate wavelet coefficients. For long filter lengths, more samples from beyond the data boundaries are required. In general, for a lifting filter with M lifting stages, $M-1$ points either side of the data set will be required. This is illustrated in figure 3.8

Several possible solutions to this problem are presented in [5]. The simplest solutions involve padding the signal, either with zeros, or with repeated or mirrored sets of the recorded signal. Whilst simple, these solutions can lead to discontinuities in the wavelet transform. However, for the application in ISRIE, where the signals have lengths that are significantly greater than the boundary padding, these discontinuities can be ignored.

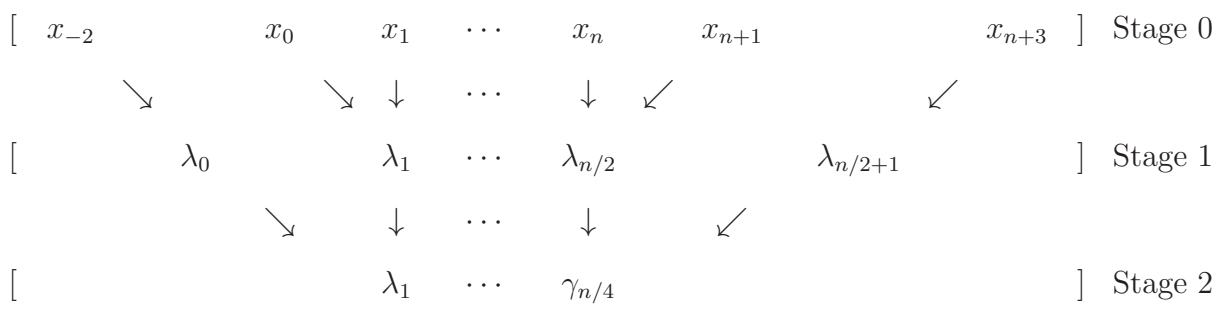


Figure 3.8: Edge padding requirements

The MATLAB code in appendix A.4 contains an implementation using zero padding to perform a lifting transform on a series of data, given the lifting coefficients Pa. The implementation for the inverse transform is given in A.5

3.8 Conclusion

This chapter has provided an introduction to two time-frequency transformations, the STFT and the DTCWT, which are used extensively in the separation methodologies in the succeeding chapters. Of the myriad of wavelet transform implementations available, the DTCWT has been chosen because it can directly replace the STFT, which is used throughout the signal separation literature (refer to chapter 2).

This chapter has highlighted the periodic nature of the Fourier transform, which makes it an ideal transform for application to constant tone sources. For varying-frequency sources, the STFT was introduced using a Hamming window. This improved time resolution at the expense of frequency resolution. Finding the optimum window size for a particular source to provide a good time-frequency representation has been noted as requiring some *a-priori* knowledge of the source signal's properties.

It has been noted that the dyadic sampling property of the wavelet transform removes the need for a windowing function and hence the requirement for *a-priori* knowledge of the signal properties. This benefits the aim of developing a separation algorithm

that can separate arbitrary source types present in the soundscape.

The wavelet basis for the transform can, however, be chosen from an infinite range and the time-frequency performance will vary depending upon this selection. The wavelet basis chosen for use in this thesis is that of the original work, and its performance has not been verified in this chapter. Further work is described in subsequent chapters, comparing the performance of the DTCWT with that of the STFT. Optimising wavelet filters is considered beyond the scope of this thesis.

3.9 Chapter Bibliography

[1] M. Adams and F. Kossentini. Performance evaluation of reversible integer-to-integer wavelettransforms for image compression. In *Proc. of Data Compression Conference*, 1999.

[2] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis*, 5 (3):332–369, July 1998.

[3] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992. ISBN 978-0898712742.

[4] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998.

[5] A. Jenson and A. la Cour-Harbo. *Ripples in Mathematics*. Springer Verlag, 2001.

[6] N. Kingsbury. A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. In *Proc. IEEE Conf. on Image Processing*, number 1429, Vancouver, September 2000. URL `http://www-sigproc.eng.cam.ac.uk/~ngk/publications/ngk00b.zip`.

[7] N. Kingsbury. Design of q-shift complex wavelets for image processing using frequency domain energy minimisation. In *Proc. IEEE Conf. on Image Processing*, number 1199, Barcelona, September 2003. URL `http://www-sigproc.eng.cam.ac.uk/~ngk/publications/ngk_icip03a.pdf`.

[8] N. G. Kingsbury. The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement. In *Proc. European Signal Processing Conference*, pages pp 319–322, Rhodes, September 1998.

[9] N. G. Kingsbury. Shift invariant properties of the dual-tree complex wavelet transform. In *Proc. IEEE Conf. on Acoustics, Speech and Signal Processing*, page paper SPTM 3.6, Phoenix, AZ, March 1999.

[10] I. Selesnick, R. Baraniuk, and N. Kingsbury. The dual-tree complex wavelet transform. *Signal Processing Magazine, IEEE*, 22(6):123–151, November 2005.

[11] Y. Sheng. *Wavelet transform*, pages 747–827. The Electrical Engineering Handbook Series. CRC Press, Fl (USA), 1996.

[12] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, March 1998. doi: 10.1137/S0036141095289051.

[13] G. Uytterhoeven, G. Uytterhoeven, F. V. Wulpen, F. V. Wulpen, M. Jansen, M. Jansen, M. Jansen, D. Roose, D. Roose, D. Roose, A. Bultheel, A. Bultheel, and A. Bultheel. Waili: Wavelets with integer lifting. Technical report, Department of Computer Science, Katholieke Universiteit Leuven, 1997.

[14] P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[15] C. Valens. A really friendly guide to wavelets. `http://pagesperso-orange.fr/polyvalens/clemens/download/arfgtw_26022004.pdf`, 1999. URL `http://pagesperso-orange.fr/polyvalens/clemens/download/arfgtw_26022004.pdf`.

[16] C. Valens. The fast lifting wavelet transform. `http://polyvalens.pagesperso-orange.fr/clemens/download/tflwt_26022004.pdf`, 1999. URL `http://pagesperso-orange.fr/polyvalens/clemens/download/tflwt_26022004.pdf`.

Chapter 4

Validation of assumptions

Contents

4.1 Chapter overview

Literature concerning the application of sparse separation techniques is generally focused on the separation of speech and harmonic music. The purpose of this chapter is to validate the assumptions underlying time-frequency sparse source separation for

applications other than speech, which are representative of separation tasks ISRIE may perform in rural and urban soundscapes.

A series of data sets are formed representing typical separation tasks, and separation performance is analysed. A speech bench mark using this method is also given for comparison, both to the separation results achieved in this chapter, and also to published speech separation results.

4.2 Methodology

4.2.1 Test Cases

Recordings of sources typical of both urban and rural soundscapes are used to form test cases representative of separation tasks required by ISRIE. Three classes of sound are used; transport; light plant; and birdsong, chosen to be representative of ecology applications, and also because of their prevalence within urban environments.

Bird song separated from bird song

This test case is designed to simulate a typical bio-diversity study in a rural setting. The prominent noise sources are all bird song, and are all potentially of interest. By separating out the individual noise sources, automatic classification of the birdsong into species allows the number and type of birds within the soundscape to be calculated.

Bird song separated from transport and plant noise backgrounds

These environments are designed to extend the test case of separating bird song in a rural setting to two typical urban environments.

Transport separated from transport

Designed to simulate soundscape recordings by a road side. This setting is typical of planning assessments, and also may be of interest for applications classifying vehicular type.

Transport separated from bird song and plant noise backgrounds

The above test case is extended to consider other urban and rural recording environments.

Plant noise separated from plant noise

This test case is the separation from plant sources from plant sources. The sources chosen to simulate this test case are air conditioning plant, pumps and fan units, typical of urban industrial estates and data centres. The test case is designed to test the ability to identify the noise levels associated with a single source. This is perhaps the most challenging of the test cases, as typically the emissions from such sources are constant broadband noise with little time-variant harmonic content, which is unlikely to satisfy the ω-disjoint orthogonality condition.

Plant noise separated from bird song and transport backgrounds

These test cases are designed to test the applicability of this sparse source separation in typical scenarios using British Standard 4142, which is discussed in chapter 1.

4.2.2 Sparse separation

For each test case, the sparsity and separation obtainable in each mixture is to be calculated in three domains; time, STFT, and wavelet using the DTCWT.

The time domain sparsity measure is used to provide a baseline for the sparsity between the sources in each mixture, as many of the sources used in the test set, particularly the birdsong, are discontinuous in the time domain. Exploiting time domain sparseness can be considered analogous to the methodology used in the calculation of current sound metrics (see chapter 1).

Comparison of results in the STFT domain can be made with the results of speech mixtures in [3]. The methodology used in [3] does not document the window size used. The window size is a significant factor in the performance of the separation in the STFT. Based on heuristic results, the optimum window size found for mixtures of two speech signals is 1024 samples, with the audio mixtures recorded at 44.1 kS/s with 16-bit precision. Accordingly, the STFT window used in these tests is 1024 samples.

This window size is unlikely to be optimised for all sources in each of the test cases. However, as an optimum window size across all soundscapes of interest is unlikely to be found, comparison to the speech results in [3] is made under the assumption that these published results for speech mixtures were derived using an optimum window size.

The DTCWT domain's dyadic sampling properties remove the problem of calculating a window size for the transformation, in contrast to the STFT.

4.2.3 Measuring sparseness

Central to sparse separation is the concept of ω-disjoint orthogonality. For source s_i and interference s_j, a measure of the ω-disjoint orthogonality has been proposed [3], based on the proportion of signal energy dominance for each sample in the time-frequency domain (ω, τ).

Firstly, a logical bit mask Φ is defined, equation 4.1:

$$\Phi_a(\tau,\omega) = \begin{cases} 1 \quad | \quad 20log_{10}\left(\dfrac{|S_i(\omega,\tau)|}{\sum_{j=1,j\neq i}^{N}|S_j(\omega,\tau)|}\right) > a \\ 0 \quad | \quad otherwise \end{cases} \tag{4.1}$$

where a is a threshold in dB. This provides a mask for all instances where the power of source $S_i(\omega,\tau)$ is a dB greater than the summation of other sources $S_{(j\neq i)}(\omega,\tau)$.

This mask $\Phi_a(\omega,\tau)$ is then used to define a function $r(a)$ that describes the proportion of energy for source s_i that dominates the summation of other source contributions by a dB.

$$r_i(a) = \frac{\|\Phi_a(\omega,\tau)S_i(\omega,\tau)\|^2}{\|S_i(\omega,\tau)\|^2} \tag{4.2}$$

where $\|\cdot\|$ denotes the L^2 norm. It can be seen that if $r = 1$ for $a = \infty$, ω-disjoint orthogonality is perfectly satisfied, as in equation 2.14.

Equation 4.2 provides a tool to measure the ω-disjoint orthogonality between a source and all other interference sources.

These equations can be applied to signals in all the transform domains under consideration, including time.

4.2.4 Mixing and demixing model

For this validation exercise, separation is based on an ideal binary mask for a threshold a. For each test, a mixture x is calculated as the sum of the source of interest with $M-1$ interfering sources, equation 4.3:

$$x = s_i + \sum_{j=1,j\neq i}^{M} s_j \tag{4.3}$$

The binary mask Φ_a is then calculated for a threshold a. An estimate of the source $\widehat{s_i}$, equation 4.4, is calculated using the binary mask:

$$\widehat{s_i} = \Phi x \tag{4.4}$$

This is performed for $a = 0 \rightarrow 30$ dB at 1 dB intervals for every permutation of original recordings for each test.

4.2.5 Performance Metrics

The performance of each separation is measured using the SIR metric as suggested in [1]. In this case, SIR can be calculated using the ideal binary mask Φ_a applied to the original source and all interference sources, for all values of a.

$$\mathrm{SIR}_a = \frac{\left\|\Phi_a s_i\right\|^2}{\left\|\Phi_a \sum_{j=1, j\neq i}^{M} s_j\right\|^2} \tag{4.5}$$

4.3 Data sets

4.3.1 Original recordings

Bird song

The birdsong recordings are a set of 20 samples of Japanese bird song [4], each 20 seconds long. The recordings used are displayed in figure B.1, in appendix B. Figure B.4(a) shows a typical spectrogram.

The original recordings were 48 kS/s at 16-bit resolution, and have been down sampled to 44.1 kS/s to match the other data sets used here.

Plant recordings

These recordings were made by colleagues at ISVR at typical installations around Southampton. The recordings were made at 44.1 kS/s and 16 bit resolution. The data set consists of 10 second samples, taken for 6 different sources, figure B.2. Figure B.4(b) shows a typical spectrogram.

Plant included in the data set are air-conditioning units, an industrial heater, and typical plant room installations containing pumps and fans.

Transport recordings

These recordings were taken 1 metre from the road side using the omnidirectional (W) component of a soundfield ST350 microphone. The sampling rate is 44.1 kS/s, at 16-bit resolution.

The recording location is a B-road just outside of Stamford Bridge, North Yorkshire (grid ref: 725576). The location was chosen for its quiet aspect, and low incidence of cars, to record single vehicle data. The data set consists of 10 vehicles, from which samples approximately 12 seconds long were taken. Figure B.3. Figure B.4(c) shows a typical spectrogram.

4.3.2 Generating test case mixtures

For each pair of data sets, the recordings were truncated to the same number of samples. Each recording was then normalised to unit energy (note, figures B.1, B.2 and B.3 are normalised to unit magnitude for display clarity). This normalisation step was performed to improve the repeatability of the results by removing recording variations in the original sources. This has the consequence that sound sources that are concentrated into short temporal periods will be perceived as louder, whilst sources that are continuous will be perceived to be quieter. In the interests of repeatability this is unavoidable, and can be considered analogous to recording these sources from a greater

of lesser distance.

Mixtures were then created for each possible combination of sources within the two sets. For mixtures of a single type of noise source, a further mixture containing two interfering sources was also generated. This allows the performance of separation of soundscapes with more simultaneously active sources to be gauged.

No Sources in mixture	Bird song		Plant		Transport	
	2	3	2	3	2	3
Bird song	380	6840	120	-	240	-
Plant	120	-	30	120	72	-
Transport	240	-	72	-	132	1320

Table 4.1: Number of mixtures created for each test case

No Sources in mixture	Bird song		Plant		Transport	
	2	3	2	3	2	3
Bird song	2.1	38	0.3	-	0.8	-
Plant	0.3	-	0.08	0.3	0.2	-
Transport	0.8	-	0.2	-	0.44	4.4

Table 4.2: Combined length of audio for each test case (hours)

The metric r_a used to determine the sparsity of each source within the mixture is calculated, along with the SIR. As both sources have unity energy, this measure is also the SIR gain that the separation provides.

The mean SIR for each dataset was calculated. Table 4.1 shows the number of mixtures created for each test case, with the equivalent audio time for each test case displayed in table 4.2

4.4 Results

The results for the separation of the mixtures as described by the test cases are given in appendix C.

The results single type mixtures of 2 and 3 sources of bird song, plant, and transportation recordings are shown in figures B.5(a) to B.7(b).

Results for 2 source mixtures of differing source type are given in figures B.8(a) to B.13(b). Note that the axis are not equally scaled, and care should be taken when comparing graphs.

In all except a single case, the results show a marked improvement in sparsity performance achieved in the STFT domain over the time domain. Of particular note is the case of the mixture of 3 sources of bird song, figures B.5(a) and B.5(b). For a threshold of $a = 30$ dB, an improvement of over 3 dB source power remaining following the application of the binary bit mask, whilst providing a SIR gain of over 40 dB.

The DTCWT results can also be seen to perform better than the time domain, typically within 1 dB of the results obtained in the STFT domain.

The exception to this increase in performance over the time domain sparsity levels is seen in the 3 source mixture of plant sources, figures B.6(a) and B.6(b). The same set of figures also show little improvement in the STFT over the time domain for the case of the 2 source mixture. This poor performance in the separability of sources in the STFT domain shows that the STFT transform used is very poorly matched to exploit any sparseness in the frequency domain between mixtures of plant sources, which in this set of test data contains wideband noise sources.

(a) Average R_a (%) from literature

N	Threshold a dB		
	5	10	15
2	92	87	80
3	86	78	66

(b) Average SIR (dB) from literature

N	Threshold a dB		
	5	10	15
2	18.10	21.76	25.53
3	15.50	19.27	23.19

(c) Average R_a (%) achieved

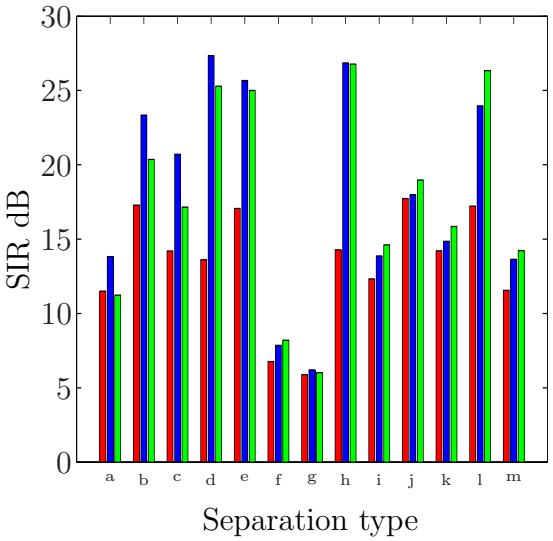
N	Threshold a dB		
	5	10	15
2	91	83	77

(d) Average SIR (dB) achieved

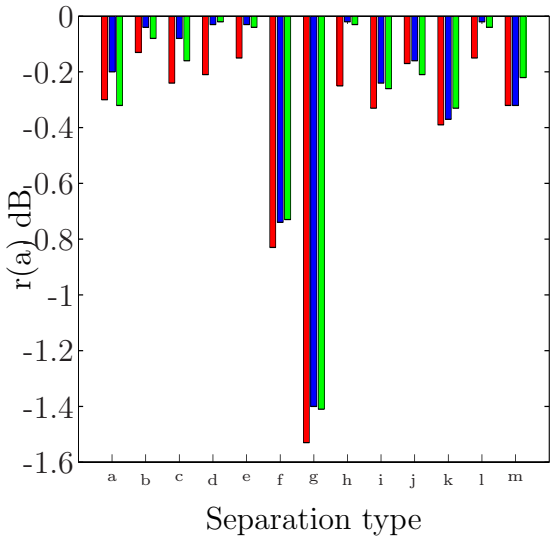
N	Threshold a dB		
	5	10	15
2	17.42	22.51	26.28

Table 4.3: Results published in [2] for the separation of mixtures of two speakers in the STFT domain (figures a and b), compared to speech mixture benchmark achieved using this method (figures c and d)

As an aid for comparison of performance between different domains across the test cases, results for a subset of mask thresholds for both r_a results and SIR results are



(a) SIR achieved using binary masking with threshold $a = 0$ dB



(b) Comparison of ω-disjoint orthogonality. Plot of mean for all sources mixtures using threshold $a = 0$ dB

Figure 4.1: Comparison of results between the mixture types using threshold $a = 0$ dB as the basis for binary masking. The time domain samples are shown in red, the STFT results (with window size 1024 samples) are shown in blue, and the DTCWT results are shown in green.
Key: a = Speech from speech. b = bird from bird. c = bird from 2 birds. d = bird from plant. e = bird from transport. f = plant from plant. g = plant from 2 plant. h = plant from bird. i = plant from transport. j = transport from transport. k = transport from 2 transport. l = transport from bird. m = transport from plant

given in tables B.2 and B.1 respectively. A subset of the results for $r(0)$ is shown in figure 4.1. It can be seen clearly in this figure that the SIR results achieved (figure 4.1(a)) is strongly coupled to the degree of ω-disjoint orthogonality exibited in the orginal sources (figure 4.1(b))

For comparison, results for speech mixtures of 2 and 3 sources published in [3] are given in tables 4.3(a) and 4.3(b) for r_a and SIR performance respectively. It can be seen from tables 4.3(c) and 4.3(d) that processing speech mixtures using the method discussed in this chapter leads to results very similar to those published. For the case of $N = 2$, results are within 1dB for the SIR metric.

The performance of the separation for mixtures of bird song with either bird song, transport or plant sources far exceeds the performance for the speech mixtures in [3]. An improvement approaching an order of magnitude is achieved in the SIR metric. This is attributable to the high value of r_a achieved in both transform domains for mixtures containing birdsong. Performing the separation in the transform domain for

these mixtures can be seen to improve SIR by over 15 dB in some cases, proving the suitability of the transform for mixtures of this type.

Separation of transport noise from transport noise is comparable to the results in table 4.2(a) and 4.2(b). An important observation here is that the improvement in both transform domains over the time domain is only of the order of 1-2 dB, which suggests that the separation performance gains little from the time-frequency transforms. Instead the result is dependent on the inherent time domain sparsity present in the source mixtures, and where overlaps in the time domain occur w-disjoint orthogonality is low, as the sources contain similar frequency components. However, transport sources are periodic by nature, and so exploiting the sparseness in this way is valid approach, even if the performance gains from the added complexity of a time-frequency transform is only of the order of a few dB.

The poorest performance of all the test cases is separation of plant sources from plant sources, which are approximately an order of magnitude worse than the published speech results. The results for r_a confirm that the sparseness of this mixture is also the worst of all the test cases. This poor performance is attributed to the continuous time domain presence of both noise sources, combined with the wide-band nature of the plant noise spectrum. Performance is improved using the STFT and DTCWT, both providing between 1-2 dB performance improvement over the time domain.

4.5 Chapter synopsis

This chapter has examined the applicability of sparse source separation for typical soundscape test cases containing non-speech signals. In all cases, separation performance is better than the original mixture with good performance achieved for blindly separating mixtures for several example test cases of target applications. This has demonstrated that non-speech noise sources may be separated by any algorithm or method that employs binary masking in the time-frequency domain.

The existence of sparse representations using both the STFT and DTCWT was demon-

strated by the metric r_a. Good separation is dependant on the assumption of ω-disjoint orthogonality, so this measure is an excellent indicator of the likely separation performance for any mixture of signals. This metric is also used in work on speech separation [3], allowing a direct comparison of results to be made with existing literature.

Separation of the sources from the mixture was achieved using the ideal binary mask for a given threshold a. The metrics achieved for the SIR show for most of the test cases at least as good a performance as is obtained for the separation of speech mixtures.

The best performing test cases are those where dissimilar source types make up the soundscape. This is due to the tendancy for the frequency components of dissimilar sources not to coincide, a characteristic which improves separation using a time-frequency transform.

Applications discussed in chapter 1 where separation performance is maximised include ecologoical sounds, which tend to be tonal with harmonics as well as sparse in the time domain.

This chapter has also shown that significant separation of ecological sources is possible from interference made up of mechanical sources. The converse has also been shown to be effective, i.e. the separation of mechanical sound sources in the presence of zoological noise. An example application is isolating interference of the dawn chorus in long term (24 hour) sound recordings, typical in PPG 24 and BS 4142 applications.

Separation performance for another typical example of a noise nuisance application has been demonstrated: separation of traffic noise from plant noise. In a typical suburban setting, the results here show that it is possible to successfully separate noise from passing traffic from a long term recording of plant noise, typically air conditioning units in BS 4142 scenarios.

4.6 Chapter Bibliography

[1] R. Gribonval, E. Vincent, C. Fevotte, L. Benaroya, et al. Proposals for performance measurement in source separation. In *4th International symposium on independant*

component analysis and blind signal separation (ICA2003), April 2003.

[2] S. Rickard and F. Dietrich. Doa estimation of many w-disjoint orthogonal sources from two mixtures using duet. In *Tenth IEEE Workshop on Statistical Signal and Array Processing*, pages 311–314, 2000. doi: 10.1109/SSAP.2000.870134.

[3] S. Rickard and Z. Yilmaz. On the approximate w-disjoint orthogonality of speech. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, volume 1, pages I–529–I–532, 2002. doi: 10.1109/ICASSP.2002.1005793.

[4] K. Tsuruhiko and M. Michio. *The songs and calls of 333 birds in Japan: Non-songbirds.*, volume 1. Shogakukan Inc. ,Tokyo, 1996. ISBN 4-09-480072-7.

Chapter 5

Source separation

Contents

5.1 Chapter overview

The previous chapter explored the applicability of exploiting ω-disjoint orthogonality to effectively separate typical sources found within soundscapes in a transform domain.

This chapter aims to extend the two sensor model used by separation algorithms such as DUET [4] to a model capable of localising the direction of arrival of sources within a 3D soundscape, whilst also providing a method for the separation of the sources using time-frequency masking, as demonstrated in the previous chapter. The underlying assumptions of this method are discussed, and a set of standard performance metrics for analysing the performance of this method are given.

A series of experiments showing the performance of the algorithm under ideal and real conditions are presented. The effect of violating the required *omega*-disjoint assumption on source separation performance is shown and discussed.

5.2 Methodology

5.2.1 Mixing model

The two sensor model has been shown to be effective for both the under-determined separation of N sources from two mixtures, and also for the estimation of the direction of arrival of sources within a 2D half plane. Various extensions to this model, particularly those associated with the DUET algorithm discussed in chapter 2, have provided means to extend this performance to 3D. The majority of these extensions rely on large spaced microphone arrays, and knowledge of the geometry of the array. This setup is impractical for several of the applications discussed in chapter 1, where a monitoring station may have to be left unattended for extended periods of time. However, some of the methods reviewed in chapter 2 [6, 11], provide a model based on using a coincident microphone array, providing a compact solution and consistent physical array form factor.

The coincident microphone array provides B-format audio containing 3D information on sound pressure levels. The microphone chosen for recordings used in this chapter is the Soundfield ST350.

The B-format microphone contains a very closely spaced tetrahedral array of directional

sensors in an approximately 5 cm diameter enclosure to form an effectively coincident array at the wavelengths of interest. The audio channels from the four sensors are then subject to post-processing to give B-format sound based on the theory of ambisonics [2, 3]

The microphones in the array are denoted as left back (LB), left front (LF), right back (RB) and right front (RF), at azimuth and elevation locations in degrees, in a Cartesian coordinate system:

$$\begin{aligned} LF &= (-45°, 45°) \\ RF &= (135°, 45°) \\ LB &= (-135°, -45°) \\ RB &= (45°, -45°) \end{aligned} \tag{5.1}$$

where $(0°, 0°)$ is denotes the X-axis.

The B-format audio is then found by the summation of the sensor output according to equations 5.2

$$\begin{aligned} x &= LF - RB + RF - LB \\ y &= LF - RB - RF + LB \\ z &= LF - RB - RF - LB \\ w &= LF + RB + RF + LB \end{aligned} \tag{5.2}$$

where x, y and z are figure of eight responses along the Cartesian axes, and w is an omnidirectional response.

The mixing model for B-format sound is expressed in equation 5.3 for sources 1 to N and their 3D locational coordinates defined in terms of azimuth and elevation.

$$\begin{bmatrix} w(t) \\ x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ \cos(\theta_1)\cos(\lambda_1) & \dots & \cos(\theta_N)\cos(\lambda_N) \\ \sin(\theta_1)\cos(\lambda_1) & \dots & \sin(\theta_N)\cos(\lambda_N) \\ \sin(\lambda_1) & \dots & \sin(\lambda_N) \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \tag{5.3}$$

where $x(t)$,$y(t)$,and $z(t)$ are the mixtures observed on the Cartesian axes, $w(t)$ is the

mixture observed by the omnidirectional sensor. (θ_i, λ_i) are the azimuth and elevation for the direction of arrival of source s_i

5.2.2 Direction of arrival estimation

Spatial impulse response rendering (SIRR) [5, 10], along with directional audio coding [7, 8, 9] are techniques for the reproduction of room acoustics using multichannel loudspeaker systems.

Recordings are made in the room to be replicated using a coincident microphone array, and the resulting B-format audio is processed to extract the room's impulse response. This is later used to faithfully reproduce the original sound through an arbitrary loudspeaker system in another listening environment.

The method used by these two techniques for the extraction of source localisation information in the STFT domain can equally be applied here to form a direction of arrival estimate $\mathbf{d}$, in the cartesian coordinate system [5]. This is shown in equations 5.4 and 5.5.

$$\mathbf{d}(\omega, \tau) = -\Re\left(W^*(\omega, \tau)\mathbf{v}(\omega, \tau)\right) \quad \forall\ (\omega, \tau) \tag{5.4}$$

$$\mathbf{v}(\omega, \tau) = X(\omega, \tau)\begin{bmatrix}1\\0\\0\end{bmatrix} + Y(\omega, \tau)\begin{bmatrix}0\\1\\0\end{bmatrix} + Z(\omega, \tau)\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{5.5}$$

This method for calculating a direction of arrival estimate for each time-frequency point in the STFT domain can also be used to calculate a direction of arrival estimate for signals in the dual-tree complex wavelet transform (DTCWT) domain, as unlike the discrete wavelet transform, the DTCWT provides readily accessible phase information. This allows the DTCWT domain to be used as a direct alternative to the STFT domain. See chapter 3.

Another method for the analysis of the direction of arrival using coincident microphone arrays using the discrete wavelet transform is presented in [1]. The direction of arrival is estimated in the published method by using the B-format audio signals to simulate a cardioid response, as in equation 5.6.

$$M_\theta = W + X\cos(\theta) + Y\sin(\theta) \tag{5.6}$$

The direction of arrival is then found by finding the value of θ that maximises M_θ by performing a sweep of θ throughout the range $\theta = -\pi \rightarrow \pi$.

Although the published method is only for 2D signals, the methodology can be extended to 3D by redefining M as a function of the azimuth θ and elevation λ. Equation 5.6 can then be extended as equation 5.7

$$M_{(\theta,\lambda)} = W + X\cos(\theta)\cos(\lambda) + Y\sin(\theta)\cos(\lambda) \tag{5.7}$$

This method for localising sources in either 2D or 3D is computationally expensive, particularly in 3D, where calculating the direction of arrival by sweeping θ, λ in 1 degree increments requires over 100000 iterations.

By extending the method based on equations 5.4 and 5.5, an analytic solution to the problem can be found requiring a significant reduction in the computational complexity, whilst also giving an exact result rather than a solution based on a numerical analysis, provided that the approximate ω-disjoint orthogonality condition is met, i.e, one source dominates the energy at each time-frequency point.

5.2.3 Separation basis

Separation of the 3D soundscape can now be achieved based on the method of filtering using a binary bit mask in a sparse transform domain. A transform, either STFT or DTCWT, is applied to the B-format signals w, x, y, z to give the transformed signals

W, X, Y, Z. For each point in the transformed domain, the direction vector $\mathbf{d}$ is then calculated.

The bit mask Φ_n associated with estimating source n is then constructed as in equation 5.8 using the direction of arrival estimate $\mathbf{d}$ in equation 5.4 as the basis of the filtering decision.

$$\Phi_n = \begin{cases} 1 & | \quad \arccos(\hat{\mathbf{e}}_n \cdot \hat{\mathbf{d}}) \leq \delta \\ 0 & | \quad otherwise \end{cases} \qquad \forall n \tag{5.8}$$

where $\hat{\mathbf{e}}_n$ is the direction of arrival for source n, either known *a-priori* or estimated (see chapter 6). δ allows an error margin in radians to be set from the source location $\hat{\mathbf{e}}_n$

The sources may then be filtered in the sparse domain by applying the binary bit mask Φ_n for each source to the omni-directional component of the B-format audio W in the sparse transform domain.

$$\acute{S}_n = \Phi_n W \tag{5.9}$$

The inverse transform can then be performed on the filtered result for estimated source $\acute{S}_n$ to give $\acute{s}_n$ if a time domain signal is required, or left in the sparse domain and subjected to further processing such as feature extraction.

5.2.4 Assumptions

In addition to the constraint of approximate ω-disjoint orthogonality (equation 2.14 discussed in chapters 2 and 4) the soundscape sources are also required to exhibit radial sparsity for successful separation to be achieved using the above method. This is shown in equation 5.10, and implies that the direction of arrival for each source is distinct.

$$\hat{\mathbf{d}}_i \cdot \hat{\mathbf{d}}_j \neq 0 \qquad \forall j \neq i \tag{5.10}$$

5.3 Performance measures

5.3.1 SIR improvement

In the preceding chapter, equation 4.5 was given to provide a metric for the SIR performance using an ideal binary bit mask for normalised sources. In the more general case of non-normalised sources, a new metric must be defined, for measuring the SIR improvement of the separation algorithm:

$$\begin{aligned} SIR(n)_{gain} &= \frac{SIR(n)}{SIR_{mixture}} \\ &= \frac{\frac{\|\Phi_n S_n\|^2}{\|\Phi_n S_j\|^2}}{\frac{\|S_n\|^2}{\|S_j\|^2}} \\ &= \frac{\|\Phi_n S_n\|^2 \|S_j\|^2}{\|\Phi_n S_j\|^2 \|S_n\|^2} \end{aligned} \tag{5.11}$$

As this measure is defined in terms of the transform domain bit mask, this measure does not take into account any noise that may be added through the inverse transform process.

5.3.2 PSR - Preserved signal ratio

The preserved signal ratio of each source (PSR_n) is the energy of the ratio of the filtered signal estimate for each source $\acute{s}_n$ to the original source s_n. If the ω-disjoint orthogonality is strictly met, then PSR_n is equivalent to r_n, defined in equation 4.2. However, if this condition is only approximately satisfied, then some of the energy

present in the source estimate $\acute{s}_n$ will be due to other sources, and the PSR is defined as equation 5.12

$$PSR_n = \frac{\|\Phi_n S_n\|^2}{\|S_n\|^2} \tag{5.12}$$

As with the previous metric, PSR_n relies on the bit masking in the transform domain. However, unlike the SIR measure, as both the DTCWT and STFT transforms are energy invariant, PSR should not alter as a result of the inverse transform.

5.4 Experiments

5.4.1 Characterising microphone directional performance

The proposed method for directional estimation, equation 5.10, relies on the Soundfield microphone's directional performance. The recording environment, such as echoic surfaces, may also effect the directional information recorded at the microphone to a lesser extent.

This experiment aims to characterise the directional performance of the microphone (Soundfield ST350) compared to the ideal mixing model given in equation 5.3, under anechoic conditions.

Methodology

Two male speakers were recorded independently reading extracts from a novel whilst stationary at locations around a microphone positioned in the centre of an anechoic chamber. The recordings were made with a sampling frequency of 44.1 kS/s with 16 bit precision.

An estimate of the precise location of the speakers relative to the microphones coordinates was made using the maximum peaks method described in chapter 7.

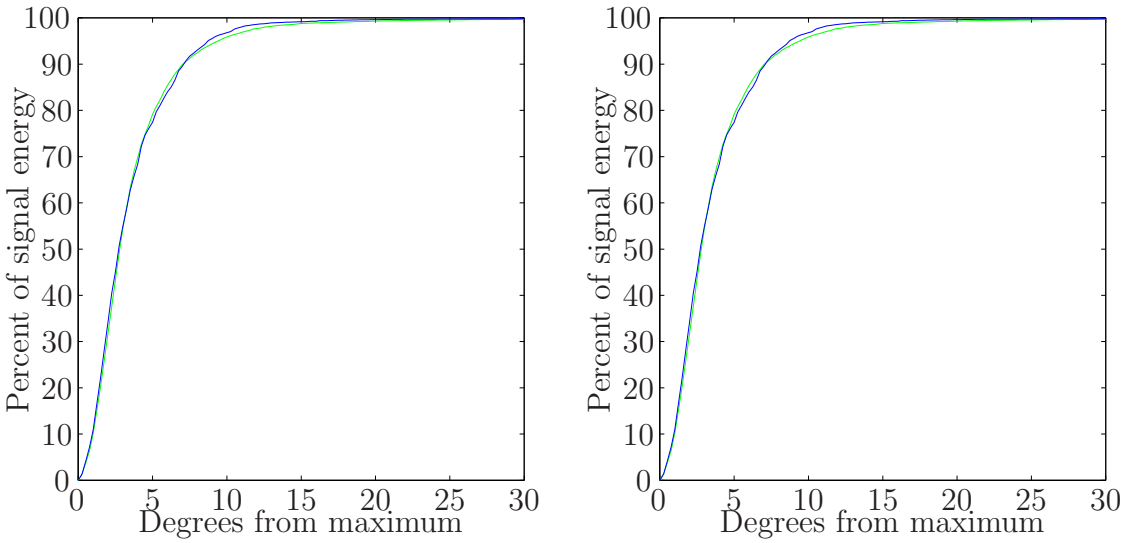
For each recording, the energy within angle δ of the estimated source location for each source is calculated by applying a binary bit mask, and the PSR calculated.

Results

The PSR for both sources is shown in figure 5.1. The performance for both sources in both domains is very similar. This is result is as expected if the directional performance is to be attributed solely to the microphone array performance.

Following the calculation of the directional information d in the transform domains, a normalised 3D spherical geodesic histogram was created for each source to plot the spread of the signal energy. Figure 5.2 shows the results for both sources in both domains. The source energy can be seen to drop away rapidly from the peak, with the bin containing the maximum collecting over 60% of the signal energy in all cases.

Approximately 90% of the signal energy lies within 7 degrees of the peak for each source. This result is comparable with the directional performance of the array used in [1], which used B-format arrays in the wavelet domain to localise sources. The accuracy achieved in the referenced work is also 7 degrees, although this is for sources lying on a 2D plane.



(a) Plot for source 1

(b) Plot for source 2

Figure 5.1: Plot of energy remaining following masking for increasing δ. The STFT results are plotted in blue, the DTCWT plotted in green

For localisation, the method proposed here therefore provides comparable performance in 3D to this published method in 2D, whilst managing a reduction in computational complexity as previously discussed.

This localisation performance is dependant on the assumed ω-disjoint properties of the sources. For comparison with figure 5.2, figure 5.3 is provided, showing the localisation for two plant recordings and two bird recordings from the previous chapter. The bird recordings have good *omega*-disjoint properties, the plant recordings poor ω-disjoint properties. The effect on the peak spreading caused by interference between time frequency components is clearly visible in the case of the plant mixture.

5.4.2 Separation performance

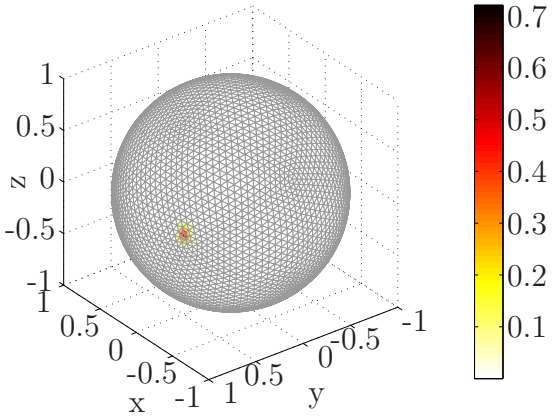
The aim of this experiment is to test the performance of the proposed separation method for sources recorded using using a B-format array, and the effect of the choice of threshold δ on that performance.

The previous experiment has the distribution of source energy for a single source, and it would be reasonable to expect a similar threshold to produce optimum performance. However, as the sources are only approximately ω-disjoint orthogonal, peak spreading will occur to some degree for many of the time-frequency points within the transform domains, and the effect this has on the separation performance needs to be investigated.

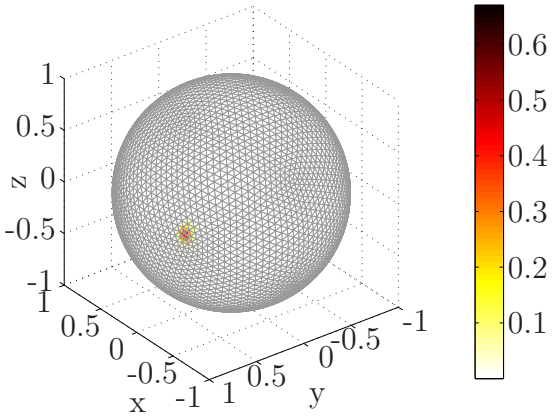
Methodology

The two male speakers recorded independently under anechoic conditions used for the previous experiment are used again here. The four channels of the B-format audio for each source were then transformed into both transform domains (STFT and DTCWT), and summed to form a B-format mixture within each transform domain.

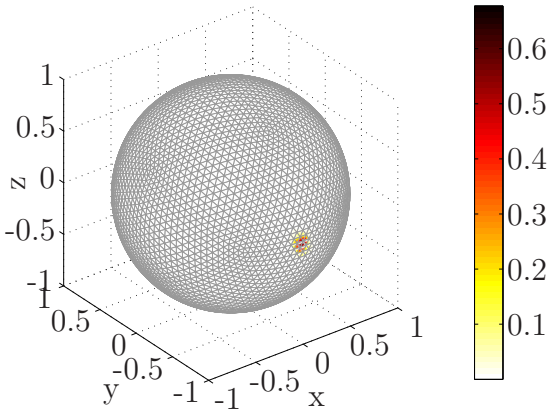
The benefits of this approach as opposed to the recording of both speakers together is that to provide performance metrics for the separation algorithm, the original source



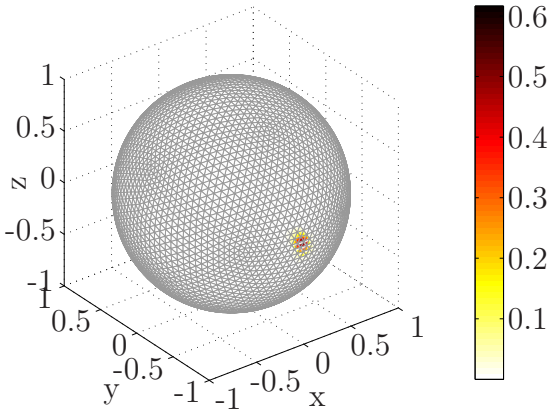
(a) Source 1 in the STFT domain



(b) Source 1 in the DTCWT domain

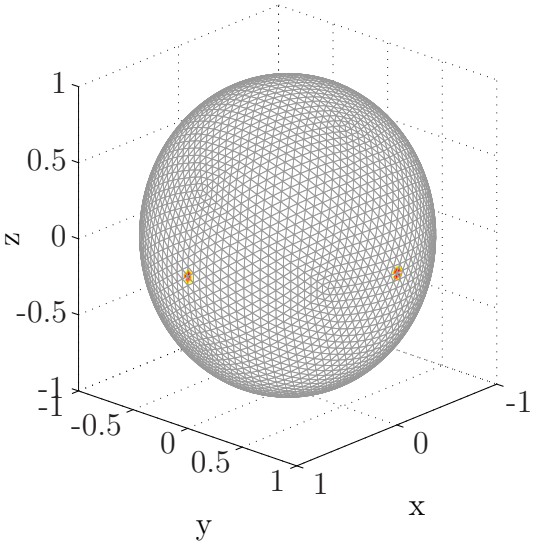


(c) Source 2 in the STFT domain

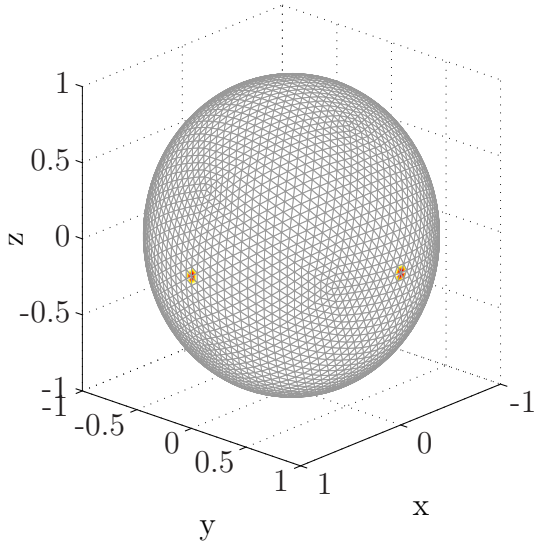


(d) Source 2 in the DTCWT domain

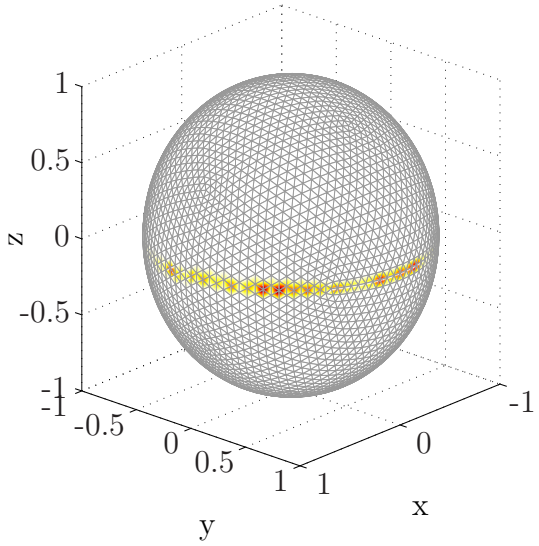
Figure 5.2: Directional histogram of normalised source energy plotted on a 3D geodesic grid



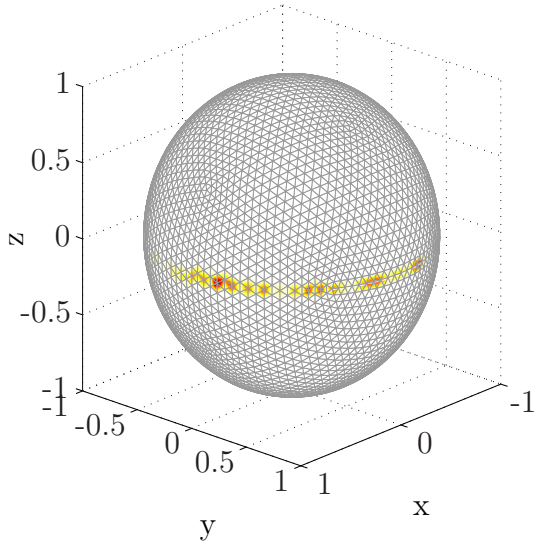
(a) Location estimation for a mixture of two bird sources in the STFT domain



(b) Location estimation for a mixture of two bird sources in the DTCWT domain



(c) Location estimation for a mixture of two plant sources in the STFT domain



(d) Location estimation for a mixture of two plant sources in the DTCWT domain

Figure 5.3: Location estimation showing peak spreading caused by poor ω-disjoint attributes

must be known. This approach is valid, as sound pressure levels are additive, i.e, $f(a+b) = f(a)+f(b)$. This additive property is also possessed by both the STFT and the DTCWT.

A directional estimate is then calculated for every time-frequency point in the transformed domain using the B-format mixture.

For a known or estimated location for each source to be separated from the mixture, the distance from each time-frequency point was then calculated.

A mask was formed for each source for varying δ angular thresholds. These masks were then applied and estimates of each source were then found for each threshold.

Finally, the SIR performance metric discussed in this chapter, and the SDR performance metric discussed in chapter 1 were calculated for each mask.

Pseudo-code for this algorithm is shown below

```
% STEP 1 - Record sources
  for n : 1 to number of sources N
    [w_n, x_n, y_n, z_n] = record speaker{s_n}

% STEP 2 - Transform recordings
  for n : 1 to number of sources N
    [X_n, X_n, Y_n, Z_n] = Transform{w_n, x_n, y_n, z_n}

% STEP 3 - Form mixtures
  W = sum_1^N W_n        X = sum_1^N X_n
  Y = sum_1^N Y_n        Z = sum_1^N Z_n

% STEP 4 - Calculate directional information
  d = Directional Estimation{W, X, Y, Z}
```

% STEP 5 - Find angular separation from source locations

for n : 1 to number of source locations N

$\mathbf{d_n} = \arccos(\hat{\mathbf{e}_\mathbf{n}} \cdot \hat{\mathbf{d}})$

% STEP 6 - Calculate bit mask

for δ : 0 to δ MAX

$\Phi_{n,\delta} =$ Calculate Mask$\{\mathbf{d_n}, \delta\}$

% STEP 7 - Calculate source estimates

$S_{(n,\delta)}^{'} = \Phi_{(n,\delta)} \; W$

% STEP 8 - Calculate SIR metrics

$SIR =$ SIR Calculation$\{S_{(n,\delta)}, S_{(m \neq \; n,\delta)}\}$

% STEP 9 - Find time domain source estimates

$s_{(n,\delta)}^{'} =$ Inverse Transform$\{S_{(n,\delta)}^{'}\}$

% STEP 10 - Calculate SDR metric

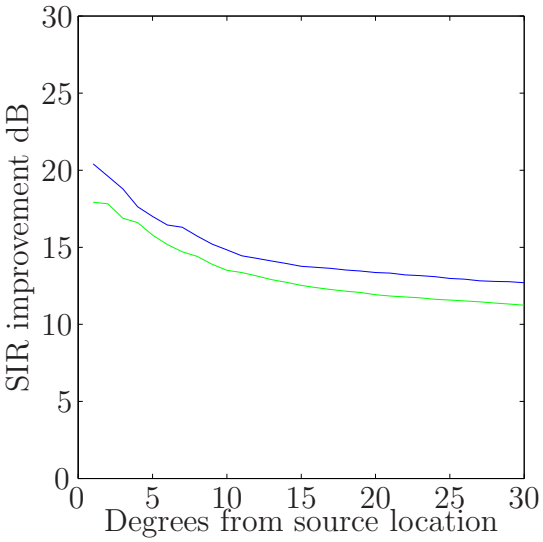
$SDR =$ SDR Calculation$\{s_{(n,\delta)}^{'}, s_{(n,\delta)}\}$

Results

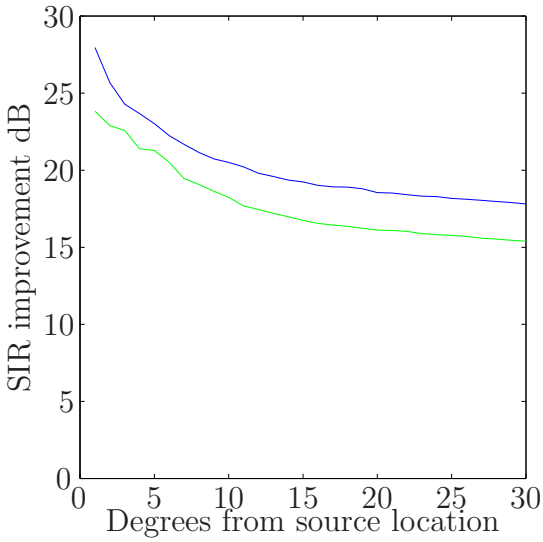
The SIR improvement metric is plotted in figures 5.4(a) and figure 5.4(b). Contrary to what may be expected following the outcome of the previous experiment, SIR improvement is greatest for small values of angular threshold δ for both sources.

This is seemingly at odds with the previous experiment, which showed that larger values of δ up to approximately 10 degrees from the source location are required to capture most of the source energy, with 90% lying within 7 degrees.

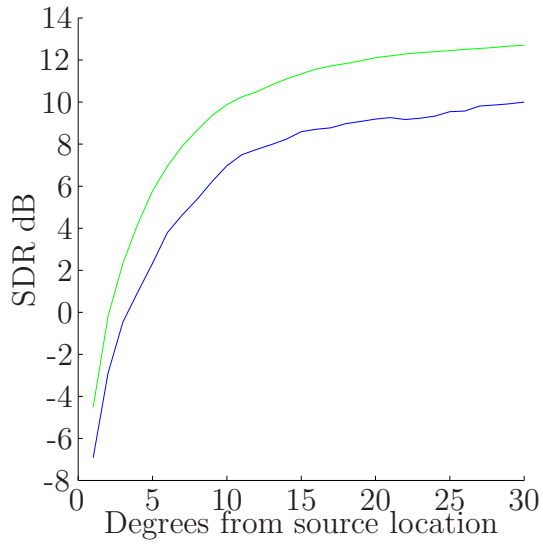
This apparent contradiction can be explained by the interference terms, which increases in magnitude toward the interfering source location. The interference causes peak spreading, with the angular deflection being proportional to the magnitude of the



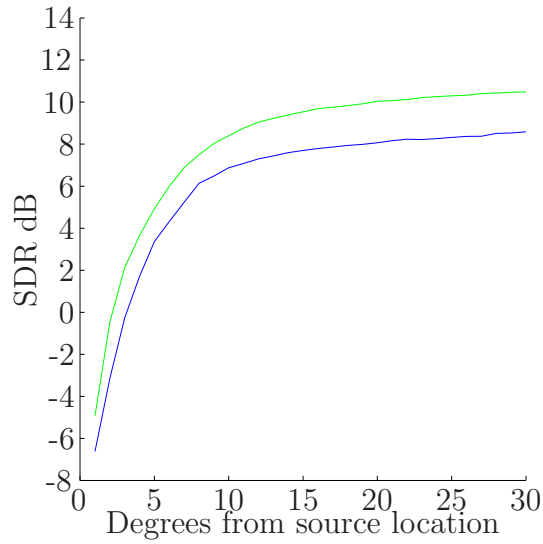
(a) SIR improvement for source 1



(b) SIR improvement for source 2



(c) SDR for source 1



(d) SDR for source 2

Figure 5.4: Performance metrics for the separation of two recorded B-format speech mixtures for varying δ threshold. STFT results are plotted in blue, DTCWT results in green

interference.

Once the angular threshold δ is large enough to take into account the peak spreading caused by both the microphone and interference, the SIR interference stabilises. This roughly constant performance value is due to the inherent ω-disjoint orthogonality between the sources.

If δ were to be increased such that the interfering source lay within the threshold, the the SIR metric would decrease accordingly. Again, this is intuitive, as directional separation is no longer effective.

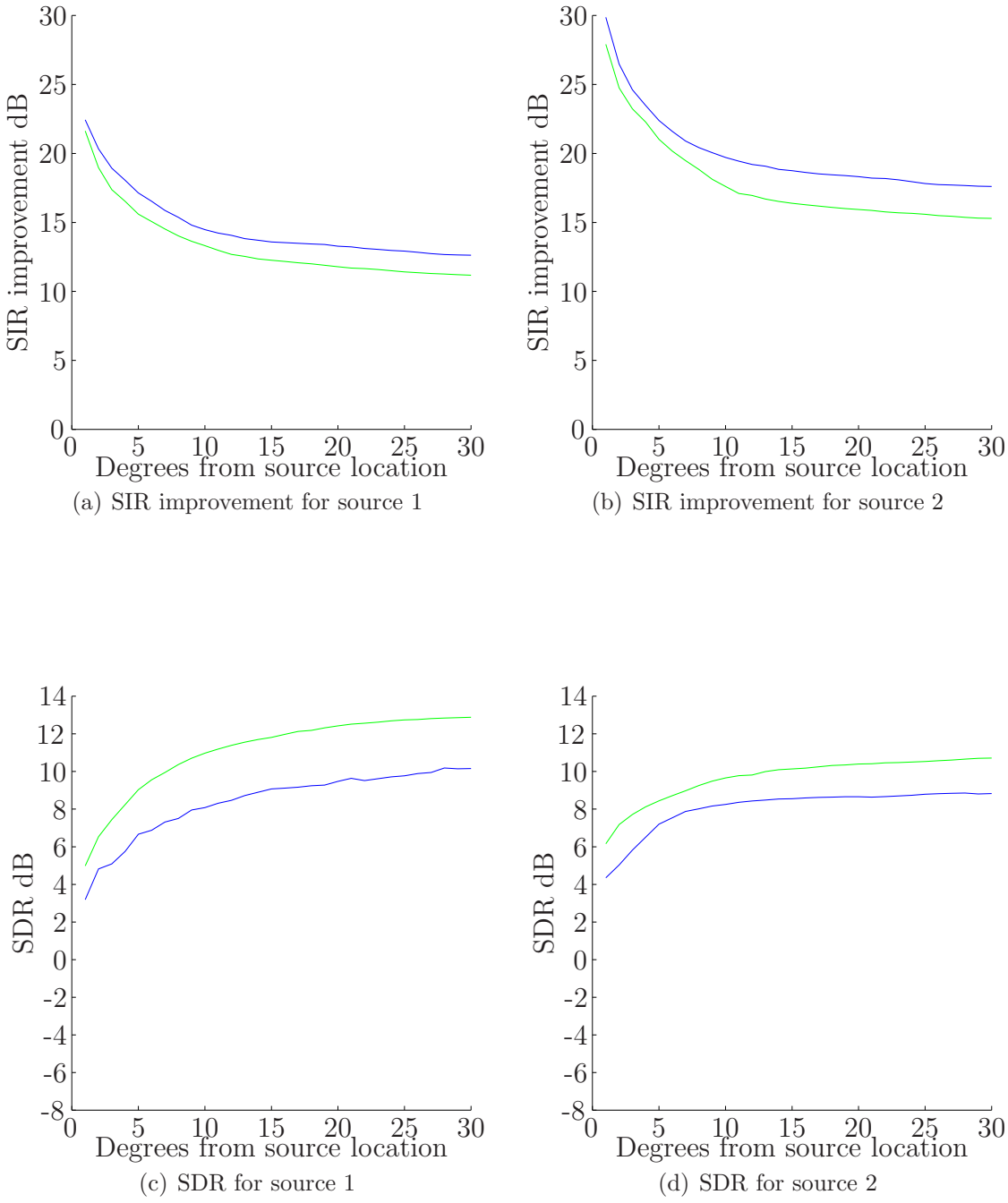
The performance difference between the STFT and the DTCWT transforms converges to a uniform offset as angular threshold is increased. The STFT provides a 3 dB performance margin when compared to the DTCWT. This is attributed to the suitability of the window size chosen for this particular separation application.

The results for the SDR metric are plotted in figure 5.4(c) and figure 5.4(d). Unlike the SIR metric, SDR improves with an increasing δ angular threshold before converging on a steady value.

An SDR of 0 dB is achieved within a few degrees of the source location, confirming the results of the previous experiment that a significant proportion of the source energy lies close to the sources true location.

For this metric, the DTCWT outperforms the STFT. This is significant, as this metric compares the estimated source with the original source directly in the time domain, taking into account all filtering and transformation stages.

The disparity between the performance of the transforms between the SIR and SDR metrics is attributed to the inverse transformation stage.



(a) SIR improvement for source 1

(b) SIR improvement for source 2

(c) SDR for source 1

(d) SDR for source 2

Figure 5.5: Performance metrics for the separation of two ideally mixed B-format speech mixtures for varying δ threshold. STFT results are plotted in blue, DTCWT results in green

5.4.3 Comparison of performance to the ideal B-format model

The aim of this third experiment is to gauge the extent to which the performance seen in the previous experiment is inherent in the separation algorithm, and to what extent the microphone array performance impacts the separation performance. This is to ensure that the algorithm is not unduly compromised by a poorly performing microphone array, which may easily be replaced with any commercially available model.

Methodology

The methodology for this experiment is almost identical to that of the previous experiment. The B-format recordings are replaced with an artificially created B-format mixture created using the omnidirectional source. These are mixed using the ideal B-format model to place the sources at the estimated source locations found in the previous experiments, replacing the microphones directional response with the ideal model.

This can be seen from the pseudo-code in the previous section as a modification to step 3.

Results

The results for the SIR and SDR metrics are plotted in figure 5.5. The results for all metrics are improved for small values of angular threshold δ. The SIR improvement is approximately 3 dB at 1 degree.

This improvement is quickly lost at increasing values of δ. The results for values of δ where the performance is settling are similar to those achieved in the previous experiment, lending weight to the conclusion that performance in this area is dependant on the underlying ω-disjoint orthogonality between the sources.

Therefore, for environments where sources are well spaced, microphone choice is non-critical. Where sources are closely spaced at less than about 15 degrees apart, the

directional performance of the microphone array becomes increasingly important in maximising the separation performance using the proposed algorithm.

5.5 Chapter synopsis

This chapter has built upon the concept of using ω-disjoint orthogonality as a basis for source separation, the suitability of which for soundscape analysis was explored in the preceding chapter.

A novel directional separation algorithm has been proposed. The key benefits of the algorithm are summarised as:

- Compact COTS (commercial off-the-shelf) microphone array
- Capable of separation in three dimensions
- No limit to the number of sources separable
- Based on the well proven concept of time-frequency binary masking
- Either STFT and DTCWT can be chosen to be used, depending on application.

Experiments for a typical speech application have been conducted to characterise the algorithm's performance. This has lead to the provision of SIR, SDR and PSR metrics for the separation performance for the new algorithm.

The separation algorithm has also been shown to provide a locational accuracy equal to the results published in [1], for a reduction in computational cost.

The DTCWT has been shown to be an effective alternative to the STFT, providing a superior result for the SDR performance metric in this application.

5.6 Chapter Bibliography

[1] C. A. Dimoulas, G. V. Papanikolaou, K. A. Avdelidis, and G. M. Kalliris. Sound source localisation and B-format enhancement using soundfield microphone sets. In *Audio Engineering Society 122th Convention*, May 2007.

[2] M. Gerzon. Periphony: With-height sound reproduction. *Journal of the Audio Engineering Society*, 21(1):2–10, 1973.

[3] M. Gerzon. The design of precisely coincident microphone arrays for stereo and surround sound. In *50th Convention of the Audio Engineering Society*, 1975.

[4] A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '00*, volume 5, pages 2985–2988, 5–9 June 2000. doi: 10.1109/ICASSP.2000.861162.

[5] J. Merimaa and V. Pulkki. Spatial impulse response rendering. In *7th International conference on digital audio effects*, pages 139–144, October 2004.

[6] R. Mukai, H. Sawada, S. Araki, and S. Makino. Real-time blind source separation and DOA estimation using small 3-d microphone array. In *International Workshop on Acoustic Echo and Noise Control*, pages 45–48, September 2005.

[7] V. Pulkki. Directional audio coding in spatial sound reproduction and stereo upmixing. In *Audio Engineering Society 28th International conference*, pages 7–1, June 2007.

[8] V. Pulkki. Spatial sound reproduction with directional audio coding. *Journal of the Audio Engineering Society*, 55(6):503–516, June 2007.

[9] V. Pulkki and C. Faller. Directional audio coding: Filterbank and STFT-based design. In *Audio Engineering Society 120th Convertion*, page 6658, May 2006.

[10] V. Pulkki, J. Merimaa, and T. Lokki. Reproduction of reverberation with spatial impulse response rendering. In *Audio Engineering Society 116th Convertion*, page 6057, May 2004.

[11] K. Teramoto, T. Khan, and S. I. Torisu. Acoustical blind source separation based on linear advection. In *SICE annual conference*, pages 1–118, 2007.

Chapter 6

Clustering Audio sources

Contents

6.1 Introduction

The separation algorithm developed in the previous chapter is dependant on the sound source's location being known *a-priori*. The purpose of this chapter is to investigate techniques to provide DOA estimates for the audio source clusters. It should be noted that although this work is directly relevant to the audio source separation discussed in

the previous chapter, here the problem of clustering DOA vectors will be treated as an independent problem where possible.

Firstly a histogram approach, clustering mixing parameters is considered, and an approach extending this to clustering on a 3 dimensional spherical surface is considered. Finally a novel approach using a plastic self-organising map (PSOM) is considered that has been shown to promising in 2D radar applications, and this is again extended to the 3 dimensional case, mapping the coordinate system from Cartesian space to a unit sphere.

6.2 Histogram Approach

6.2.1 Background

Using histograms as a tool for directional clustering in sparse source separation is not a new concept. In [5], a 2D histogram using amplitude and delay differences between two omni-directional sensors was used to localise sources in the 2D half plane. Perhaps it should be noted that this method implicitly assumes convolutive or echoic sound propogation. See chapter 2.

This histogram approach combined with a suitable peak detection algorithm, or even a simple thresholding function, promises to provide a simple to implement approach to the problem of source clustering. However, there are some considerations in the construction of a histogram for three dimensional vectors that require addressing.

The directional information calculated as part of the separation algorithm, equation 5.4 consists of Cartesian vectors, the magnitude of which is proportional to the energy in each time-frequency bin in the transform domain.

Finding the unit norm for each of these directional vector maps the clustering problem onto a unit sphere, simplifying the clustering task.

6.2.2 Latitudinal-Longitudinal bound bins

Using spherical coordinates to describe the DOA vectors is a logical step. The azimuth and elevation for each vector, equivalent to the latitudinal and longitudinal location on a sphere, can be used to form the bins of the histogram.

Spherical coordinates

The direction of arrival vectors must be converted from the Cartesian coordinate system into the spherical coordinate system. This is performed using the transform shown in equation 6.1

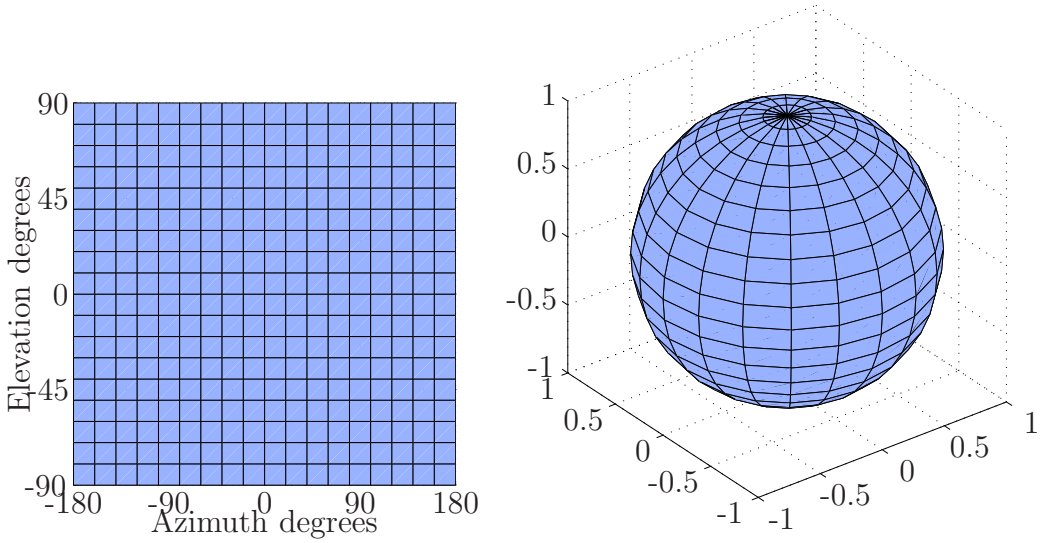
$$\begin{vmatrix} \theta \\ \lambda \\ r \end{vmatrix} = \begin{vmatrix} \arccos\left(\dfrac{x}{\sqrt{x^2+y^2}}\right) \\ \arccos\left(\dfrac{z}{\sqrt{x^2+y^2+z^2}}\right) \\ \sqrt{x^2+y^2+z^2} \end{vmatrix} \tag{6.1}$$

Angular Resolution

Using a regularly spaced grid of bins based on latitudinal and longitudinal coordinates provides a simple means of assigning a particular vector into the appropriate histogram bin, requiring only comparison with the x and y axis grid lines. This latitudinal-longitudinal grid is a 2D mapping of a 3D object and, once transformed back to the 3D sphere, undesirable traits in the regularity of the bin becomes apparent.

Figure 6.1shows this non-uniform bin size on the 3D sphere. The is particually marked near the poles, where resolution is much higher than nessesary.

The effect of these non-uniform bin sizes is that for unit vectors distributed normally over the unit sphere surface, the vector count attributed to each bin is non-uniform. A possible solution to this is scaling the bin count by the bin area to give a uniform magnitude. This doesn't remove the drawback that in order to achieve a fine resolution



(a) A regular regularly spaced grid of azimuth and elevation values

(b) The same grid of azimuth and elevation values mapped to a unit sphere

Figure 6.1: Histogram bins using spherical coordinates

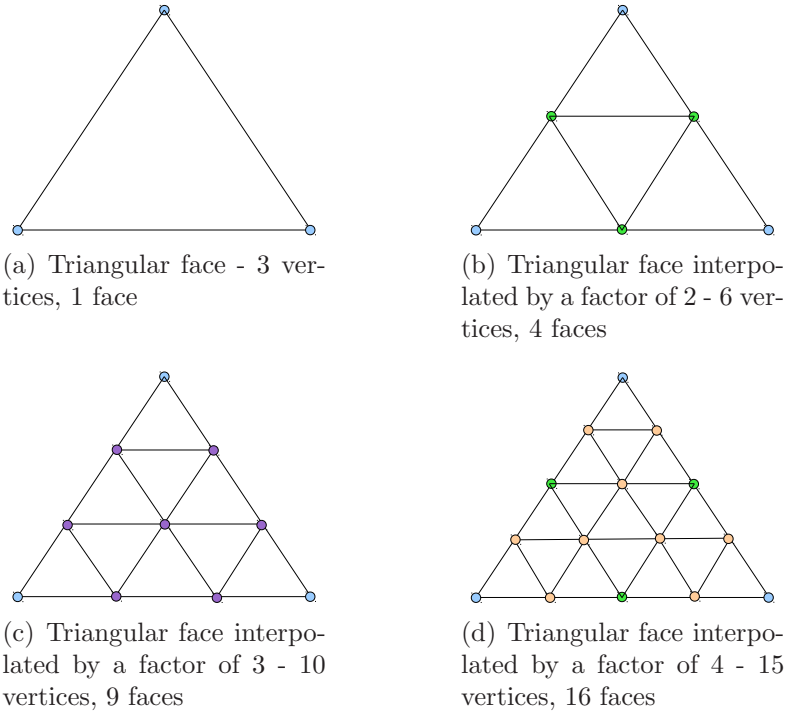
at the equator, more bins than required for this resolution will be generated near the poles.

6.2.3 Geodesic Histogram

A solution to this problem of non-uniform bin area is to abandon the approach based on a regularly spaced grid in 2D, and instead form a histogram based on a geodesic grid that approximates a sphere.

The approach used here to form such a grid is based on interpolating an icosahedron. Each edge of a face of the icosahedron is interpolated by factor I to give $3I$ edges around the perimeter of each original triangular face. This interpolation of the face is shown in figure 6.2 and the isohedron is shown in figure 6.3(a).

The number of faces this subdivision of the original triangular face provides can be calculated as a function of the interpolation factor. This is repeated for all faces, and the total number of faces for a given interpolation factor I can be given by equation 6.2 below. The number of vertices needed to describe these faces is given in equation 6.3.



(a) Triangular face - 3 vertices, 1 face

(b) Triangular face interpolated by a factor of 2 - 6 vertices, 4 faces

(c) Triangular face interpolated by a factor of 3 - 10 vertices, 9 faces

(d) Triangular face interpolated by a factor of 4 - 15 vertices, 16 faces

Figure 6.2: Interpolation of a triangular face

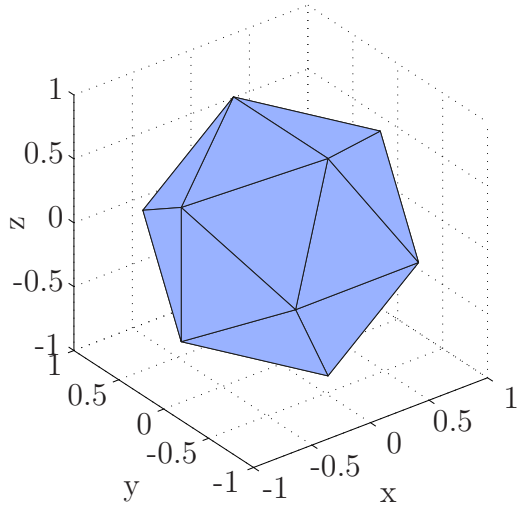
$$\text{Faces} = 12 \times I^2 \tag{6.2}$$

$$\text{Vertices} = 12 + 10\left(2\sum_{i=1}^{I+1}\{i\} - 3(i-1) - 6\right) \tag{6.3}$$

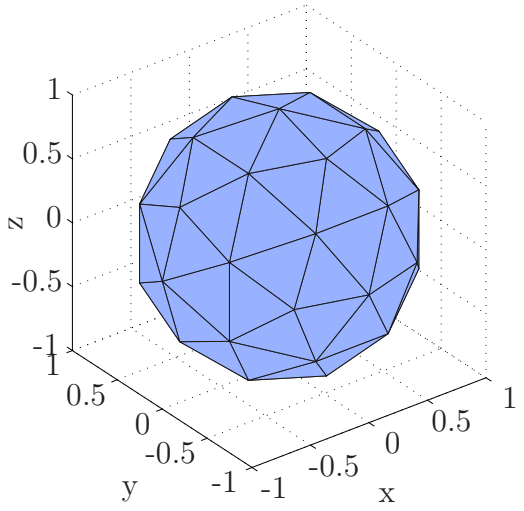
An approximation of a unit sphere can then be formed by the normalisation of the vectors describing the vertices onto a unit circle. Examples can be seen in figure 6.3. Following this transformation, vertices are not in general uniformly spaced. Code for the generation of a spherical geodesic grid for an arbitrary interpolation factor is included in appendix C.1

Bin assignment

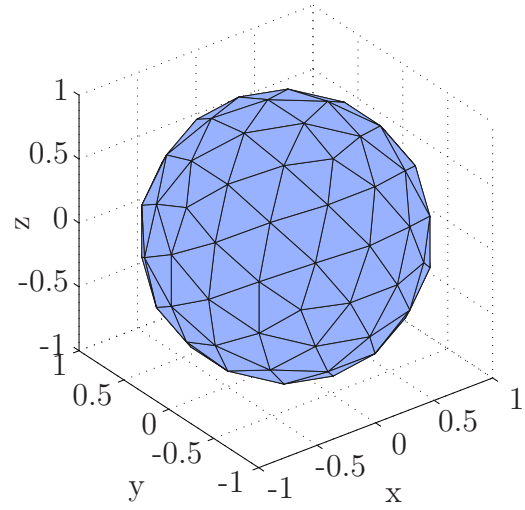
The directional vectors for the time-frequency points can be assigned to the closest vertex of a geodesic grid on the unit sphere.



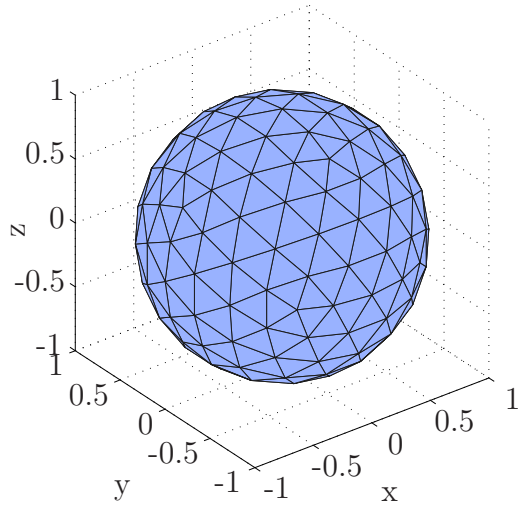
(a) Icosahedron - 12 vertices, 20 faces



(b) Icosahedron interpolated by a factor of 2 - 42 vertices, 80 faces



(c) Icosahedron interpolated by a factor of 3 - 92 vertices, 180 faces



(d) Icosahedron interpolated by a factor of 4 - 162 vertices, 320 faces

Figure 6.3: Creating a geodesic grid by interpolation of an icosahedron

The closest vertex can be calculated using the vector dot product. The angle θ between any two unit vectors $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ is given by

$$\theta = \arccos\left(\hat{\mathbf{v}}_1 \cdot \hat{\mathbf{v}}_2\right) \tag{6.4}$$

Comparing a directional vector in the time-frequency domain with each geodesic bin location allows the smallest angle to be found, and the directional vector to be assigned to that bin. The cos term can be removed to simplify the calculation, searching instead for the greatest dot product between the unit vectors.

If a higher resolution is required this exhaustive search becomes increasingly computationally expensive as the number of histogram bins increases. This can be mitigated by a factor of approximately 12 by first performing a search to find the three vertices of an icosahedron that describe the face containing the directional vector, and then performing a further search at higher resolutions only on vertices contained within this face. This recursive search can be extended into a tree structure if very high resolution is required. Source code for an exhaustive search algorithm is available in C.2

6.2.4 Estimating static source locations by peak estimation

Following the formation of a histogram, source locations are found by locating maxima within the histogram. Standard search algorithms such as Maximum likelihood may be used for this [6]. Alternatively, if there is *a-priori* knowledge of the number of sources N, and it is assumed these contain the majority of the power of any source within the soundscape, finding the N largest peaks is sufficient. This is the approach that was used in chapter 5, where results for location estimation for a two source mixture can be seen in figure 5.2

6.2.5 Varying source locations and numbers

Whilst the assumption that the source locations are static is valid in many practical solutions, some applications will have sources that vary in location over time. See chapter 1 for details of anticipated applications.

The above approach can be adapted for application to moving sources by estimating the source locations within a time window, in which they are assumed static. The Hamming window used for the STFT is ideal for such purposes, provided that in this interval both sources remain present.

For non-stationary sources that are intermittent, the challenge of estimating source locations increases dramatically, and must take into account a particle tracking approach, where a history for each particle is kept in memory, and if the source is present in an estimation window, this information is used to form an estimate of current position. This is an area for further research.

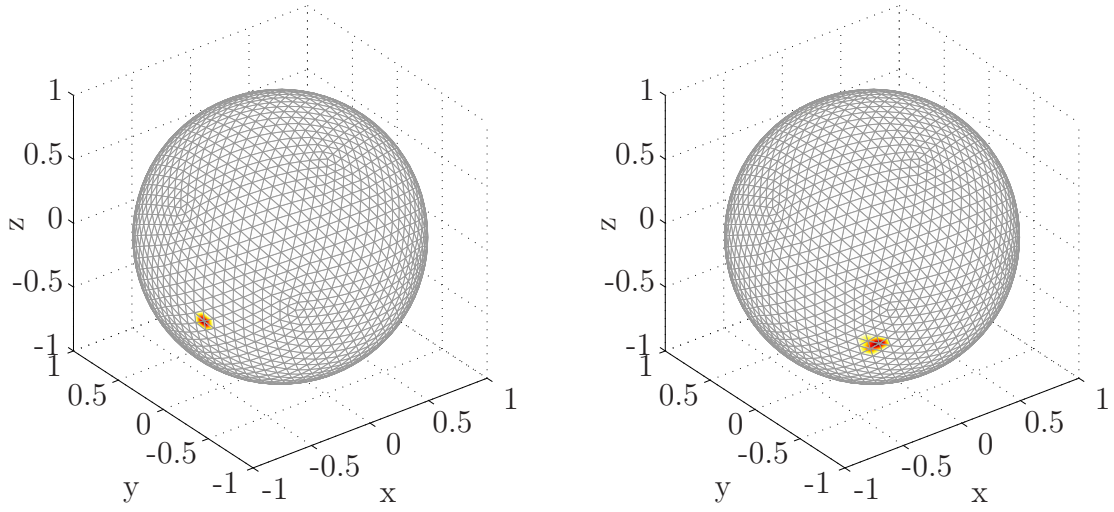
Experiment

A 2 kHz test tone was generated 1.5 meters from a B-format microphone (Soundfield ST350). The source was moved radially about the microphone for the duration of the recording to a total angular displacement of approximately 90 degrees. The recording environment was an echoic office measuring approximately 5m by 8m.

The signal was analysed in the STFT domain, and the peak approach was used to estimate the source location for each STFT window. A window duration of 0.1 s was used for the location estimation.

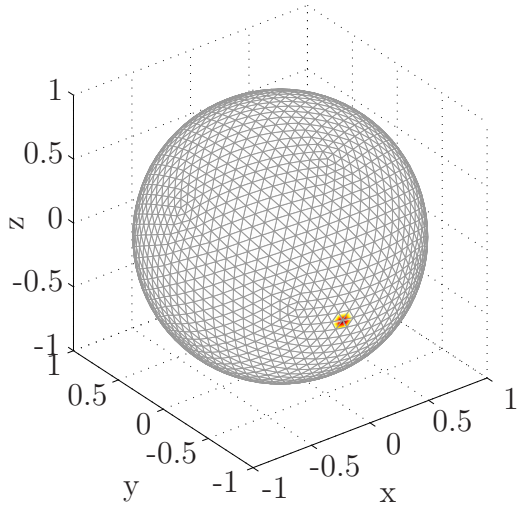
The resulting histogram is a 4D dataset which was viewed in the form of a video. The video shows strong identification of the source location and the algorithm can be seen to track the source radially in the correct location. Stills from the video are shown in figure 6.4 for times 1, 2, 3, and 4 s.

The azimuth and elevation for the estimated source positions are plotted in figures

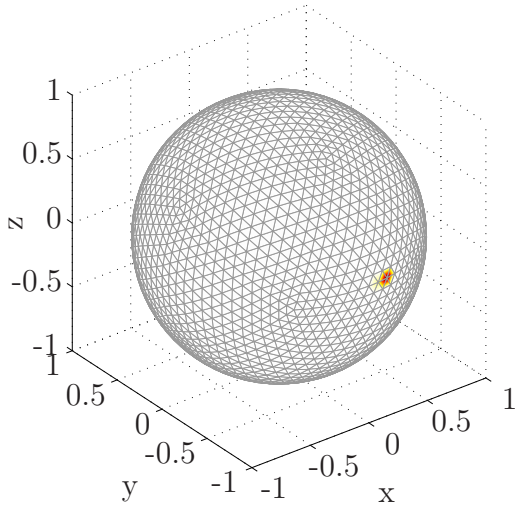


(a) Time 1s

(b) Time 2s

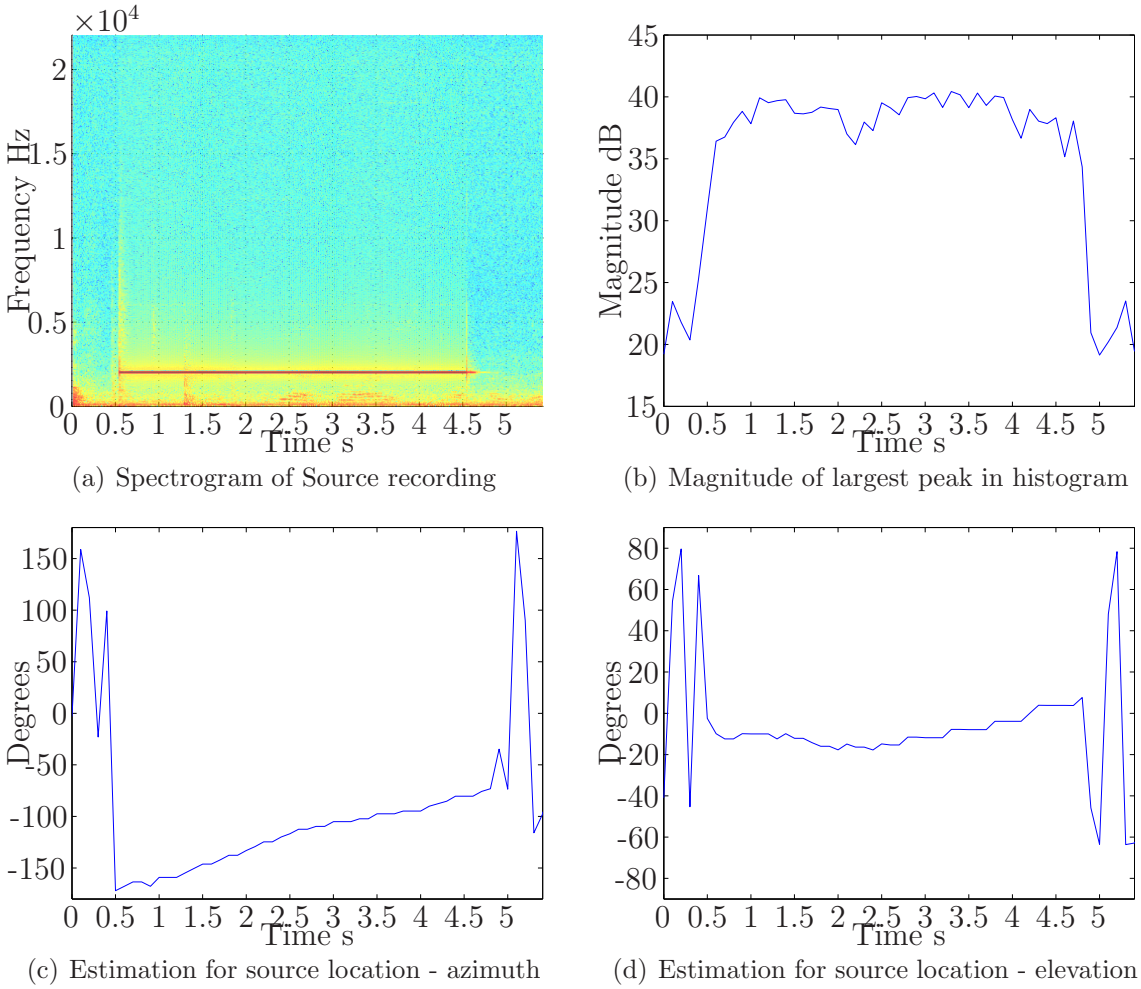


(c) Time 3s



(d) Time 4s

Figure 6.4: Directional tracking histogram for a non-stationary source in an echoic environment



(a) Spectrogram of Source recording

(b) Magnitude of largest peak in histogram

(c) Estimation for source location - azimuth

(d) Estimation for source location - elevation

Figure 6.5: Location estimate for a non-stationary source in an echoic environment

6.5(c) and 6.5(d) respectively. The magnitude of the largest peak detected is plotted in figure 6.5(b).

The azimuth and elevation vary significant during the first and last half seconds of the recordings. Figure 6.5(a) shows a spectrogram of the omnidirectional component of the recording. This period of ill-determined location can be seen to correspond with times when the source is not present.

For times whilst the source is present, both the azimuth and elevation can be seen to be consistently tracking the source. Whilst the crude setup of the experiment does not allow analysis of the accuracy of the localisation over time, an informal inspection confirms that the azimuth varies by approximately 90 degrees, whilst the elevation is approximately constant. This is in agreement with the experimental setup.

The magnitude of the maximum peak in the histogram plotted in figure 6.5(b) shows strong correlation with the spectrogram of the source in figure 6.5(a). Using an empirically set threshold of 30 dB for this magnitude response is a simple method for detecting the presence of the source, and hence bounding the times that the azimuth and elevation estimates are valid.

6.3 Clustering using a Plastic Self-Organising Map

The applicability of an alternative algorithm for clustering audio directional vectors has been investigated. The Plastic Self-Organising Map (PSOM) [2, 4, 3] is an adaptive learning algorithm for clustering multidimensional data.

The algorithm is an extension to the self organising map (SOM) proposed in [1]. The SOM is a grid of interconnected neurons initialised over the input space. Training is performed using exemplar datasets for different inputs. The aim is for the network to be able to identify, following training, the correct category for the input vector, assigning it to a group. This grouping can be considered analogous to clustering input data according to physical locations.

A PSOM differs from the classical SOM as the network of neurons is allowed to cleave, with each separated network representing a source location. This is fundamentally different to the SOM as, in effect, there are multiple, potentially overlapping networks operating on the same input space.

The PSOM also differs from the SOM as there is no training phase. Instead, the network continually adapts to the input, allowing the network to alter its structure as inputs move, cease to occur, or new sources appear. This behaviour is what makes the PSOM attractive for this clustering application, as the ability to morph and split allows sources to be tracked at they move, for new sources to be identified, and for sources no longer present in a soundscape to be disregarded.

A flow diagram for the operation of the PSOM is shown in figure 6.6. Each phase of network operation is discussed in detail in the following section.

6.3.1 PSOM Operation - Euclidean space

The PSOM operation is based on six user-defined variables, which are set empirically. These are:

a_n Threshold for neuron addition. If a new input is presented to the network at a distance from the closest neuron of greater than a_n, a new group of neurons is added to the network.

a_r Threshold for Link removal. When age of the link between neurons exceeds this threshold, the link is removed. If any neuron is left with no links, it is also removed.

b_a Ageing parameter. After every iteration of the algorithm, links are aged by a function of parameter b_a.

b_c Link scaling parameter. Following the update of the neurons position, the age of the links between them is reduced by factor b_a.

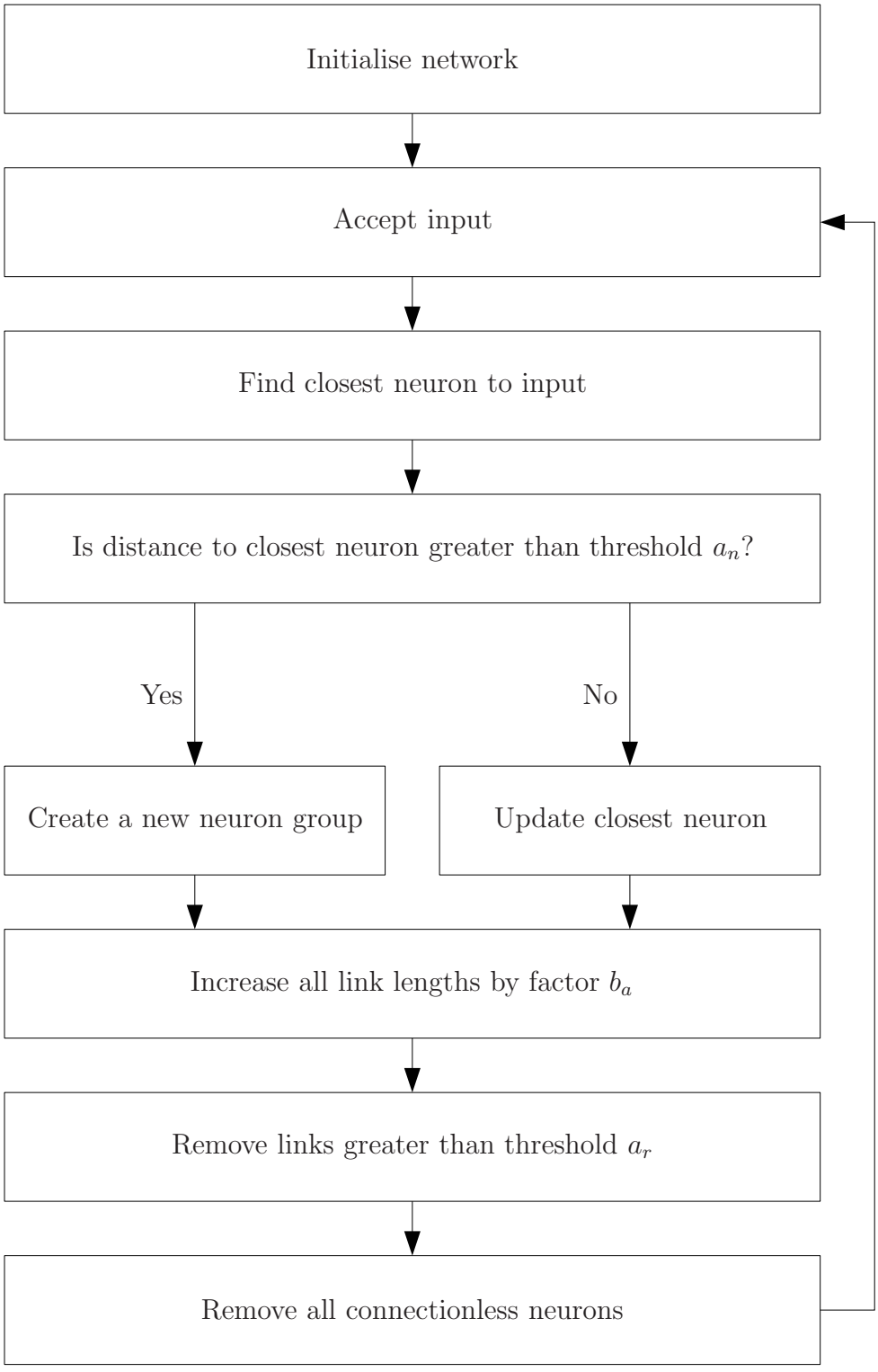


Figure 6.6: Flow representation of PSOM algorithm

b_l Scaling parameter. The neurons connected to the closest neuron are updated by a distance towards the winning neuron as a function of the distance between the neurons and b_l.

b_v Scaling parameter. If the input is less than threshold a_n from the closest neuron, the closest neuron is updated towards the input by a function of the scaling factor and the distance from the input to the closest neuron.

Initialisation

The network is seeded with three neurons x_1, x_2, x_3 connected via two links $c_{1,2}$ and $c_{1,3}$. The neurons are created at a random point in the input space. The neurons may be represented as vectors from a nominal origin.

Finding the distance from the input to the closest neuron

The network accepts an input u. The distance Δx_n to each neuron in the network is then calculated by finding the Euclidean norm for the vector between the input u and neuron x_n.

$$\Delta x_n = \|u - x_n\| \tag{6.5}$$

The closest input to u, $\min(\Delta x_n)$, is denoted the network focus z with distance to the input vector Δz.

Applying the threshold decision

Next, a decision is made about the update to make to the network.

$$\Delta z < a_n \begin{cases} true & \text{Update the focus} \\ false & \text{Add new group to the network} \end{cases} \tag{6.6}$$

Update the focus

The focus is updated to make it more similar to the input u by a function of the scaling value b_v and the distance Δz.

$$z = z + b_v \times \Delta z \tag{6.7}$$

All links from the focus are then refreshed. The new link lengths are defined as a function of the maximum link length and the distance between the focus and connected neurons. Lang and Warwick [4] note that the input space is bounded between 0 and 1. Input spaces with higher bounding values should apply an additional scaling factor c_{z_n} according to this equation.

$$c_{z_n} = a_r \times \|z - x_n\| \tag{6.8}$$

All neurons connected to the focus by a link are then updated. This update is a function of scaling factor b_c, the distance between the neurons and the focus, and also the age of the link connecting them.

$$x_n = b_c \times c_{n,z} \times \|z - x_n\| \tag{6.9}$$

Add a new group to the network

A new group is added to the network. Three new neurons connected by two links are placed around the input u. A link is also created to the focus z, such that one of the newly created neurons has three linked neurons connected to it.

Update the links between neurons

All links in the network are aged by a small amount. This is so that links that are not connected to a focus will age, and will eventually be removed from the network, allowing temporal learning.

$$c_{n,m} = c_{n,m} + b_a \tag{6.10}$$

Link removal

All links in the network are then compared against the maximum link length a_r. Links greater than this threshold are removed from the network.

$$c_{n,m} > a_r \begin{cases} true & \text{Remove } c_{n,m} \\ false \end{cases} \tag{6.11}$$

Neuron removal

Finally, all neurons in the network are checked for links. If a neuron has no links to other neurons, it is removed from the network.

The network has then finished updating and is ready to accept the next input.

6.3.2 Modified PSOM Operation - surface of unit sphere

The PSOM's operation, as published in [2], and described above is not directly applicable to clustering the input data from the separation algorithm. This is because the PSOM assumes input data and neurons are in Euclidean space. In this application,

the input vectors form a special case, they are all constrained to lie on the surface of a unit sphere.

The PSOM algorithm above can be adapted for this special case in such a way that the neurons are also constrained to the surface of a unit sphere. This requires the equations governing the update of neurons and links to be rewritten. The modified algorithm is considered below.

Initialisation

The network is seeded with three neurons connected via two links. The neuron's location is constrained to a unit sphere.

Finding the distance from input to closest neuron

Following the acceptance of the input unit vector $\hat{u}$, the distance $\Delta\hat{x_n}$ on the surface between $\hat{u}$ and all neurons $\hat{x_n}$ is found. This distance is the arc length between the vectors, and can be calculated as:

$$\Delta x_n = \arccos\left(\hat{x_n} \cdot \hat{u}\right) \tag{6.12}$$

The closest input to $\hat{u}$, $\min(\Delta\hat{x_n})$, is denoted the network focus $\hat{z}$ with a distance of $\Delta\hat{z}$ to the input vector as before.

Applying the threshold decision

The decision process is identical to the previous description, although a suitable value for threshold a_n has to be chosen for the new topology.

Update the focus

The focus is updated as previously to make it more similar to the input $\hat{u}$ by a function of the scaling value b_v and the distance $\Delta\hat{z}$. The scaling value b_v describes the percentage of the distance between the focus and input that the focus will be updated by. It is unnecessary to alter this for the new topology.

The equation for calculating the updated vector for $\hat{z}$ is altered. The previous method updated z in a direct line through Euclidean space, whilst the update here must follow the unit sphere surface.

First a unit axis a perpendicular to the input vector $\hat{u}$ and focus $\hat{z}$ is found using the vector cross product

$$a = \frac{\hat{z} \times \hat{u}}{\|\hat{z} \times \hat{u}\|} \tag{6.13}$$

This can then be used to define a rotation operator $R(\theta)$

$$R(\theta)=\begin{bmatrix} \cos(\theta)+a_x^2(1-\cos(\theta)) & a_x a_y(1-\cos(\theta))-a_z\sin(\theta) & a_x a_z(1-\cos(\theta))+a_y\sin(\theta) \\ a_y a_x(1-\cos(\theta))+a_z\sin(\theta) & \cos(\theta)+a_y^2(1-\cos(\theta)) & a_y a_z(1-\cos(\theta))-a_x\sin(\theta) \\ a_z a_x(1-\cos(\theta))-a_y & a_z a_y(1-\cos(\theta))+a_x\sin(\theta) & \cos(\theta)+a_z^2(1-cos(\theta)) \end{bmatrix} \tag{6.14}$$

where θ is defined as:

$$\theta = b_v\Delta\hat{z} \tag{6.15}$$

A new location for the updated focus can then be found using:

$$\hat{z} = R(\theta)\hat{z} \tag{6.16}$$

All the links are refreshed as before, using the new equation for the distance calculation between the focus and each connected neuron.

The position of the connected neurons is then updated using equations 6.13 and 6.14. θ is defined here as:

$$\theta = b_c \times c_{n,z} \times \arccos\left(\hat{z} \cdot \hat{x_n}\right) \tag{6.17}$$

The remainder of the PSOM operation is the same as before, although adjustments to the network variables need to be made to scale the network appropriately from a unit bounded Euclidean space to operating on the surface of a unit sphere. These variables are set empirically.

6.3.3 Implementation and analysis

A PSOM topology was implemented in C++. The implementation for this application is similar to the one in [4]. The exception to this is the neurons are not placed in Euclidean space, instead they are constrained to the surface of a unit sphere using the methodology described above.

Trial audio datasets of single sources in B-format, including a microphone response were used to test the performance of the PSOM. No empirically determined set of values for the network variables could be found that gave a stable clustering response.

Instead the network tends to diverge to one of two states. If the variables are set such that the network favours neuron creation, then the noise levels present in the audio mixtures quickly cause the network to spread over the entire surface of the sphere. Conversely, if the network is made to adapt rather than add, by increasing threshold a_n, the PSOM fragments, and is left with many networks overlapping the same input space. Because of this overlap, further inputs keep all networks from ageing and subsequent removal.

For a PSOM to be an effective method for clustering of audio data, the parameters need to be altered to be a function of the network state. Well established networks benefit from parameters that do not encourage new neuron growth, and the scaling parameters should be reduced to account for the successful history of well established neurons. Newly cleaved networks would benefit from high scaling values, to quickly move the new network away from its parent to help avoid overlapping networks.

How to achieve these suggestions for improvement is not apparent, and is an area where significant further research is possible.

Another concept that may help the stability of the networks is selecting the winning neuron for an input as a function of both this distance between the input and neurons, and also the neuron's history. This would help to prevent the networks overlapping, as established neurons would be more likely to win inputs, aiding the removal of neurons from other networks amongst the successful network.

6.4 Chapter synopsis

This chapter has introduced two contrasting approaches to the problem of clustering directional vectors.

Using histograms to find source locations by calculating the maximum peaks for a known number of static sources is the simplest method to implement. In terms of viability, this is sufficient for several of the applications considered in chapter 1, where a long term study of the noise level of 1 or more static installations is considered, such as PPG 24 or BS 4142 applications.

The approach based on using a PSOM to track a dynamic soundscape has the potential to be applicable to a much wider range of possible applications. An extension to the methodology to allow the PSOM to be used for clustering inputs on a unit spherical surface has been presented. For this to become a viable clustering method, further research is needed into the field of PSOMs, in particular a method for moderating the network parameters dynamically based on the network state is needed, as it is believed that there is no single set of parameters that will cause the network to operate efficiently under all input conditions.

6.5 Chapter Bibliography

[1] T. Kohonen. *Self organising maps*. Springer, 3rd edition, 2001.

[2] R. Lang. Initial study into the plastic self organising map. Technical report, Dept. Cybernetics, University of Reading, May 2001.

[3] R. Lang. *The Plastic Self Organising Map*. PhD thesis, Dept. cybernetics, University of Reading, 2003.

[4] R. Lang and K. Warwick. The plastic self organising map. In *the 2002 International Joint Conference on Neural Networks*, pages 727–732, 2002.

[5] S. Rickard and F. Dietrich. Doa estimation of many w-disjoint orthogonal sources from two mixtures using duet. In *Tenth IEEE Workshop on Statistical Signal and Array Processing*, pages 311–314, 2000. doi: 10.1109/SSAP.2000.870134.

[6] S. Rickard, R. Balan, and J. Rosca. Real-time time-frequency based blind source separation. In *in Proc. of Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 651–656, 2001.

Chapter 7

Conclusions and areas for further research

Contents

7.1 Chapter overview

This chapter summarises the research and conclusions for the subject areas covered in each chapter of this thesis, and notes important results. These results are then compared to the stated aim and objectives of this thesis, and shows how each of these

has been addressed. Areas of further work that have been identified within this thesis are then discussed, noting both the challenge and potential benefit of each.

7.2 Applications for ISRIE

The rational behind the need for an instrument for soundscape recognition, identification and evaluation (ISRIE) has been explained, along with the origins and scope of the initial research proposal for the project.

A review of noise metrics currently in use for these applications has been identified, and shortcomings in their fitness for purpose discussed. The ability to separate a soundscape into its constituent parts as a means to better evaluate the soundscape was identified as a key enabler to improving the performance of further processing tasks such as classification and noise metering.

A review of potential applications where ISRIE would be beneficial was then conducted. These identified three broad areas of application:

- Providing more reliable metering for assessing current noise legislation such as BS 4142 and PPG 24.
- Long term noise studies such as noise mapping and health studies.
- Zoological applications, both focused on the effects of anthropogenic noise on animal populations, and on ecological studies monitoring or classifying zoological noise sources.

7.3 Background literature review

An initial review of the literature identified several standard mixing models for general N source, M sensor recording set-ups. Also identified were three standard descriptions of the recording environment. It was realised that ISRIE could potentially be

applied under most combinations of these. The assumption was made that due to the wide variety of environments identified as potential applications for ISRIE, the under-determined model was the most suitable and any candidate methodologies were required to operate under this model.

The goal of separation under ISRIE was then narrowed down to a subcategory of audio separation tasks, namely significance oriented, i.e. preservation of as much of the original signal as possible with the maximum suppression of interfering sources, as the aim is to aid further processing tasks such as classification.

A standard performance metric was identified for signal-to-interference (SIR) to characterise the performance of the separation algorithms. This was deemed the most applicable measure given the stated separation aim. A fundamental limitation on SIR performance was identified for separation methodologies relying on linear transformations of the mixture, which was used to guide selection of an appropriate methodology to research.

Several classes of algorithms were then examined, covering a range of methodologies, assumptions and sensor models. Sparse separation was determined to be a suitable model for ISRIE to exploit, as several works had successfully achieved impressive results for under-determined mixtures using a two sensor model.

It was noted that there is a disparity in the performance metrics published for the performance analysis of different algorithms. This was identified as a failing in previous works that needed addressing in this thesis: performance metrics used need to be stated and ideally standard metrics such as SIR used.

7.4 Methodologies

During the survey of sparse source methodologies, it was noted that most implementations relied on exploited sparseness in the STFT domain. The performance of this transform in terms of the sparseness achieved was identified in the literature as being

dependent on the window size used. As the optimum window size for a mixture is dependent on the time and frequency content of the mixture, *a-priori* knowledge is required to achieved the optimal transform. It was identified that other time-frequency transforms may also be applied to exploit sparseness within a mixture.

The discrete wavelet transform (DWT) was identified as offering the benefit of time-frequency sampling on a dyadic grid, removing the need to select an optimal window length *a-priori*. Shortcomings in applying the DWT were identified. The main hurdle to the application of the DWT is that filtering within the DWT domain leads to subsequent aliasing in the inverse transform.

The dual-tree complex wavelet transform (DTCWT) was identified as offering improvements over the DWT, for the penalty of a doubling in implementation complexity. The DTCWT is a transform used in image processing research due to advantages over the DWT that are applicable here:

- The complex form of the input in the DTCWT means that phase information about the input is accessible.
- The complex form also means that the data is in the same format as the STFT, meaning the DTCWT can be used in existing algorithms with no modification to the algorithm.
- The aliasing effect following the inverse DTCWT caused by filtering in the DTCWT domain is significantly reduced, reducing distortion in the filtered output.

Two methods of implementation for the DTCWT are presented, using lifting or finite impulse response FIR filters. The mathematics of converting between these implementation is also included. The implementation used is specified, along with the filter generation software, to allow for results in this work to be repeatable.

7.5 Evaluation of underlying assumptions

Sparse source separation was identified in chapter 2 as the preferred separation methodology due to good published results for the separation of speech mixtures. As the applications for ISRIE identified in chapter 1 dictate that separation of mixtures containing non-speech noise sources is required, the assumptions underpinning the sparse source methodology were examined using mixtures of representative test cases.

Three typical sound sounds were chosen to represent the three main areas of applications for ISRIE previously discussed. Mixtures were formed within and between these sample groups to broaden the application environments that could be characterised using these subsets of sound groups. The types and number of mixtures used for the testing are given.

Metrics were defined to enable the analysis of the test results, and the test methodology is given in full to allow reproduction of this work.

The sparseness for each mixture has been calculated for each mixture in the time, STFT and DTCWT domains, and full averaged results in each domain for each data set are provided. Tables giving a comparison of all data sets are also provided. The STFT and DTCWT performance are compared with the published results for speech sources, and also against the time domain, which is used as a baseline to determine the performance improvement each transform gives for each data set.

Analysis of the results is provided. The performance of sparse source separation is singled out for ecological applications, where the performance metric is an order of magnitude better than published results for speech mixtures. Performance of mixtures of plant were singled out as performing poorly, with performance an order of magnitude worse than the same results for speech. Consideration is given to the likely causes for this, and to applications where plant sources may be encountered.

7.6 Separation algorithm

This chapter introduces a novel separation methodology that exploits ω-disjoint sparseness in a transform domain. Chapter 4 showed the assumption of ω-disjoint orthogonality is sufficiently well met to achieved good sparsity in most cases.

The methodology proposed uses an off the shelf B-format microphone array to capture audio from a 3D soundscape, extending the separation model to 3D from 2D. This is an improvement over the DUET algorithm used for performance comparisons in chapter 4, which is limited to a 2D half plane.

A method used to find directional information in B-format audio used in the DirAC algorithm (see chapter 5) has been applied to the both the STFT and DTCWT domain transforms of the B-format signals. This method relies on the ω-disjoint sparseness to provide accurate results. The use of the DTCWT with this algorithm is novel, and may lead to performance improvements in the original application if applied throughout the original DirAC algorithm.

This directional information has been used to form a directional binary mask, which is used to filter the source estimates from the mixture. Binary masking was shown in chapter 4 to be an effective method for separating sources.

The mixing model and the mathematics describing the separation have been given. The imposition of an additional assumption has been identified and noted.

Performance metrics for the separation method have been defined, and the relationship between metrics in chapters 4 and 5 discussed to allow direct comparison of performance results.

Performance of the algorithm has been characterised in an anechoic environment, and the reasons for using this environment have been stated. The use of speech mixtures here rather than the test cases is a matter of practicality - recording transport sources within an anechoic chamber poses a logistical challenge.

Finally, the performance of the new separation algorithm using a real microphone array

was compared against the ideal mixing model for a B-format mixture.

The results for all the experiments are discussed, and a comparison between the achieved angular resolution of the proposed algorithm is found to be similar to that of another published work using B-format arrays as a means of localising sources in 2D. Application of the proposed algorithm to this localisation work is also considered, and shown to provide similar angular resolution performance with a significant decrease in computational cost.

7.7 Clustering algorithms

Chapter 5 demonstrated a separation algorithm that was effective given *a-priori* knowledge or an estimate of the location of sources of interest within the soundscape. Chapter 6 detailed work on the development of clustering techniques to form such locational estimates.

Peak estimation using a histogram was identified as a simple yet effective method for determining location estimates for simple soundscapes where only a few sources are of interest, and their locations are fixed. Such soundscapes can be seen in the applications in chapter 1, for example for 24 hour noise recordings of an industrial noise source in legislative applications.

A novel histogram was constructed over the surface of a sphere using a geodesic grid. The benefits of using such a grid over alternative techniques was discussed. The implementation of an arbitrary precision geodesic grid was achieved, and the MATLAB code is included in this work. Also included was C code that performs the assignment of input vectors to the appropriate histogram bin.

An extension to this simple clustering using a rolling window approach to allow for tracking moving sources is also considered.

An alternative clustering approach using a dynamic self learning plastic self organising map (PSOM) of neurons is also considered. Novel work has been implemented in

transforming the algorithm from a Euclidean space to a spherical surface. Further improvements to this clustering model are suggested.

7.8 Summary

This work began with the stated aim of developing a separation algorithm for sources within 3D soundscapes, and has proposed a solution, provided a set of assumptions are met. A series of objectives were proposed to meet this aim, all of which have been addressed in this thesis. To recapitulate these, they were to identify applications where the source separation aspect of ISRIE could have benefit; to review current blind source separation methodologies and develop a methodology suitable for ISRIE; to show that assumptions implicit in this methodology are met in non-speech signals where existing research tends to focus; and finally to investigate direction of arrival information for the separated sources.

This thesis has examined the potential benefit to existing applications for an instrument for soundscape recognition, identification and evaluation. The role of sound source separation and how it supports such an instrument has been explained. Target applications for ISRIE such as noise nuisance monitoring and acoustic ecological studies

A review of applicable separation methodologies was undertaken, and methods exploiting ω-disjoint sparseness such as the DUET (disjoint unmixing and estimation technique) algorithm were identified as a starting point for reasearch. A novel separation algorithm capable of separating sources within a 3-dimensional soundscape was then proposed relying on the assumption of ω-disjoint sparseness. This assumption has been tested against example soundscapes and found to be approximately true in most cases.

The novel application of the DTCWT in achieving this ω-disjoint sparseness has also been proposed, and comparative results between the STFT and the DTCWT have been provided. Few separation methods exploit the wavelet transform, and this thesis

has shown the DTCWT can be used as a direct substitute for the Fourier transform for phase calculations.

Finally this work has given some consideration to the problem of finding the location of sources within the soundscape. Methods for identifying static and moving sources using a novel spherical geodesic grid have been proposed. An approach using a plastic self-organising map for tracking more complex soundscape source movements has been proposed, and extended to make it more suitable for this application. The PSOM approach, whilst unsuccessful, is believed to offer great promise in this application. The potential benefit of the PSOM is greatly reduced computational complexity to achieve high-resolution clustering in 3 dimensions, whilst also coping with temporally intermittent sources. The main hurdle to achieving this is developing a feedback method for the network parameters to control the growth and tracking of the neurons to a particular sources characteristics.

Other than the ω-disjoint assumption, factors in the recording environment such as strongly echoic surfaces and other multi-path propagation channels also affect the suitability of the proposed separation algorithm. This is because the separation relies on the estimated spatial position of the source in each time-frequency bin. The separation performance for heavily reverberant environments, even with sources that are strongly ω-disjoint, will be degraded, as the multi-path channels, if they arrive at the array within the same time window as the direct path, will cause the spatial estimate to be degraded via the peak spreading mechanism discussed in chapter 6.

Spaced echoic surfaces, such as street canyons, pose less of a problem to the separation algorithm, as the multi-path channels are more likely to be significantly delayed and arrive in a different time window. For frequency-varying sources, this will result in the multi-path artefacts being included in the estimated source. Therefore, although the echoic artefacts will be present in the separated sources, the sources will be correctly separated.

Echoic environments will, in general, still degrade the separation performance, as the echoic artefacts act to make ω-disjoint signals less ω-disjoint, as the time-frequency

components of the signal will be replicated in time at delays corresponding to the propagation paths. In soundscapes where there may be many sources simultaneously active, this may make the difference between success and failure of the separation algorithm.

In light of these considerations, soundscapes where the proposed algorithm can be expected to perform well are in suburban or rural settings, where the number of simultaneously active sources can expected to be lower than in urban centres, and where strongly echoic surfaces are less likely to be encountered.

7.9 Areas for further research

7.9.1 Field overview

Separation of sources from soundscapes is a vast and hard problem. This thesis has proposed a method suitable for separating acoustic sources in soundscapes, particularly rural and wilderness soundscapes, where the ω-disjoint assumption is more likely to be met.

The long term goal of an instrument that can separate all distinct sounds from within a general soundscape, under all environmental conditions is, in the author's view, at least a decade away. It is likely that such an instrument will require the combination of several separation approaches, possibly combined with array beamforming methods to improve the SIR. As mentioned in the next section, combining such an instrument with feedback of a performance measure from a post-processor to adapt its separation parameters may also yield improvements. For a battery-powered portable instrument to be able to separate, track and identify multiple moving sources within a 3D soundscape, modest increases must also be attained in processing power, as such a task is beyond the standards of today's embedded processors using the algorithms discussed in this work.

7.9.2 Areas of further research identified by this work

Several areas have been identified where further research could bring advances to the state of the art. These are discussed below for each area.

Time-frequency transforms

The DTCWT has been shown to be applicable in place of the STFT. Several avenues for further research are possible on this theme.

Comparison of performance between different choices for the wavelet basis used in the DTCWT. The choice of wavelet basis is not investigated in this thesis, and is affects the time-frequency representation achieved for a given source. Other bases may lead to more sparse representation than that achieved in this thesis.

This thesis has shown that the DTCWT can be successfully applied to existing techniques, specifically DUET and DirAC. However, many separation algorithms exist that exploit time-frequency transformations, and may benefit from the application of the DTCWT in place of the STFT.

Alternatively, there are other time-frequency transforms that have not been considered in this work that may be applied to the separation algorithm developed in this work.

Testing the assumption of source sparsity within soundscapes

Research into this assumption for further specific soundscape types would compliment the work contained in this thesis.

The results in this thesis for the sparsity measure are given for a range of threshold values for an ideal binary mask. Identifying the threshold for which the performance of the separation algorithm proposed in this work and the ideal model are matched would allow the the ideal model to be used as a predictor for the performance of the separation algorithm. This would be particularly beneficial given the difficulty of

forming quantitative metrics for the performance of the separation algorithm in realistic soundscape applications.

Separation algorithm

Further testing in controlled environments such as anechoic or echoic chambers is recommended for sound sources more representative of target applications. This is hampered by the logistics of recording in such an environment.

Alternatively, further research into metrics that allow for definitive performance measures to be achieved for sources recorded outside such an environment would allow characterisation of the separation algorithm across a wider selection of soundscapes.

Source localisation

The PSOM method for source clustering introduced in chapter 6 has great potential for estimating source localisation. Optimising the algorithm to operate for all soundscapes under a range of noise conditions is a large task that may be suitable for an extended period of research. In particular, developing a feedback mechanism from the clustering performance to provide effective control of the network parameters is a complex control engineering problem. If such a mechanism can be found, then source estimation and tracking for unknown, varying numbers of non-stationary sources, with unknown initial locations could be achieved. The benefit of this is not limited to the work in this thesis, but would be applicable across a wide variety of research fields.

Integration

This thesis is concerned with the development of a system of sound source separation with the ultimate aim of improving the performance of an instrument for soundscape identification, recognition and evaluation.

Research into the integration of the proposed system from this thesis with post-

processing tasks such as classification is required. This research should aim to quantify any performance enhancements gained using separation as a pre-processing stage.

A post-processing stage for the classification of noise may be useful for enhancing performance of the separation algorithm, for example by varying the angular threshold used in the filtering stage. Such research will be needed before a practical ISRIE capable of unattended deployment is possible.

Appendix A

MATLAB Code - Wavelet utilities

Contents

A.1 DTCWT first stage coefficients

$\widetilde{h}(z^{-1})$	$h(z)$	$\widetilde{g}(z^{-1})$	$g(z)$
0	0	0	0
-0.0884	-0.0112	0.0112	-0.0884
0.0884	0.0112	0.0112	-0.0884
0.6959	0.0884	-0.0884	0.6959
0.6959	0.0884	0.0884	-0.6959
0.0884	-0.6959	0.6959	0.0884
-0.0884	0.6959	0.6959	0.0884
0.0112	-0.0884	0.0884	0.0112
0.0112	-0.0884	-0.0884	-0.0112
0	0	0	0

A.2 FIR filter coefficients to polyphase coefficients conversion

```
function [he,ho,ge,go,power] = FIR2polyphase(h,g,p)
% Takes filters h and g with h(1)z^p and g(1)z^p, and returns the polyphase
% equivilent, along with the max power in the polyphase components
%
% calculate max power of polyphase components
power = ceil(p/2);
he=[];ho=[];ge=[];go=[];

%if odd power, set ho and go and then remove highest power term in h and g
if rem(p,2)~= 0
    he(1)=0;    ho(1)=h(1);
    ge(1)=0;    go(1)=g(1);
    h(1)=[];    g(1)=[];
end

% for h filter
for n=1:length(h)
    if rem(n,2) ~= 0        % if even power
        he(end+1,1) = h(n);
    else                    % if odd power
        ho(end+1,1) = h(n);
    end
end

% for g filter
for n=1:length(g)
    if rem(n,2) ~= 0        % if even power
        ge(end+1,1) = g(n);
    else                    % if odd power
        go(end+1,1) = g(n);
    end
end

if rem(p,2)~= 0
    ho(end+1)=0;
    go(end+1)=0;
end
```

A.3 Factorise polyphase coefficients into lifting stages

```
function [Ps,Pa] = polyphase2Lift(he,ho,ge,go,p)
% Performs factorisation of synthasis filters into lifting steps
% Inputs must be equal length filters.

if (nnz(he) ~= nnz(ho)) || (nnz(ge) ~= nnz(go)) || (nnz(he) ~= nnz(ge))
    display('ge and ho must be of equal length');
end

%first step
if(rem(nnz(he),2)~=0)                       % if odd length
    [si,ge] = poly_longdiv_2(ge,go,p,'l');
    [temp] = poly_times(si,ho,p);
    he=he-temp;
    he(end-1)=0;
else                                        % if even length
    [si,ge] = poly_longdiv_2(ge,go,p,'f');
    [temp] = poly_times(si,ho,p);
    he=he-temp;
    he(1)=0;
end
    Ps(:,1)=si(p:p+2);

    [ti,ho] = poly_longdiv_2(ho,he,p,'f');
    [temp] = poly_times(ti,ge,p)
    cancel(1) = find(go,1,'first');
    cancel(2) = find(go,1,'last');
    cancel(cancel==p+1) =[];
    go=go-temp;
    go(cancel)=0
    Ps(:,2)=ti(p:p+2);

    i=2;
    while(nnz(ho) > 0 || nnz(ge) > 0)
        [si,ge] = poly_longdiv_2(ge,go,p,'f');
        [temp] = poly_times(si,ho,p);
        cancel(1) = find(he,1,'first');
        cancel(2) = find(he,1,'last');
        cancel(cancel==p+1) =[];
        he=he-temp;
        he(cancel)=0;
        Ps(:,2*i-1)=si(p:p+2);

        if nnz(ho) > 0
            [ti,ho] = poly_longdiv_2(ho,he,p,'f');
            [temp] = poly_times(ti,ge,p);
            cancel(1) = find(go,1,'first');
            cancel(2) = find(go,1,'last')
```

```
        cancel(cancel==p+1) =[]
        go=go-temp
        go(cancel)=0;
        Ps(:,2*i)=ti(p:p+2);
    end
    i=i+1;
end

%move scaling factor into lifting steps
k= he(p+1);
if rem(length(Ps),2)~=0 %odd length
    % last term in Ps is si, use
    % [k 0; 0 1/k] = [1 k-k^2;0 1][1 0;-1/k 1][1 k-1;0 1][1 0;1 1];
    Ps(2,size(Ps,2))=Ps(2,size(Ps,2)) + k - k^2;
    Ps(:,size(Ps,2)+1)=[0;-1/k;0];
    Ps(:,size(Ps,2)+1)=[0;k-1;0];
    Ps(:,size(Ps,2)+1)=[0;1;0];
else
    %last term in Ps is ti
    % [k 0; 0 1/k] = [1 0;-1 1][1 1-1/k;0 1][1 k-1;0 1][1 0;1 1];
    Ps(2,size(Ps,2))=Ps(2,size(Ps,2)) -1;
    Ps(:,size(Ps,2)+1)=[0;1-1/k;0];
    Ps(:,size(Ps,2)+1)=[0;k;0];
    Ps(:,size(Ps,2)+1)=[0;1/k^2 - 1/k;0];
end
Pa = -Ps(:,end:-1:1);
```

A.4 Lifting transform

```
function varargout = liftingTransform(x,Pa)
% [lp,hp] = liftingTransform(x,Pa)
% [y] = liftingTransform(x,Pa)
%performs the a wavelet lifting transform on x.
%Assumes scaling factor is combined as a series of lifting stages
x = x(:);

% make signal even lengthed
cut = 0;
if rem(length(x),2)~=0
    x=[x;0];
    cut = 1;
end

%pad 2 zeros for each si and ti pair applied.
pad_length = 2 * ceil(length(Pa)/2); % ensures even length zero padding
x = [zeros(pad_length,1);x;zeros(pad_length,1)];

%apply each lifting step in Pa
%first stage (Pa(end) is a lifting (si) step
L = size(Pa,2);
for i = 0:L-1
    if rem(i,2) == 0 %even, therefore lifting (si) step
        % lifting step, update odd using even
        x(3:2:end-3) = x(3:2:end-3) + Pa(1,L-i) .* x(6:2:end); % z^{1} term
        x(3:2:end-3) = x(3:2:end-3) + Pa(2,L-i) .* x(4:2:end-2); % z^{0} term
        x(3:2:end-3) = x(3:2:end-3) + Pa(3,L-i) .* x(2:2:end-4); % z^{-1} term
    else %odd, therefore dual (ti) step
        x(4:2:end-2) = x(4:2:end-2) + Pa(1,L-i) .* x(5:2:end-1); % z^{1} term
        x(4:2:end-2) = x(4:2:end-2) + Pa(2,L-i) .* x(3:2:end-3); % z^{0} term
        x(4:2:end-2) = x(4:2:end-2) + Pa(3,L-i) .* x(1:2:end-5); % z^{-1} term
    end
end

%remove padding.
x(1:pad_length)=[];
x(end-pad_length+1:end)=[];
if cut ==1
    x(end)=[];
end

if nargout == 2
    varargout{1} = x(1:2:end); %lp
    varargout{2} = x(2:2:end); %hp
else
    varargout{1} = x;
end
```

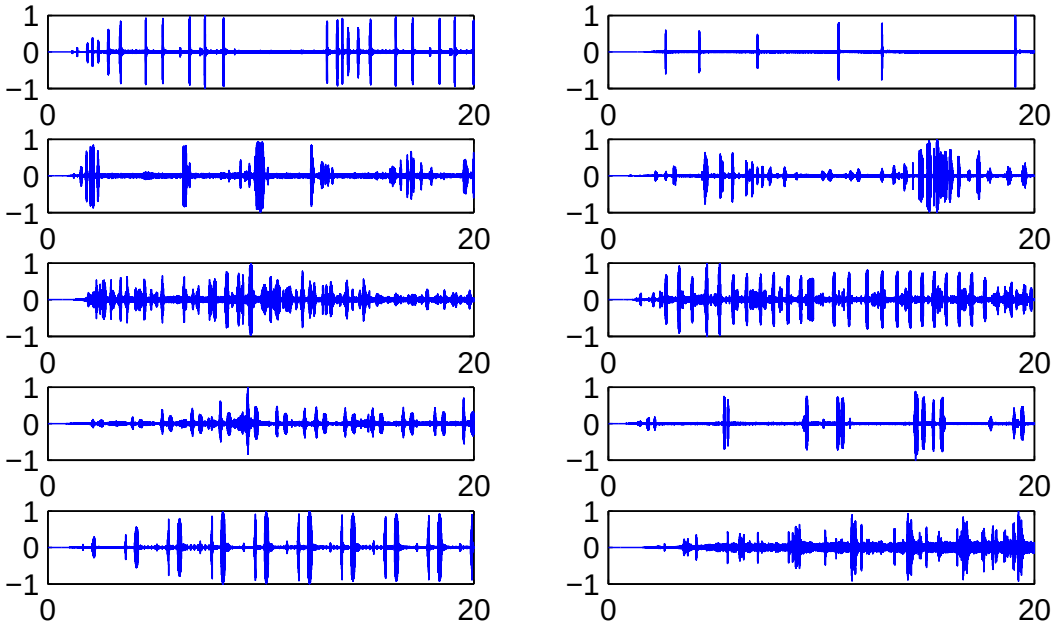
A.5 Inverse lifting transform

```
function x = inverseLiftingTransform(x,Ps)
% y = inverseLiftingTransform(x,Ps)
%performs the inverse wavelet lifting transform on x.
%Assumes scaling factor is combined as a series of lifting stages
%assumes lifting steps are in range z -> z^-1
% Takes Ps in the form that
x = x(:);

% make signal even lengthed
cut = 0;
if rem(length(x),2)~=0
    x=[x;0];
    cut = 1;
end

%zero pad
% pad 2 zeros for each si adn ti pair applied.
pad_length = 2 * ceil(length(Ps)/2); % ensures even length zero padding
x = [zeros(pad_length,1);x;zeros(pad_length,1)];

%apply each lifting step in Ps
%first stage (Ps(end) is a lifting (ti) step
L = size(Ps,2);
for i = 0:L-1
    if rem(i,2) ~= 0 %odd, therefore lifting (si) step
        % lifting step, update odd using even
        x(3:2:end-3) = x(3:2:end-3) + Ps(1,L-i) .* x(6:2:end); % z^{1} term
        x(3:2:end-3) = x(3:2:end-3) + Ps(2,L-i) .* x(4:2:end-2); % z^{0} term
        x(3:2:end-3) = x(3:2:end-3) + Ps(3,L-i) .* x(2:2:end-4); % z^{-1} term
    else %even, therefore dual (ti) step
        x(4:2:end-2) = x(4:2:end-2) + Ps(1,L-i) .* x(5:2:end-1); % z^{1} term
        x(4:2:end-2) = x(4:2:end-2) + Ps(2,L-i) .* x(3:2:end-3); % z^{0} term
        x(4:2:end-2) = x(4:2:end-2) + Ps(3,L-i) .* x(1:2:end-5); % z^{-1} term
    end
end

%remove padding.
x(1:pad_length)=[];
x(end-pad_length+1:end)=[];
if cut ==1
    x(end)=[];
end
```

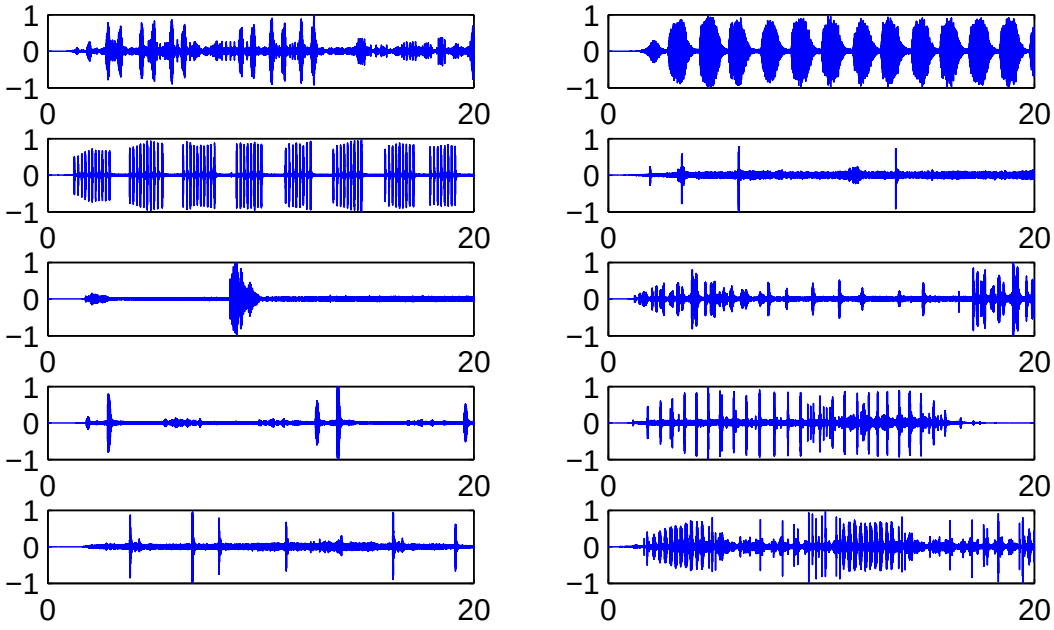
Appendix B

Validation of ω-disjoint orthogonaity assumption

Contents



(a) Sources 1 to 5 displayed in descending order in the left column, 6 to 10 on the right



(b) Sources 11 to 15 displayed in descending order in the left column, 16 to 20 on the right

Figure B.1: Unity scaled time domain bird song sources from recordings [1]

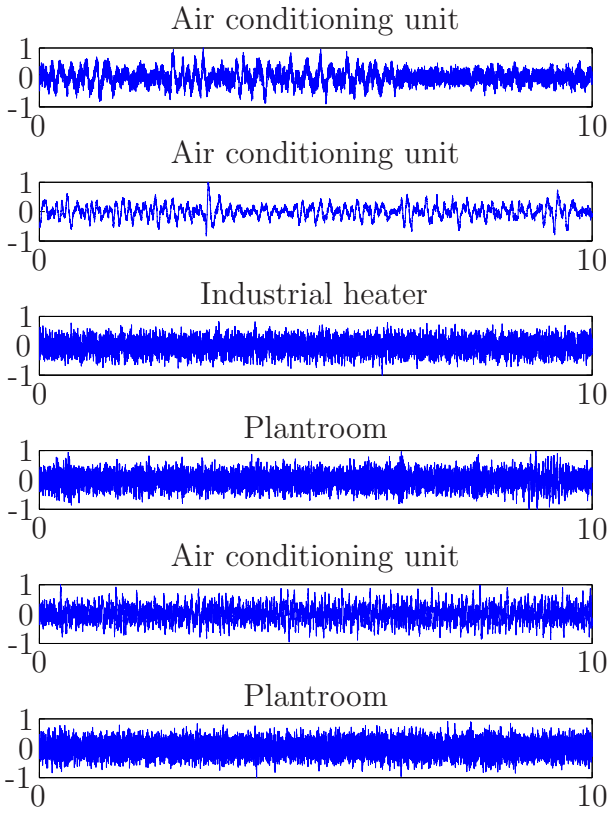


Figure B.2: Unity scaled time domain plant sources

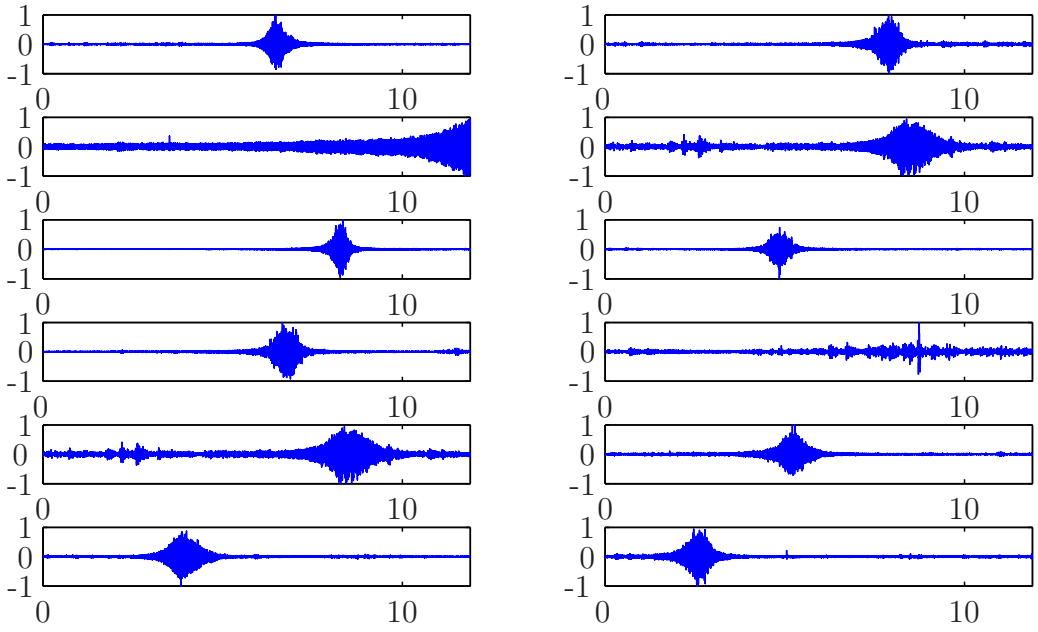
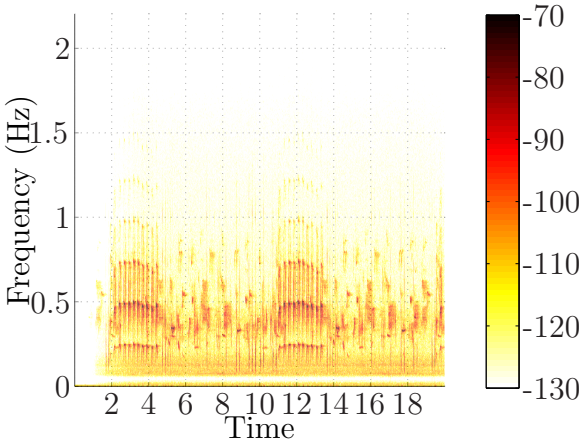
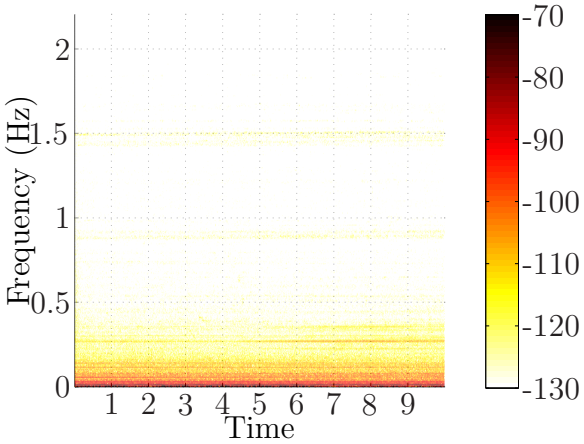


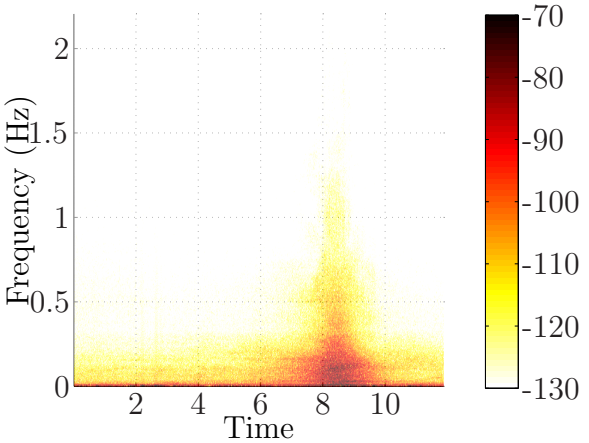
Figure B.3: Unity scaled time domain transport sources



(a) Spectrogram of a typical bird source. 1024 sample Hamming window used



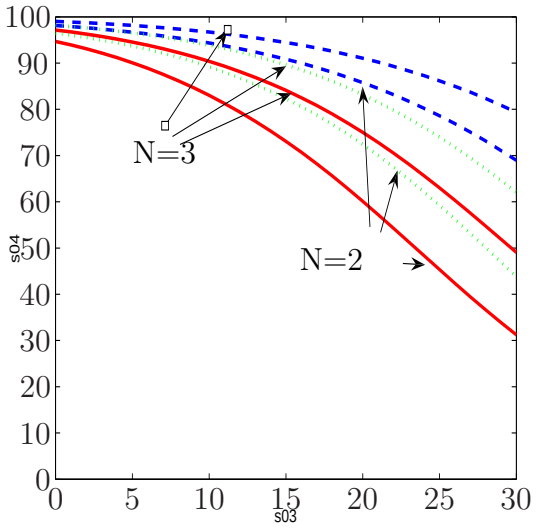
(b) Spectrogram of a typical plant source. 1024 sample Hamming window used



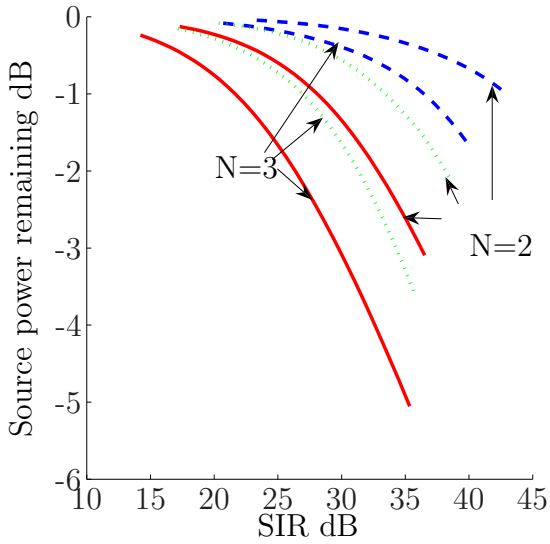
(c) Spectrogram of a typical transport source. 1024 sample Hamming window used

Figure B.4: Spectrograms for typical bird, plant and transport sources

$\times 10^4$

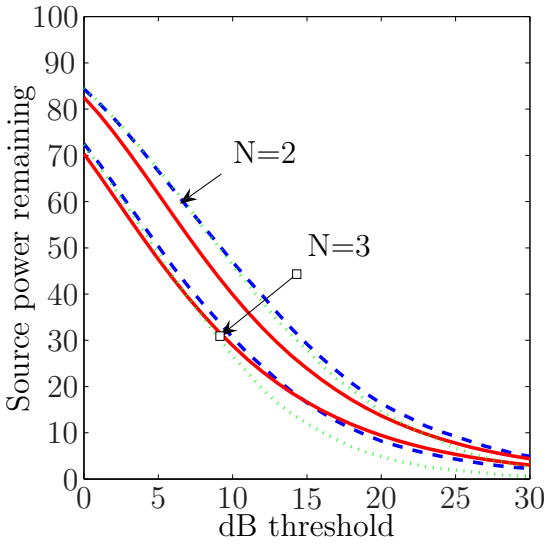


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$ and for $N = 2, 3$

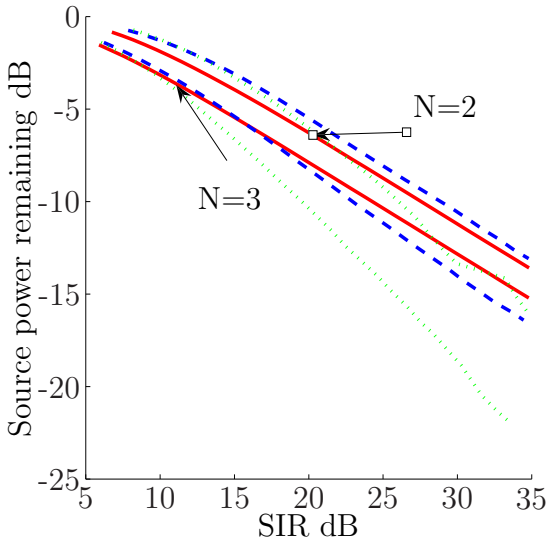


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, and $N = 2, 3$.

Figure B.5: Results for separation of mixtures of bird song recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

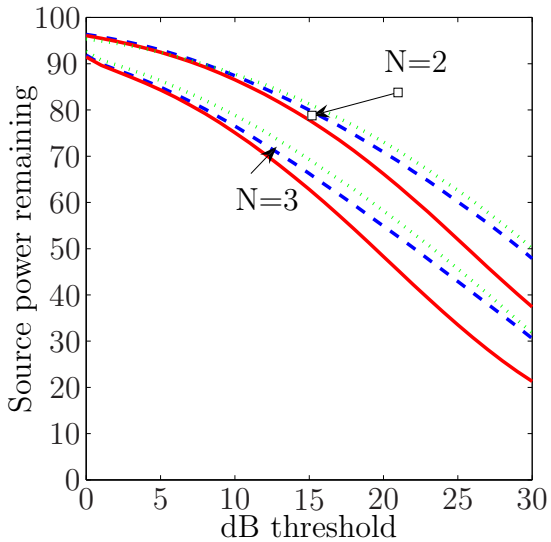


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$ and for $N = 2, 3$

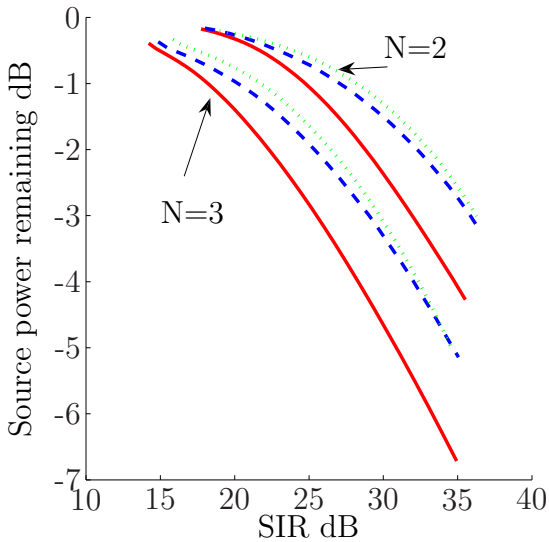


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, and $N = 2, 3$.

Figure B.6: Results for separation of mixtures of plant recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

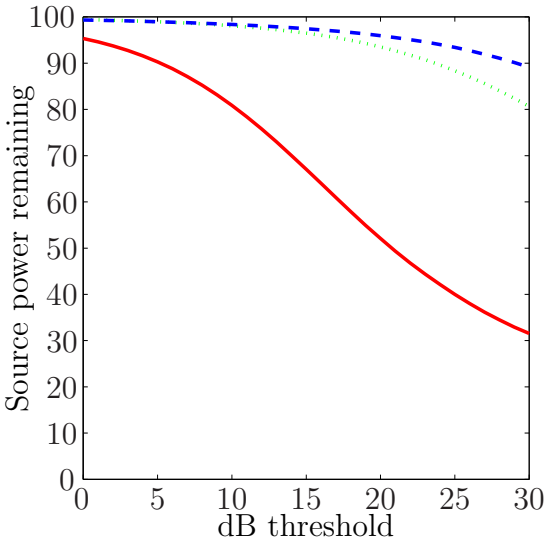


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$ and for $N = 2, 3$

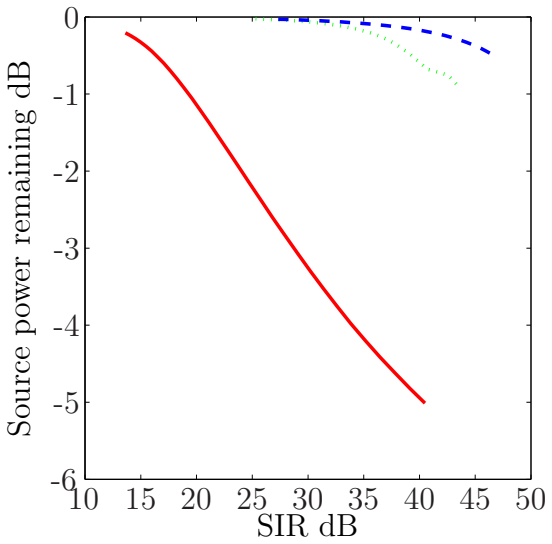


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, and $N = 2, 3$.

Figure B.7: Results for separation of mixtures of transport recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

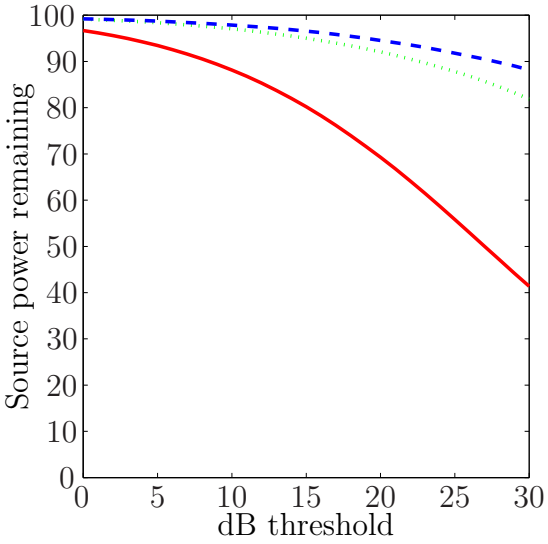


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$

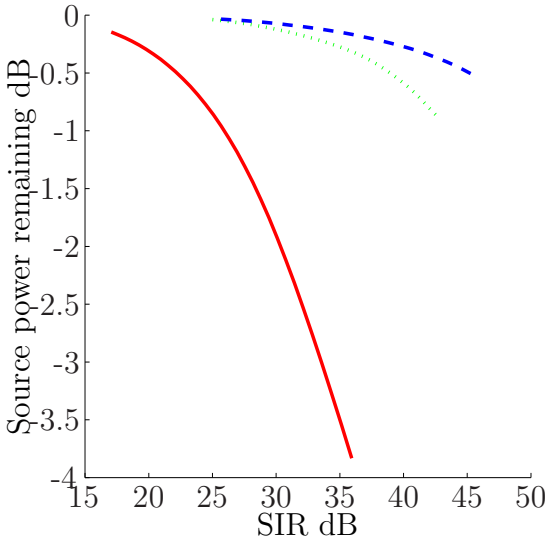


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.8: Results for the separation of bird song recordings from mixtures of bird song and plant recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

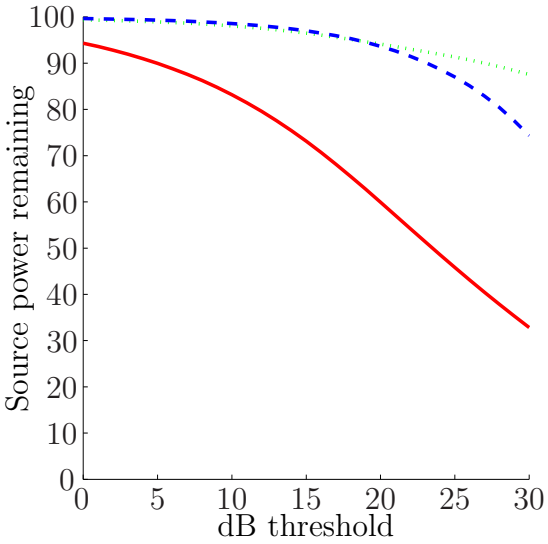


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$

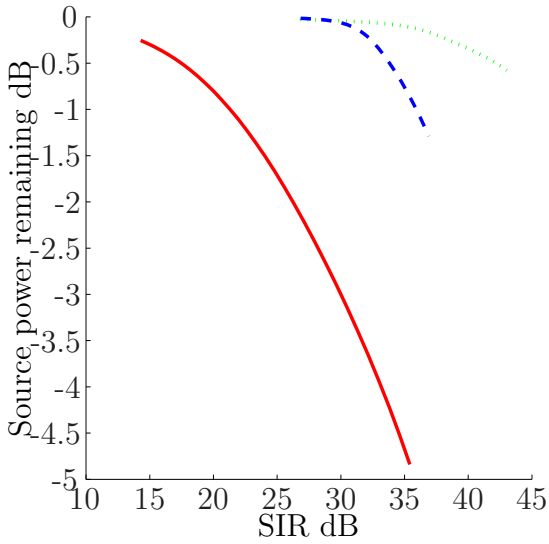


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.9: Results for the separation of bird song recordings from mixtures of bird song and transport recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

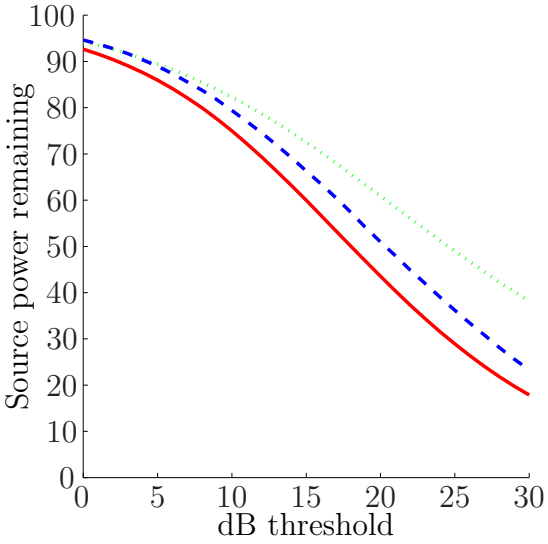


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$

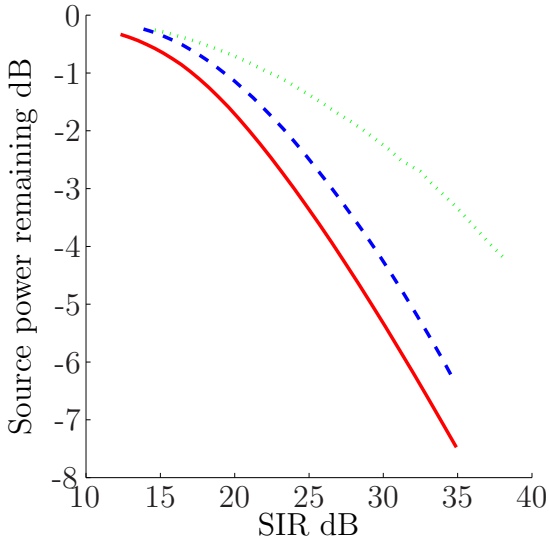


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.10: Results for the separation of plant recordings from mixtures of plant and bird song recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

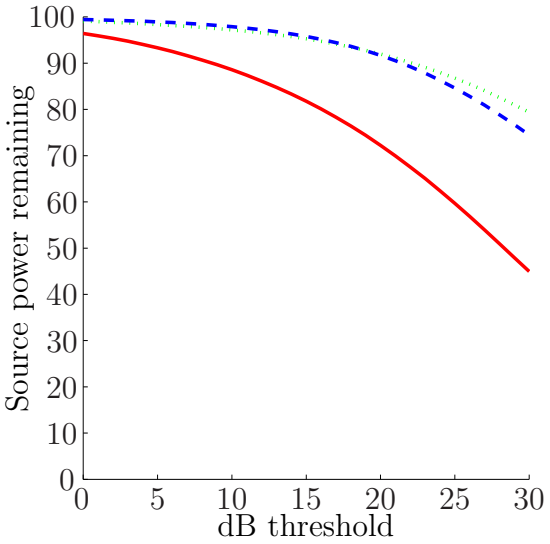


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$

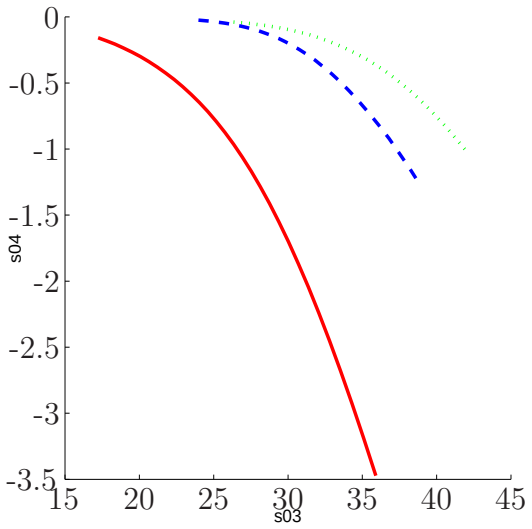


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.11: Results for the separation of plant recordings from mixtures of plant and transport recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

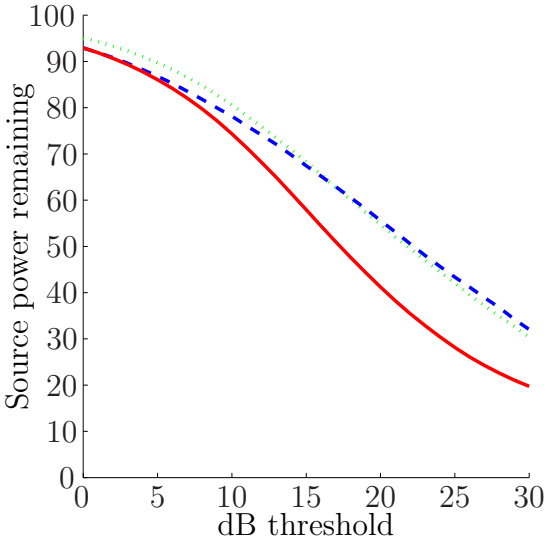


(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$

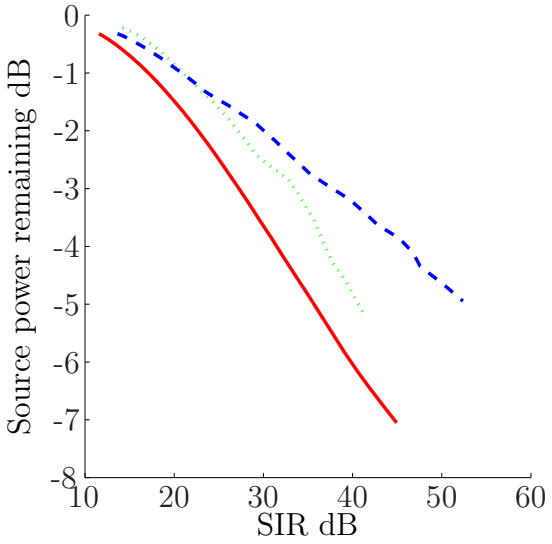


(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.12: Results for the separation of transport recordings from mixtures of transport and bird song recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain



(a) Approximate W-disjoint orthogonality. Plot of mean $r(x)$ for $x = 0, 1, \ldots, 30$, $N = 2$



(b) Ideal separation with binary mask of x dB. Plot of $r(x)$ dB against SIR dB for $x = 0, 1, \ldots, 30$, $N = 2$.

Figure B.13: Results for the separation of transport recordings from mixtures of transport and plant recordings. Green dotted lines are the mean values in the DTCWT domain, blue dashed lines are the mean values in the STFT domain, with a window size of 1024 samples with 50% overlap. Red lines are the mean values for the time domain

B.1 Appendix Bibliography

[1] K. Tsuruhiko and M. Michio. *The songs and calls of 333 birds in Japan: Non-songbirds.*, volume 1. Shogakukan Inc. ,Tokyo, 1996. ISBN 4-09-480072-7.

S	I	Domain	Threshold a dB						
			0	3	5	10	15	20	30
Sp	Sp	Time	11.50	13.65	15.16	20.25	23.95	28.95	39.14
		STFT	13.82	15.76	17.42	22.51	26.28	31.00	39.94
		DTCWT	11.23	13.89	15.85	22.04	26.29	31.21	40.30
B	B	Time	17.29	19.09	20.31	23.33	26.36	29.52	36.53
		STFT	23.34	25.19	26.47	29.68	32.92	36.28	43.00
		DTCWT	20.36	22.11	23.32	26.28	29.08	32.05	38.41
B	2B	Time	14.20	16.01	17.25	20.39	23.63	27.14	35.35
		STFT	20.71	22.60	23.87	27.10	30.44	33.66	39.77
		DTCWT	17.15	18.87	19.99	22.88	25.94	28.98	35.72
B	P	Time	13.62	14.99	15.97	18.89	22.82	27.92	40.49
		STFT	27.34	29.15	30.42	33.71	37.03	40.31	46.69
		DTCWT	25.29	27.02	28.29	31.39	34.28	37.06	43.69
B	T	Time	17.06	18.69	19.80	22.67	25.69	28.79	35.94
		STFT	25.67	27.39	28.54	31.71	35.26	39.07	45.90
		DTCWT	25.00	26.43	27.46	30.36	33.56	36.84	42.59
P	P	Time	6.76	8.95	10.58	15.10	19.87	24.82	34.82
		STFT	7.85	9.99	11.53	15.45	19.73	24.60	34.80
		DTCWT	8.20	10.12	11.49	15.11	19.21	23.76	34.74
P	2P	Time	5.89	8.39	10.19	14.90	19.82	24.80	34.80
		STFT	6.21	8.53	10.16	14.53	19.25	24.47	34.46
		DTCWT	6.02	8.15	9.68	13.74	18.45	23.38	33.48
P	B	Time	14.28	15.94	17.07	19.98	23.32	27.14	35.42
		STFT	26.85	27.91	28.56	29.99	31.28	32.62	36.90
		DTCWT	26.78	29.24	30.75	34.27	36.69	38.71	43.02
P	T	Time	12.32	14.06	15.21	18.26	21.69	25.68	34.91
		STFT	13.87	15.35	16.45	19.31	22.61	26.33	34.77
		DTCWT	14.62	16.42	17.72	21.18	25.11	29.51	38.05
T	T	Time	17.73	19.21	20.18	22.63	25.19	28.06	35.52
		STFT	18.00	19.67	20.77	23.78	26.86	29.95	36.42
		DTCWT	18.98	20.64	21.80	24.71	27.52	30.31	36.28
T	2T	Time	14.21	15.71	16.72	19.36	22.38	25.98	34.93
		STFT	14.85	16.72	17.97	21.29	24.47	27.68	35.06
		DTCWT	15.85	17.70	18.94	21.95	24.81	27.74	34.53
T	B	Time	17.22	18.93	20.07	22.90	25.75	28.75	35.91
		STFT	23.97	25.00	25.73	27.64	29.81	32.37	38.94
		DTCWT	26.33	27.97	28.91	31.04	33.33	35.91	41.90
T	P	Time	11.56	13.28	14.63	18.87	24.44	30.88	44.93
		STFT	13.64	15.74	17.28	21.49	27.55	33.90	52.37
		DTCWT	14.23	15.99	17.16	20.59	25.27	30.84	41.30

Table B.1: Overview of SIR results for all test cases. Key: S=Sources, I=Interfering source(s), Sp=speech, B=Bird song, P=Plant, T=Transport

S	I	Domain	Threshold a dB						
			0	3	5	10	15	20	30
Sp	Sp	Time	-0.30	-0.48	-0.63	-1.26	-1.79	-2.55	-3.99
		STFT	-0.20	-0.29	-0.41	-0.82	-1.16	-1.58	-2.47
		DTCWT	-0.32	-0.55	-0.73	-1.31	-1.71	-2.14	-3.03
B	B	Time	-0.13	-0.19	-0.24	-0.44	-0.75	-1.25	-3.09
		STFT	-0.04	-0.06	-0.08	-0.15	-0.25	-0.41	-1.00
		DTCWT	-0.08	-0.12	-0.15	-0.28	-0.48	-0.80	-2.07
B	2B	Time	-0.24	-0.35	-0.45	-0.81	-1.36	-2.21	-5.05
		STFT	-0.08	-0.12	-0.15	-0.25	-0.42	-0.67	-1.61
		DTCWT	-0.16	-0.22	-0.28	-0.50	-0.85	-1.40	-3.59
B	P	Time	-0.21	-0.33	-0.44	-0.92	-1.74	-2.83	-5.01
		STFT	-0.03	-0.04	-0.05	-0.07	-0.11	-0.18	-0.50
		DTCWT	-0.02	-0.04	-0.04	-0.08	-0.15	-0.29	-0.93
B	T	Time	-0.15	-0.22	-0.29	-0.55	-0.96	-1.59	-3.83
		STFT	-0.03	-0.05	-0.06	-0.09	-0.15	-0.24	-0.55
		DTCWT	-0.04	-0.06	-0.07	-0.13	-0.22	-0.36	-0.87
P	P	Time	-0.83	-1.51	-2.10	-4.00	-6.22	-8.64	-13.58
		STFT	-0.74	-1.28	-1.77	-3.29	-5.35	-7.87	-13.08
		DTCWT	-0.73	-1.26	-1.74	-3.35	-5.54	-8.39	-15.99
P	2P	Time	-1.53	-2.47	-3.23	-5.40	-7.79	-10.26	-15.21
		STFT	-1.40	-2.27	-2.98	-5.14	-7.83	-10.85	-16.41
		DTCWT	-1.41	-2.33	-3.13	-5.75	-9.20	-13.09	-21.90
P	B	Time	-0.25	-0.36	-0.46	-0.80	-1.36	-2.22	-4.84
		STFT	-0.02	-0.02	-0.03	-0.06	-0.13	-0.29	-1.29
		DTCWT	-0.03	-0.04	-0.05	-0.08	-0.16	-0.26	-0.58
P	T	Time	-0.33	-0.50	-0.66	-1.25	-2.22	-3.61	-7.48
		STFT	-0.24	-0.38	-0.51	-1.00	-1.78	-2.93	-6.33
		DTCWT	-0.26	-0.38	-0.48	-0.85	-1.40	-2.15	-4.18
T	T	Time	-0.17	-0.26	-0.34	-0.63	-1.09	-1.79	-4.27
		STFT	-0.16	-0.24	-0.32	-0.58	-0.97	-1.49	-3.20
		DTCWT	-0.21	-0.28	-0.34	-0.57	-0.90	-1.36	-3.01
T	2T	Time	-0.39	-0.60	-0.74	-1.24	-2.02	-3.16	-6.71
		STFT	-0.37	-0.58	-0.71	-1.16	-1.79	-2.61	-5.14
		DTCWT	-0.33	-0.52	-0.64	-1.03	-1.59	-2.35	-4.98
T	B	Time	-0.15	-0.23	-0.3	-0.53	-0.87	-1.41	-3.47
		STFT	-0.02	-0.04	-0.05	-0.09	-0.19	-0.38	-1.28
		DTCWT	-0.04	-0.06	-0.07	-0.12	-0.21	-0.36	-1.00
T	P	Time	-0.32	-0.49	-0.65	-1.29	-2.37	-3.85	-7.05
		STFT	-0.32	-0.48	-0.62	-1.07	-1.71	-2.55	-4.95
		DTCWT	-0.22	-0.35	-0.47	-0.93	-1.66	-2.62	-5.18

Table B.2: Overview of ω-disjoint measure $r_(a)$ results for all test cases. Key: S=Sources, I=Interfering source(s), Sp=speech, B=Bird song, P=Plant, T=Transport

Appendix C

Code - Clustering

Contents

C.1 Spherical geodesic grid generation

```
function [tri,x,y,z] = geodesic(n)
%[tri,x,y,z] = geodesic(n)
r = (1 + sqrt(5))/2;

x = [0;0;0;0; 1;1;-1;-1; r;r;-r;-r];
y = [1;1;-1;-1; r;-r;r;-r; 0;0;0;0];
z = [r;-r;r;-r; 0;0;0;0; 1;-1;1;-1];
       % [A B C; (Top, Left, Right)]
       %            A
       %          B   C
iso = uint64([3 9 1 ; 3 6 9  ; 3 8 6  ; 3 11 8; 3 1 11; ... % top pentagon
              2 10 5; 2 4 10 ; 2 12 4 ; 2 7 12; 2 5 7 ;...% bottom pentagon
              5 9 1 ; 9 10 5 ; 10 6 9 ; 6 4 10; 4 8 6 ;
              8 12 4; 12 11 8; 11 7 12; 7 1 11; 1 5 7 ]);

tri = uint64([]);
% have to do the hard stuff here!

if n >1
    no_new = sum([2:n]);
```

```
% for top pentagon
for t = 1 :10

    A = iso(t,1);
    B = iso(t,2);
    C = iso(t,3);

    %calculate Vector AC
    ACx = (x(A) - x(C))/n;
    ACy = (y(A) - y(C))/n;
    ACz = (z(A) - z(C))/n;
    %calculate Vector AB
    ABx = (x(A) - x(B))/n;
    ABy = (y(A) - y(B))/n;
    ABz = (z(A) - z(B))/n;

    % create points in triangle
    comp =0;
    point_start = 12 +((t-1) *no_new);
    for e = 1 : n-1

        dx = ((0:(n-e)) * -ACx) - (e*ABx);
        dy = ((0:(n-e)) * -ACy) - (e*ABy);
        dz = ((0:(n-e)) * -ACz) - (e*ABz);

        x((point_start + comp) + (1:(n+1-e))) = x(A) + dx;
        y((point_start + comp) + (1:(n+1-e))) = y(A) + dy;
        z((point_start + comp) + (1:(n+1-e))) = z(A) + dz;
        comp = (n+1-e)+comp;
    end

    tri_start = ((t-1)*n^2);

    % create mesh for triangle
    m=1;
    p=1;
    mesha = zeros(sum(1:n-1),3);

    for f = 1 : n-1
        for d = 1 : n-f
            mesha(m,:) = [(point_start + p),
                          (point_start + p - (n-f+1)),
                          (point_start + p +1)];
            if (t ==5 || t==10) && d == 1
                tri(tri_start - 4*(n^2) + f,2) = (point_start + p);
            end
            p = p + 1;
            m=m+1;
        end
        p=p+1;
```

```
    end
    %fix the first diagonal of each
    fix(1) = 1;
    for l = 2:n-1
        fix(l) = fix(l-1)+n-(l-2);
    end

    mesha(1:n-1,2)= point_start - no_new + fix';
    meshb= zeros(sum(1:n),3);
    meshb(1,:)=[A,point_start+1,point_start-no_new + fix(1)];

    for  l = 1:n-2
        meshb(l+1,:)= [point_start - no_new + fix(l),
                       point_start + l + 1,
                       point_start - no_new + fix(l+1)];
    end

    meshb(n,:) = [point_start-no_new+ fix(n-1),point_start+n,C];
    m=n+1;
    p=1;

    for f = 1 : n-1
        for d = 1 : n-f
            meshb(m,:) = [(point_start + p),
                          (point_start + p + (n-f+1)),
                          (point_start + p +1)];
                %fix the join on the first triangle
            if (t==5 || t==10)  && d==1
                mesha_size = size(mesha,1);
                tri(tri_start - 4*(n^2) + mesha_size + f,3) =
                                                (point_start + p);
                tri(tri_start - 4*(n^2) + mesha_size + 1 + f,1) =
                                                (point_start + p);
            end
            p = p + 1;
            m=m+1;
        end
        p=p+1;
    end
    meshb(end,2)=B;

    tri(tri_start+(1:n^2),1:3) = [mesha;meshb];
end

for t= 11:20
   % interconnections
    A = iso(t,1);
    B = iso(t,2);
    C = iso(t,3);
```

```
    %calculate Vector AC
    ACx = (x(A) - x(C))/n;
    ACy = (y(A) - y(C))/n;
    ACz = (z(A) - z(C))/n;
    %calculate Vector AB
    ABx = (x(A) - x(B))/n;
    ABy = (y(A) - y(B))/n;
    ABz = (z(A) - z(B))/n;

    % create points in triangle
    comp =0;
    no_new2 = sum(1:n-1);

    point_start = 12 +(10 *no_new + (t-11)*no_new2);

    for e = 1 : n-1

        dx = ((0:(n-e-1)) * -ACx) - (e*ABx);
        dy = ((0:(n-e-1)) * -ACy) - (e*ABy);
        dz = ((0:(n-e-1)) * -ACz) - (e*ABz);

        x((point_start + comp) + (1:(n-e))) = x(A) + dx;
        y((point_start + comp) + (1:(n-e))) = y(A) + dy;
        z((point_start + comp) + (1:(n-e))) = z(A) + dz;
        comp = (n-e)+comp;
    end
end
for t= 11:2:19
     % create mesh for triangle
   % interconnections
    A = iso(t,1);
    B = iso(t,2);
    C = iso(t,3);

     m=1;

     mesha = zeros(sum(1:n-1),3);
     point_start = 12 +(10 *no_new + (t-11)*no_new2);
     p=n;
     % to fix the edges...
     fix = 1;
     for f = 2:n-1
     fix(f)=fix(f-1)+n-f+1;
     end
     for f = 1 : n-1
         for d = 1 : n-f
             mesha(m,:) = [(point_start + m),
                           (point_start + m - (n-f)),
                           (point_start + m +1)];
             m=m+1;
```

```
     end
     mesha(m-1,3) = (12 + ((((t+1)/2)-6)*no_new) + p);
     mesha(f,2)=fix(n-f) + 12 + 10*no_new + (t-12)*no_new2;
     p = p + n-f;
 end
if t==11
   mesha(1:n-1,2)=(fix(n-1:-1:1)' + 12 + 10*no_new + 9*no_new2);
end

tri= [tri;mesha];
meshb= zeros(sum(1:n),3);
meshb(1,:)=[A,point_start+1,point_start-no_new2+ fix(n-1)];
for  l = 1:n-2
    meshb(l+1,:)= [point_start - no_new2 + fix(n-l),
                   point_start + l + 1,
                   point_start - no_new2 + fix(n-l-1)];
end
meshb(n,:) = [point_start-no_new2+ fix(1),
             (12 + ((((t+1)/2)-6)*no_new))+n,
             C];

fix2(1)=n;
for f=1:n-1
   fix2(f+1)=fix2(f)+n-f;
end
fix2(n)=C;
m=n+1;
p=1;
for f = 1 : n-1
    for d = 1 : n-f
        meshb(m,:) = [(point_start + p),
                      (point_start + p + (n-f)),
                      (point_start + p +1)];

        p = p + 1;
        m=m+1;
    end
    meshb(m-1,2:3)=(12 + ((((t+1)/2)-6)*no_new)) +
                                        [fix2(f+1),fix2(f)];
end
meshb(end,2)=B;

if t==11
   meshb(1:n,3)=
      [(fix(n-1:-1:1)' + 12 + 10*no_new + 9*no_new2);C];
   meshb(1:n,1)=
      [A;(fix(n-1:-1:1)' + 12 + 10*no_new + 9*no_new2)];
end

tri =[tri;meshb];
```

```
end

for t=12:2:20

    A = iso(t,1);
    B = iso(t,2);
    C = iso(t,3);
    m=1;

    mesha = zeros(sum(1:n-1),3);
    point_start = 12 +(10 *no_new + (t-11)*no_new2);
    p=n;

    fix = 1;
    for f = 2:n-1
        fix(f)=fix(f-1)+n-f+1;
    end
    for f = 1 : n-1
        for d = 1 : n-f
            mesha(m,:) = [(point_start + m),
                          (point_start + m - (n-f)),
                          (point_start + m +1)];
            m=m+1;
        end
        mesha(m-1,3) = (12 + ((t/2-1)*no_new) + p);
        mesha(f,2)=fix(n-f) + 12 + 10*no_new + (t-12)*no_new2;
        p = p + n-f;
    end

    tri = [tri;mesha];
    meshb= zeros(sum(1:n),3);
    meshb(1,:)=[A,point_start+1,point_start-no_new2+ fix(n-1)];

    for  l = 1:n-2
        meshb(l+1,:)= [point_start-no_new2+ fix(n-l),
                       point_start+l+1,
                       point_start-no_new2+ fix(n-l-1)];
    end

    meshb(n,:) = [point_start-no_new2+ fix(1),
                  (12 + (((t/2)-1)*no_new))+n,
                  C];
    fix2(1)=n;

    for f=1:n-1
        fix2(f+1)=fix2(f)+n-f;
    end

    fix2(n)=C;
```

```
        m=n+1;
        p=1;

        for f = 1 : n-1
            for d = 1 : n-f
                meshb(m,:) = [(point_start + p),
                              (point_start + p + (n-f)),
                              (point_start + p +1)];
                p = p + 1;
                m=m+1;
            end
            meshb(m-1,2:3)=
                (12 + (((t/2)-1)*no_new)) + [fix2(f+1),fix2(f)];
        end

        meshb(end,2)=B;
        tri=[tri;meshb];
    end
else
    tri=iso;
end

norm = sqrt(x.^2+y.^2+z.^2);
 x=x./norm;
 y=y./norm;
 z=z./norm;
 clear('norm');
```

C.2 Geodesic histogram clustering

```
/**
 ************************************************************
 * Function = [hist] = static_cluster(dx,dy,dz,power,tx,ty,tz)
 ************************************************************
 * Finds tx,ty,tz which is closest to each dx,dy,dz, and adds
 * the power associated with that signal to hist, the index
 * of tx,ty,tz.
 **/

#include "mex.h"

void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
    // ind=dot_mult(dx,dy,dz,tx,ty,tz)
 double *dx,*dy,*dz,*power,*tx,*ty,*tz;
 double dist, p_dist;
 double *ind;
 int d,t, dtotal, ttotal;
 int t_ncols, d_ncols,t_mcols, d_mcols;
 int best;
 // get d vector in;
 dx = mxGetPr(prhs[0]);
 dy = mxGetPr(prhs[1]);
 dz = mxGetPr(prhs[2]);
 // get d vector power
 power = mxGetPr(prhs[3]);
 // get array of t vectors in
 tx = mxGetPr(prhs[4]);
 ty = mxGetPr(prhs[5]);
 tz = mxGetPr(prhs[6]);

 // Get the dimensions of the matrix input t //
 // tx, ty, tz should all be same size //
  t_ncols = mxGetN(prhs[4]);
  t_mcols = mxGetM(prhs[4]);
  d_ncols = mxGetN(prhs[0]);
  d_mcols = mxGetM(prhs[0]);

  dtotal = d_ncols*d_mcols;
  ttotal = t_ncols*t_mcols;

  /* Set the output pointer to the output matrix. */
  plhs[0] = mxCreateNumericMatrix(t_mcols,t_ncols,
                                  mxDOUBLE_CLASS ,mxREAL);

  /* Set the output pointer to the ind */
```

```
  ind = mxGetPr(plhs[0]);

  // make zero
  for(t=0;t<ttotal;t++)
  {
      ind[t] = 0;
  }

  // for each input
  for(d=0;d<=dtotal-1;d++)
  {
    p_dist = (dx[d] * tx[0]) +
             (dy[d] * ty[0]) +
             (dz[d] * tz[0]);
    //ind[d] = 1;
    best = 0;
    for(t=1;t<=ttotal-1;t++)
    {
        //find the largest value.
        // On the bases that acos (largest dist) is closest.
        dist = (dx[d] * tx[t]) +
               (dy[d] * ty[t]) +
               (dz[d] * tz[t]);
        if(dist > p_dist)
        {
            best = t;
            p_dist = dist;
        }

    }
    ind[best] = ind[best] + power[d];
  }

}
```

Appendix D

Publications

Refereed conference presentations and papers

2008 O. Bunting, E.D. Chesmore, Instrument for soundscape recognition, identification and evaluation (ISRIE): Source separation. *In Proc. of the Institute of Acoustics Spring Conference: Widening Horizons in Acoustics*, Vol 30(2), Reading UK, 10-11 April 2008

2009 O. Bunting, J. Stammers, D. Chesmore, et al, Instrument for soundscape recognition, identification and evaluation (ISRIE): technology and practical uses. *In Proc. of Euronoise 2009*, 10pp, Edinburgh, Scotland, 26-28 October 2009

Non-refereed conference presentations

2008 O. Bunting, E.D. Chesmore, Soundscape source extraction using wavelet-based sparse representations. *In Proc of 2nd ASA-EAA Joint Conference, Acoustics '08*, Paris, France, 29 June - 4 July 2008

Published papers are included for reference in the following pages:

Proceedings of the Institute of Acoustics

INSTRUMENT FOR SOUNDSCAPE RECOGNITION, IDENTIFICATION AND EVALUATION (ISRIE): SOURCE SEPARATION

O Bunting Intelligent Systems Research Group, University of York, York, UK
D Chesmore Intelligent Systems Research Group, University of York, York, UK

1 INTRODUCTION

The monitoring of soundscapes is performed for many purposes. Such monitoring is routine in urban planning for both residential and industrial buildings, and also for assessing industrial noise pollution and residential noise complaints. Monitoring of soundscape is also prevalent in other research fields, and examples can be found from as far afield as investigating adverse medical effects caused by residential soundscapes[1], the effect of noise on wildlife[2], and monitoring ecological populations in national parks[3].

Typically, soundscape noise monitoring is expressed in A-weighted sound pressure levels, averaged over some period of time, often during the hours of night or day. The ISRIE project[4] aims to develop portable instrumentation to characterize a soundfield by localizing the constituent sources both spatially and temporally. Temporal localization would make it possible to automatically identify infrequent loud noise events such as military aircraft, pneumatic drills, or railways. These are potential sources of irritation in residential areas, yet only add a small contribution to A-weighted averaged levels.

Being able to decompose a soundscape enables more automated soundscape monitoring to existing standards such as PPG 24[5] and BS 4142[6]. It would also pave the way for a review of existing legislation. For rural or ecological soundscape monitoring, spatial and temporal localization of sound sources paves the way for improvements in automatic species recognition.

This research is part of a collaboration between York, Newcastle, and Southampton Universities, with each looking at various aspects of the ISRIE project. The legislative aspects are covered by Southampton university. Some sound propagation modeling and work using microphone arrays and wireless networks for monitoring soundscapes is covered at Newcastle. In York, research is on the separation and automatic identification of sounds. This paper outlines preliminary results achieved in the area of separation and localization.

2 BACKGROUND

2.1 Separation methods

Separation of the original sources from a set of signals from sensors such as microphones is referred to as source separation. This can be performed either blind, or with some *a-priori* knowledge of the sources to improve source separation. In recent years, independent component analysis has become an important statistical method used for performing blind source separation (BSS). This exploits three assumed properties of the sources: nonwhiteness, nonstationarity, and nongaussianity[7]. Unfortunately, this technique tends to be limited to the instantaneous case where there are equal numbers of sensors and sources. In a survey of various ICA methods by *O'Grady et al*[8], there are no known ICA methods that can separate in the likely conditions that ISRIE will face, i.e. more sources than sensors in a non-instantaneous mixing environment.

However if the blind constraint is relaxed somewhat, and further assumptions about the sources are allowed, the separation problem is referred to as semi-blind source separation. A popular assumption is that of sparsity of the sources in some domain; referred to as sparse separation. It is worth noting that for some sparse methods, this is the only assumption made about the source.

A separation methodology[9] based on binary masking according to source localization in the time-frequency (TF) domain has been shown to outperform existing pitch only algorithms for speech. Source localization was derived from cross-correlation augmented with information from interaural time differences (ITD) and interaural intensity differences (IID). Using binary masks in the TF domain has been shown to offer perfect separation for 2 sensor, N source anechoic mixtures, provided the sources do not overlap in the TF domain[10]. Sources are identified as clusters on a 2D histogram of relative delay between the 2 sensors and relative amplitude for each TF point. A real time implementation of this using K-means clustering also exists[11]. This TF masking theory has also been adopted and extended by the TIFROM algorithm[12], which allows sources to overlap in the TF domain.

These sparse approaches are very attractive, as they allow separation of under-determined mixtures (the case where there are more sources than sensors) to be separated even under anechoic conditions

3 SEPARATION

3.1 Development

In this paper, the idea of sparse separation is taken and developed into a method for separating sources from a coincident microphone array. The microphone use in this research is a soundfield ST350. The DUET algorithm uses a 2D histogram of relative delay and attenuation for each TF point, thus effectively calculating the direction of arrival (DOA) for each TF point.

By using a technique utilized for spatial impulse response rendering[13], we can calculate a direction vector for each TF point by calculating

$$I_a(\omega) = \frac{\sqrt{2}}{z}\,\mathrm{Re}\left\{W^*(\omega)\left(X(\omega)e_x + Y(\omega)e_y + Z(\omega)e_z\right)\right\} \quad \textbf{(1)}$$

Where I_a is a direction vector, W,X,Y,Z are Fourier transformed B-format signals, z is the impedance of air, and e_x, e_y, e_z, are unit vectors in the relative axis.

By using this as the basis of or separation, a bit mask can be created in the TF domain based on DOA in 3D, and the original sources can be separated.

3.2 Method

The proposed method is outlined in Figure 1.

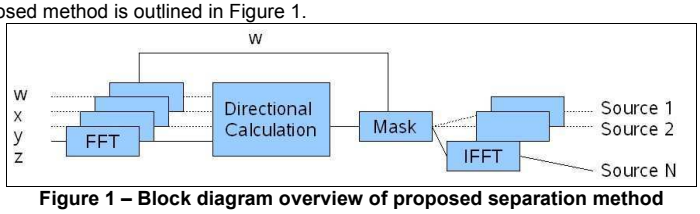


Figure 1 – Block diagram overview of proposed separation method

The B-format recording of a soundscape is divided into frames, windowed, and a Fourier transform applied. The directional vector for each TF point is then calculated. A mask is then applied to the Fourier transformed w channel for each source based on a desired direction, and TF points exceeding some tolerance of this ideal are rejected. The masked w channel is then converted back into the time domain for each source.

3.3 Example for 2 speakers

A recording of a male speaker and a female speaker were mixed into a virtual B-format with azimuth and elevation locations of (0°,0°) and (10°,20°) respectively. The time series data and the spectrogram for each can be seen below in Figure 2.

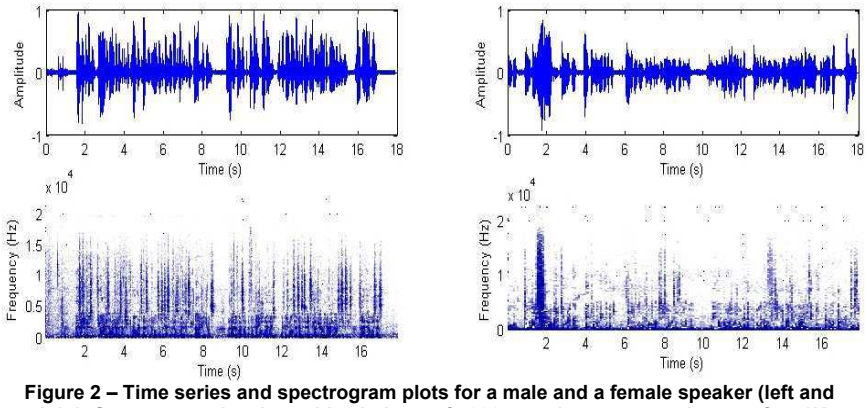


Figure 2 – Time series and spectrogram plots for a male and a female speaker (left and right). Spectrogram is taken with windows of 4096 samples, at a sample rate of 44.1K samples per second

This was then divided into frames 4096 samples long, with each frame overlapping by 50%. A Hanning window and a FFT were performed. Equation (1) was calculated for each TF point. As an aid to visualisation, the TF vectors are shown in Figure 3 mapped onto a geodesic histogram. The peaks at (0°,0°) and (10°,20°) are clearly discernable.

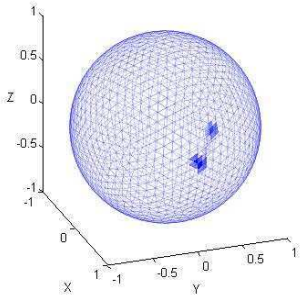


Figure 3 – Calculated directional vectors for TF points mapped onto a 3D geodesic histogram. Note the position of the main intensities at (0°, 0°) and (10°,20°).

The location of the sources, known *a-priori*, was used to inform the ideal locations for the TF masking process. Masking was performed within one degree of this ideal. An inverse FFT is then performed, and the separated sources recovered. See next section for results.

3.4 Results

The results for the previous example are shown in Figure 4

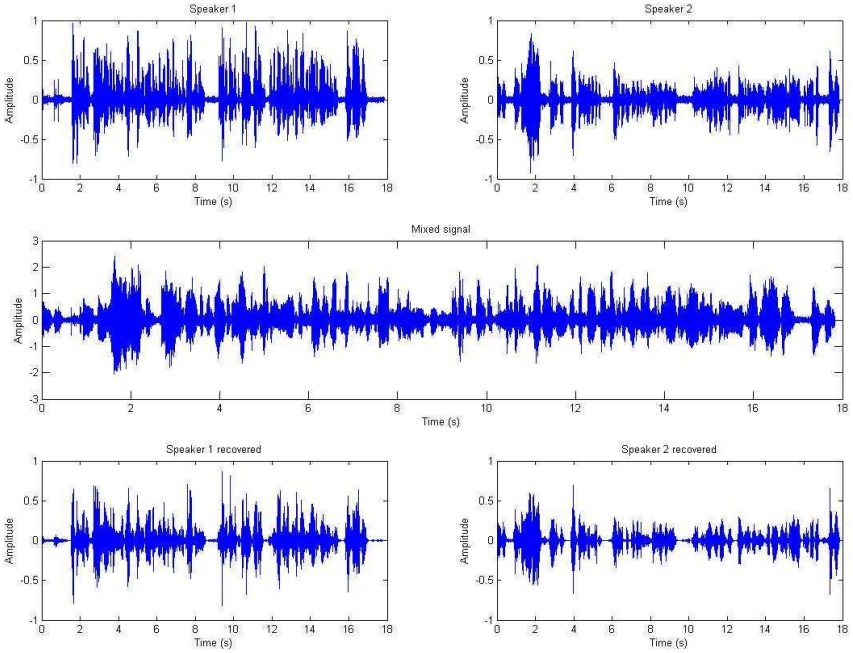


Figure 4 – The separation results for the case of two speakers at (0°,0°), (10°,20°)

The time series results for the separation shown in Figure 4 show a great similarity to the original data. Audibly, the speakers are intelligible, although artifacts have been introduced into both sources. These artifacts are due to TF points where the assumption of W-disjoint signals i.e. non-overlapping, was not valid. The relative power of these can be discerned from Figure 3, where the lightly coloured region between the two histogram peaks is due TF points that overlap between the 2 sources, causing the vectors to add and thus shift the DOA.

Although audibly the signal is degraded, the ultimate aim of ISRIE is to perform automatic identification of sources. Sound recognition and identification tasks have been shown to have robust algorithms even when artifacts are present[14].

4 CONCLUSION AND FURTHER WORK

4.1 FFT to Wavelet Transform

It has been demonstrated that the limiting factor in the application of this method is the assumption of W-disjoint sources in the soundscape. Using a STFT, the selection of the window length is critical to achieving the best compromise between time and frequency resolution, which is a task highly dependant on the mixture of sounds within the soundscape. It is therefore proposed that the method here is modified to use a wavelet transform.

An issue with a standard wavelet transform for this application is the complex data required for Equation (1) to be valid. It is therefore proposed that used be made of the dual tree complex wavelet transform (DTCWT)[15]. For an increase of processing by a factor of 2 compare to a standard wavelet transform, this performs two transforms 90 degrees out of phase from each other, allowing a direct replacement in this application.

Further benefit still would be use of wavelet packets to provide a redundant set of TF representations for the transformed data, combined with a best basis algorithm for maximising sparseness.

4.2 Clustering

The example given in Section 3.3 used *a-priori* information of the source location to provide the ideal directions for the making process. However, it is unlikely that the locations of all sources in the soundscape are known with sufficient accuracy to provide for separation using real recordings. Therefore, a clustering or decision based algorithm is required to inform the masking process. Due to the nature of the soundscape, such an algorithm would have to be capable of tracking clusters due to moving sources, and to be able to deal with sources appearing and disappearing.

To this end, research into a novel geodesic histogram based approach is being conducted, although other possibilities include k-means or a neural network approach.

5 ACKNOWLEDGEMENTS

The support of fellow ISRIE project members and members of the bio-inspired research lab in York is gratefully acknowledged. Namely Stuart Dyne, Christos Karatsovis, Gui Yun Tian, Hidajat Atmoko, Dave Chesmore, John Stammers, and Naoko Evans.

6 REFERENCES

1. A. Skånberg and E. Öhrström, 'Adverse Health effects in relation to urban residential soundscapes', J. Sound Vibrat. 250(1), 151-155 (2002)
2. J. L. Fletcher and R. G. Busnel, 'Effects of Noise on Wildlife', J. Acoust. Soc. Am, 65(3), 866-867 (March 1979)
3. R. C. Maher 'Obtaining Long-Term Soundscape Inventories in the U.S. National Park System', White paper, http://tinyurl.com/2jbfyb, (January 2004)
4. E. D. Chesmore, G Y Tian and S. J. C. Dyne, 'ISRIE – Instrument for soundscape recognition, identification and evaluation', EPSRC, (2006)
5. PPG 24: Planning and noise (1994)
6. BS 4142, 'Method for rating industrial noise affecting mixed residential and industrial areas', (1997)
7. H. Buchner, R. Aichner and W. Kellermann, 'Blind Source Separation for convolutive Mixtures, a unified treatment', (Ed. Y Huang and J Benesty) Audio Signal Processing, Kluwer Academic Publishers, Boston (2004)
8. P.D. O'Grady, B.A. Pearlmutter and S.T. Rickard, 'Survey of sparse and non-sparse methods in source separation', Int. J. Imag. Syst. Tech. 15, 18-33 (2005)
9. N Roman, D Wang and G.J. Brown, 'Speech segregation based on sound localisation', Proc. Int. Conf. Neur.Net. 4, 2861-2866 (2001)
10. Ö. Yılmaz and S. Rickard, 'Blind separation of speech mixtures via time-frequency masking', IEEE Trans. Signal. Proc, 52(7), 1830-1847 (2004)
11. S. Rickard, R. Balan and J. Rosca, 'Real-time time-frequency based blind source separation', Proc. ICA2001 (2001)
12. F. Abrard and Y. Deville, 'A time-frequency blind signal separation method applicable to underdetermined mixtures of dependant sources', Sig. Proc. 85(7), 1389-1403 (2005)
13. V. Pulkki and C Faller, 'Directional audio coding: Filterbank and STFT-based design', Proc. 120th Conv. Aud. Eng. Soc. (2006)
14. M. Cooke, P. Green, L. Josifovski and A. Vizinho, 'Robust automatic speech recognition with missing and unreliable acoustic data', 34(3), 267-285 (June 2001)
15. I. W. Selesnick, R.G Baraniuk and N.G. Kingsbury, 'The dual-tree complex wavelet transform' IEEE Signal Proc Mag 22(6) 123-151 (November 2005)

Edinburgh, Scotland
EURONOISE 2009
October 26-28

Instrument for soundscape recognition, identification and evaluation (ISRIE): technology and practical uses

Oliver Bunting
Jon Stammers
David Chesmore
University of York, YO10 5DD, UK

Omar Bouzid
Gui Yun Tian
University of Newcastle upon Tyne, NE1 7RU, UK

Christos Karatsovis
Stuart Dyne
ISVR Consulting, University of Southampton, SO17 1BJ, UK

ABSTRACT

Technological advancements in microelectronics and continuing research into signal characterisation and classification techniques have lead to promising results in developing an advanced sound meter. This instrument would be capable of characterising a sound field in terms of the relative contributions of the different noise sources. This paper provides an overview of this collaborative project, due for completion in October 2009, and the milestones that have been reached. In particular, the consideration and implementation of sensors and systems, the signal processing algorithms of source identification and classification, and the potential uses of the instrument in specific noise assessments in the UK are discussed.

1. INTRODUCTION

The collaborative work of three Universities; Newcastle upon Tyne, York and Southampton, has led to promising results in the development of an advanced sound meter that could provide a powerful measurement platform for many applications ranging from environmental noise assessments to the recording and evaluation of a variety of soundscapes.

Partners at the University of Newcastle upon Tyne have developed a multi-sensor technique for localising sound sources. In their particular method, the commercially available SoundField microphone probes have been used for 2D and 3D sound source localisation. Also, known beamforming techniques have briefly been investigated as an alternative technique for source localisation. Partners at the University of York have made use of a single SoundField microphone probe instead for developing a single-sensor technique for source localisation, separation and signal classification. Finally, partners at the University of Southampton have investigated the potential uses of ISRIE in existing noise legislation, planning and guidance and have also liaised with a wide range of stakeholders that could directly benefit from the use of such an advanced sound instrument.

2. ACOUSTIC SOURCE LOCALISATION

Over the course of the ISRIE project the co-authors at Newcastle University implemented an acoustic localisation system that is capable of locating a single sound source using at least three omni-directional microphones (i.e. 2D linear arrays) in a reverberant indoor environment with high accuracy for angle detection and small errors for distance estimation[1]. Sound source localisation in a 3D environment has been achieved by utilising the commercially available SoundField probes.

Figure 1 shows the use of three acoustic sensors in the context of a sound localisation system.

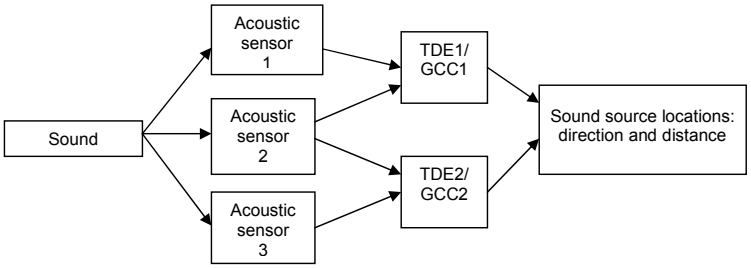


Figure 1: Three-microphone array system for acoustic monitoring[1].

The three acoustic sensors (omni-directional or 3D SoundField microphones) capture the sound simultaneously and the Time Delay Estimation (TDE) is extracted from any two sound signals from the three sensors using the Generalized Cross-Correlation (GCC). This method would ultimately derive sound source direction and distance through triangulation and geometric parameters. The three microphones are positioned in a straight line and the sides of the triangles formed by the source and each microphone represent the directional propagation paths from the source to each microphone. The direction of each propagation path is determined from the time differences between the signals arriving at the microphones. GCC is used to increase robustness to the adverse effects of early reflections and reverberation.

A. The 3 SoundField Microphone Method

Three SoundField SPS422B microphones were arranged in a straight line in order to achieve source localisation in a 3D environment[1]. Each microphone output is formed into a special signal format, the B-format, where four channels represent the velocity component in the three Cartesian directions; X (front-back), Y (left-right), Z (above-below) and one omni-directional signal, W, representing the pressure component. These signals are then fed into a PC for post-processing.

The Y and Z channel will generally be the same due the linear arrangement of the probes. The 2D configuration can be used for tilt and yaw estimation of sound direction in 3D. The X and W were therefore used for estimation in the experiment. With this arrangement, it has been possible to locate a single sound source in a reverberant indoor environment with an

accuracy of 1° for angle detection and errors less than 4% for distance estimation. A rearrangement of the soundfield array in the Z Cartesian direction was tested in order to provide estimates of yaw instead of azimuth angles. The W and Z microphone outputs were used for the estimation and the results were similar. The SoundField probes could therefore potentially be used in a commercial source localisation system, where the sensitivity of these microphones to sounds arriving from different directions will be applied to source localisation in planes other than that defined by the line of the array.

B. Beamforming Techniques

In the literature, beamforming is another suggested technique that has extensively been used in developing instruments for soundscape recognition, identification and sound source localisation[2, 3]. The beamforming technique is a technique that searches for a peak (or peaks) by achieving a full directional scan in order to determine the source(s) direction(s) from this (or these) peak(s). This can be achieved by delaying and summing the acoustic emitted signals to minimise the noise effects and enhancing (or maximising) the amplitude of the point (or direction) that represents the location of the sound source[2, 3]. The sound source can be considered to be in the near-field if the wavefront is modelled as spherical, whereas it is considered to be in the far-field if it is assumed to be planar[3]. The consequences of these assumptions are that in the near-field both the range and Direction of Arrival (DOA) can be computed, whereas in the far-field, only the DOA can be estimated due to computational costs[3]. Li[3] designed a flexible broad-band beamformer using nested Concentric Ring Array (CRA) that can be divided into sub arrays, where each sub array can cover a specified operating range. In our study, the acoustic camera, which mainly includes a microphone array of Star 36 sensors[4], a data-reader device, a laptop computer and the "NoiseImage" software[4], has been used for the investigation on flexible beamforming techniques and instrument validation. The data from this study is currently under investigation.

3. SOURCE SEPARATION

The task of automated recognition of audio signals is made considerably more complex by multiple sources being present in the audio recording, with a consequent reduction in recognition accuracy rates. To provide enhanced recognition accuracy, ISRIE employs a source separation algorithm prior to the recognition stages. The separation method developed for ISRIE is based on the assumption of W-disjoint orthogonality. That is, audio sources are sparse in a time-frequency domain. The sensor used is a Soundfield ST350, a B-format coincident microphone array[5, 6] that offers a more portable microphone system over the SPS422B.

A. Model

Consider a 3-dimensional coincident array comprising of 3 orthogonal sets of figure-of-eight microphones and an omni-directional microphone at the centre of the array. Given the location of the sources, the B-format mixture of signals in the anechoic case can be expressed as:

$$\begin{vmatrix} w(t) \\ x(t) \\ y(t) \\ z(t) \end{vmatrix} = \begin{vmatrix} 1/\sqrt{2} & \dots & 1/\sqrt{2} \\ \cos(\theta_1)\cos(\lambda_1) & \dots & \cos(\theta_N)\cos(\lambda_N) \\ \sin(\theta_1)\cos(\lambda_1) & \dots & \sin(\theta_N)\cos(\lambda_N) \\ \sin(\lambda_1) & \dots & \sin(\lambda_N) \end{vmatrix} \begin{vmatrix} s_1(t) \\ . \\ . \\ . \\ s_N(t) \end{vmatrix} \quad (1)$$

where x, y, z are the mixtures observed on the Cartesian axis, w is the mixture observed by the omni-directional sensor, and θ, λ are the azimuth and elevation for the direction of arrival of a particular source.

B. Assumptions

Separation of a given mixture is subject to two conditions on the source mixture being met. These are W-disjoint orthogonality[7] and radial sparsity. These are described formally below.

W-disjoint Orthogonality

Two sources s_i and s_j are W-disjoint orthogonal if the following condition is met.

$$S_i(\omega,\tau)S_j(\omega,\tau) = 0 \qquad \forall\ i \neq j, \omega, \tau \quad (2)$$

where $S(\omega,\tau)$ represents the time-frequency domain transformation of $s(t)$.

Radial Sparsity

This a condition placed on the geographical location of the sources. Each source must have a unique direction of arrival at the sensor.

$$(\theta_i, \lambda_i) \neq (\theta_j, \lambda_j) \qquad \forall\ i \neq j \quad (3)$$

C. Direction of Arrival (DOA) Calculation

Provided the above conditions have been met, the DOA of the B-format audio signal can be calculated in the time-frequency domain using a method from Directional Audio Coding Scheme (DirAC)[8, 9].

$$\vec{D}(\omega,\tau) = -\Re\left(W^*(\omega,\tau) * \left(\vec{e}_x X(\omega,\tau) + \vec{e}_y Y(\omega,\tau) + \vec{e}_z Z(\omega,\tau)\right)\right) \forall\ \omega,\tau \quad (4)$$

where $\vec{e}_x$, $\vec{e}_y$ and $\vec{e}_z$ are unit vectors along the Cartesian axes.

D. Source Location Estimation

Using the calculated DOA vectors, it is possible to perform source localisation using a variety of techniques. Perhaps the simplest is to construct a histogram over an arbitrary time period, and look for peaks. This method, along with another clustering method based on self-learning neural networks, has been looked at to perform this task.

E. Demixing

For each source location, which is denoted E_i, M_i describes a bit mask in the time-frequency domain for each source.

$$M_i(\omega,\tau) = \begin{Bmatrix} 1 & | & \arccos\left(\frac{\vec{E}_i}{|\vec{E}_i|}\cdot\frac{\vec{D}}{|\vec{D}|}\right) \le \delta \\ 0 & | & otherwise \end{Bmatrix} \quad \forall i \tag{5}$$

where δ provides a user defined angular margin around the source location.

The sources can then be recovered by using the mask to filter W in the time-frequency domain.

$$\hat{S}_i = M_i(\omega,\tau) * W(\omega,\tau) \tag{6}$$

from which $\hat{s}_i$ can be gained by performing an inverse time frequency transformation.

F. Results

Table 1 shows the results from a signal separation experiment.

Table 1: Results from a signal separation experiment.

Speaker	Performance Measure				Location	
	Signal-to-Interference Ratio (SIR) in mixture	SIR after masking	SIR gain	Preserved Signal Ratio (PSR) after masking	azimuth	elevation
1	-0.17 dB	12.14 dB	12.32 dB	12.32 dB	120	0
2	-2.96 dB	12.30 dB	15.27 dB	15.27 dB	280	10
3	-6.81 dB	10.89 dB	17.70 dB	17.70 dB	340	20

The separation algorithm was tested on a mixture of three male speakers reading passages from a novel. Each speaker was recorded independently under anechoic conditions, and the mixture created by the summation of the three B-format recordings. The recordings were performed in this manner to allow analytical comparison of the separated speakers with the original recording. Speakers one and two show much higher Preserved Signal Ratio (PSR) results compared to speaker three. This is perhaps unsurprising, considering that speaker three has an initial Signal-to-Interference Ratio (SIR) of −6.81 dB. All the speakers are intelligible on listening, although there is an appreciable level of crackling on speaker three. The SIR gain for all speakers shows excellent results, showing high suppression of the interfering speakers, with an average improvement in SIR of 15 dB. These results compare well to those listed for mixtures of two speakers[10].

As far as the validity of the assumptions, W-disjoint orthogonality has been shown to be a valid assumption for speech signals. Acoustic niche theory also suggests an evolutionary pressure for this to be the case in the animal kingdom. However, the authors concede that in the general case, the assumptions are not guaranteed to hold true. Further investigations into the applicability of these assumptions to a range of situations need to be performed.

4. SIGNAL CLASSIFICATION

ISRIE will also perform the classification of the separated audio signals which are provided by the signal separation as discussed previously. The output of the classification algorithms will advise the user of ISRIE which category of sounds a particular signal belongs to. It is assumed that the input signal to the classification system contains only one sound source.

A. Sound Categories

A taxonomy of sound categories has been devised specifically for the purpose of ISRIE. Figure 2 illustrates these categories.

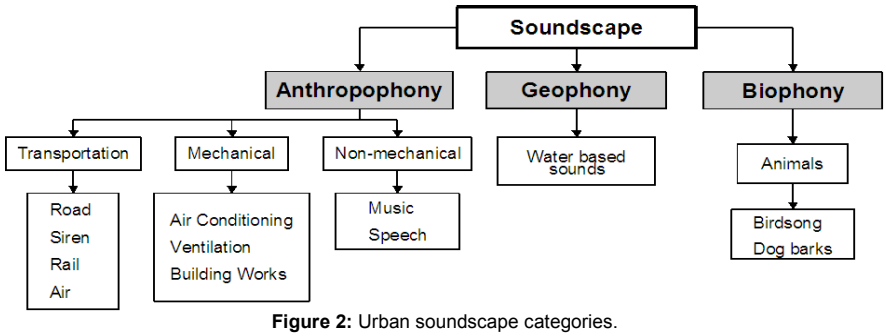


Figure 2: Urban soundscape categories.

Initially, the soundscape is split into three main categories. *Anthropophony* relates to sounds made or caused by human activity, *biophony* sounds are those made by animals, and *geophony* encompasses sounds not caused by either of the above.

B. Classification using Time-Domain Signal Coding

A typical classification system consists of two components: a feature extractor and a classifier[11]. There is sometimes a third component to provide some pre- or post-processing either at the input or output to the system. The data that is to be classified will be passed into the feature extractor whose role it is to reduce the complexity of the data before it reaches the classifier[11] thus optimising the classification process. A good overview of a selection of these techniques can be found in the comparison made by Cowling and Sitte[12].

The feature extraction method that has been used for data reduction in ISRIE is known as Time-Domain Signal Coding (TDSC). This is a purely time-domain analysis method which has previously shown to be successful in the identification of wood-boring insects[13] and in the classification of different Orthoptera[14]. The data produced by the TDSC algorithm describes a waveform by the number of samples (duration - D) and number of minima (shape – S) contained within each epoch (signal between 2 consecutive zero crossings) of the waveform. The D-S information is stored for a given frame of the waveform by means of a codebook. After a signal has been analysed using TDSC, each code within the codebook will have a number of occurrences associated with it to describe its D-S characteristics. It is this frequency information, the S-matrix, which is then used for classification. A more detailed explanation of how TDSC was developed and the other features it can extract from the full

bandwidth signal is given by Chesmore[14]. Figure 3 shows how the TDSC analysis fits into the classification system.

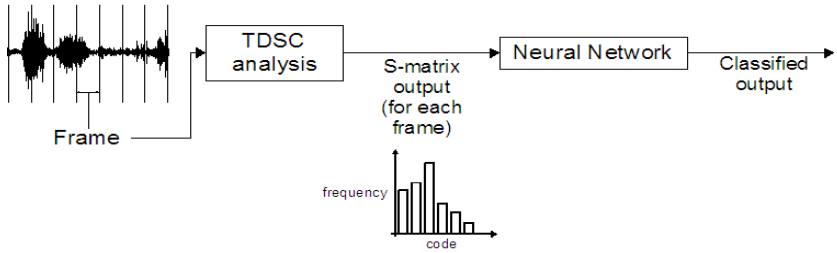


Figure 3: Proposed classification system. The S-matrices for each frame of the waveform are classified individually.

It was decided that a neural network approach in the classification would be adopted. Initially, an unsupervised Self-Organising Map (SOM) network was used but this struggled to differentiate between the test pieces of audio data. Significant improvements in classification were gained by introducing supervised learning into the system. A Learning Vector Quantisation (LVQ) network was implemented using the LVQ1 learning rule[15, 16]. Eight different categories of sounds were placed into 4 groups: group 1 contained air traffic, air conditioning and ventilation units, and building works; group 2 contained road and rail traffic; group 3 contained birdsong and also recordings of crickets; and group 4 contained some speech examples. The grouping of the sounds was chosen based on how consistent the signal was throughout the duration of the recording. After training was completed using a training set of 40 recordings, the network was tested using a 30-second test audio file which combined audio from each of the 4 groups. Network accuracy for each of the individual groups was poor for all but group 1 (88%). However, when combined results were observed for how well the system could recognise non-bioacoustic audio (groups 1 and 2), the accuracy rose to 93%. This shows that it is possible to perform an initial classification using the relatively simple methods discussed above. Work is now focused on developing the system further to incorporate classifiers to differentiate between the various bioacoustic and non-bioacoustic signals. Feed-forward neural networks with backpropagation training are being experimented with and are showing positive initial results.

5. APPLICATIONS

The uses of ISRIE could range from assisting acoustic consultants and planners in making the right decision on the most appropriate control measures in a project where noise concerns may arise, through to assisting soundscape artists and sound engineers with the recording of isolated sound events for either artistic reasons or for the subjective evaluation of different soundscapes. The usefulness of ISRIE in environmental noise impact assessments, such as PPG 24[17], BS 4142[18] and noise nuisance applications have previously been discussed[19]. Over the course of this research project, different stakeholders have also been interviewed in order to assess what measurement parameters would be required from such an instrument to log and what would be the additional benefits from the use of such an instrument.

A. BS 4142

In BS 4142 assessments, ISRIE could potentially be used to obtain the specific noise level L_{Aeq} of a source and the background noise level L_{A90} without requiring the need to measure these descriptors separately. The instrument would offer individual logged values of these two environmental noise level descriptors in order to establish the arithmetic difference between the intruding mechanical noise level and the typical background noise level without the presence of any mechanical plant or industrial noise. Also, in practice, there are instances where it is not possible to obtain separate measurements of these two descriptors, because either the mechanical source cannot be turned off in order to measure the background noise level, or the mechanical noise cannot accurately be quantified at the receptor's location due to interference from other sources, such as transportation related noise. ISRIE would be capable of deriving these parameters through its discrimination and classification algorithms as discussed above.

B. PPG 24

In PPG 24 assessments, the existing environmental noise levels are established over a 24-hour measurement period, when planning a new housing development. The measurements are normally unmanned for economic reasons since they cover such an extensive measurement period. Firstly, it is apparent that in mixed soundscapes, where for example there is almost an equal contribution of railway and road traffic noise, it is difficult to quantify the contributing noise sources, or even determine which is the dominant noise source. Therefore, it is not always feasible to establish the most representative noise source category in which the noise environment should be assessed in. ISRIE would be useful in obtaining these individual contributions in L_{Aeq} terms in order to decide which is the prominent noise source in that specific environment. Secondly, ISRIE would automatically log and classify individual events that exceed a certain criterion, such as 82 dB $L_{A,max,S}$ and assess whether these transient events are intrusive sources of noise, e.g. mechanical, or non-intrusive, e.g. birdsong or sounds from other animal life. This type of automated assessment is not possible with the use of current technology since the noise survey is normally unmanned and these individual transient events can only be evaluated and assessed at the post-processing stage.

C. Noise Nuisance

Environmental Health Officers (EHOs) of Local Authorities in the UK would make use of an advanced sound instrument for various reasons. Firstly, ISRIE would enable them to investigate complex noise complaints in the case where it is not clear which mechanical plant noise source affects the complainant's house in a highly built-up area. Secondly, the problem of low frequency noise, potentially originating from tunneling or drilling works, can be an issue for some residents in a community. These noise complaints can be difficult to assess with the current technology of sound level meters and ISRIE's characterisation capability would work well in these types of problem where the source is of tonal character. Thirdly, ISRIE would aid in monitoring the noise from music events and assist EHOs in reaching decisions upon the licensing of commercial premises that may give rise to noise complaints.

D. Other Engineering Consultancy Problems

The use of a conventional sound level might not be adequate in some cases since there can be interference from other noisy equipment when trying to quantify a particular noise source in an industrial area. There are also instances, where the noise of certain installations, such as

electrical transformers, cannot easily be quantified because either these installations are near sources of transportation noise, e.g. motorways, or because there are other electro/mechanical installations nearby that may contribute to the overall measured level. Also, as part of the Land Compensation Act, difficulties can arise when trying to establish only the road traffic components at houses that are situated miles away from a newly constructed or modified road. ISRIE would be capable of solely measuring the traffic noise components from the remaining background noise, something that is not possible with the current sound level meters. Similar measurement problems can arise when trying to quantify noise solely emanating from racing tracks that might affect nearby communities.

E. Soundscape Recordings

Recordings of soundscapes is developing in many applications ranging from creating archived sound recordings of a variety of animal sounds through to the recordings of any other types of soundscape for recreating experiences in art installations, museums and galleries. The need for carrying out recordings of sounds in isolation is important in many applications. At the moment, in order to separate different sounds, noise suppression techniques are used in order to filter out the remaining sound, or the recording is delayed until the level of the intrusive noise has dropped to such a level that it is not significantly contributing to the overall level. ISRIE would be useful in recording these sounds as isolated events and hence providing a reference instrument for sound recording.

F. Future Policy

ISRIE could enable planners to consider the balance between 'positive', e.g. natural sounds and 'negative' sounds, e.g. mechanical-like sounds in a mixed sound environment as part of a regeneration plan for improving the quality of life in urban agglomerations or assist in the design of new spaces of personal enjoyment and recreation in metropolitan cities. The first step would be to establish which types of sound are considered 'wanted' and 'unwanted' in that environment. Then, ISRIE would be used as an instrument to establish the current percentage of wanted and unwanted sounds through its source discrimination and classification algorithms as presented above. Finally, the management of these sounds would involve standard noise abatement techniques along with the potential introduction of more wanted sounds. In the end, ISRIE could be used to assess whether the desired 'mix' of wanted and unwanted sounds was achieved.

5. CONCLUSIONS

The need of a network sensor system with the development of algorithms and techniques for automatically characterising sounds in a complex sound environment is more evident than ever before. This paper has presented a number of suggested measurement platforms for the measurement of sounds along with promising techniques for signal separation and classification. The use of ISRIE could ultimately revolutionalise the way we currently perceive soundscapes and could affect the way we measure, assess and record sounds in the future.

ACKNOWLEDGMENTS

University of Southampton, York and Newcastle gratefully acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for sponsoring this research work.

REFERENCES

1. H. Atmoko, T. Gui Yun and B. Fazenda, **"**Accurate sound source localization in a reverberant environment using multiple acoustic sensors", Meas. Sci. Technol. Feb. 2008, pp. 1-10.
2. Terence Betlehem, "Acoustic Signal Processing Algorithms for Reverberant Environments", PhD Thesis, Department of Information Engineering, School of Information Sciences and Engineering, Australian National University, Nov. 2005.
3. Yunhong Li, "Broadband Beamforming and Direction Finding Using Concentric Ring Array", PhD Thesis, the Faculty of the Graduate School, University of Missouri-Columbia, Jul. 2005.
4. Acoustic sound source localisation: Download: Acoustic Camera: Applications and System Overview (PDF), Available at: http://www.acoustic-camera.com/pdfs/ac_brochure2009.pdf, Accessed: May 2009.
5. Michael Gerzon. Periphony: With-height sound reproduction. Journal Audio Eng. Soc., 21(1), pp2–10, 1973.
6. Michael Gerzon. The design of precisely coincident microphone arrays for stereo and surround sound. In Proc. 50th Convention of the Audio Eng. Soc., 1975.
7. S. Rickard and Z. Yilmaz. On the approximate w-disjoint orthogonality of speech. In Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '02), 1, pp 529–532, 2002.
8. J. Merinaa and V. Pulkki. Spatial impulse response rendering. In Proc. of the 7th Int. Conf. on Digital Audio Effects (DAFx'04), pp 139–144, October 2004.
9. Ville Pulkki. Spatial sound reproduction with directional audio coding (DirAC). Journal Audio Eng. Soc., 55(6), June 2007.
10. O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. IEEE Journal. Sig. Proc., 52(7), pp1830–1847, 2004.
11. R. Beale and T.O. Jackson, Neural Computing: An Introduction, Hilger 1998.
12. M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition", *Pattern Recognition Letters* 24, pp. 2895-2907 (2003).
13. I. Farr and E. D. Chesmore, "Automated bioacoustic detection and identification of wood-boring insects for quarantine screening and insect ecology", in *Proceedings of the Institute of Acoustics* 29, Pt. 3, pp. 201-208 (2007).
14. E.D. Chesmore, "Application of time domain signal coding and artificial neural networks to passive acoustical identification of animals", *Applied Acoustics* 62, pp. 1359-1374 (2001).
15. T. Kohonen, "Improved Versions of Learning Vector Quantization", *International Joint Conference on Neural Networks* 1, pp. 545-550 (1990).
16. H. Demuth and M. Beale, Neural Network Toolbox User's Guide, The MathWorks, Inc. 2001.
17. Planning Policy Guidance 24: Planning and noise, Department of the Environment, 1994.
18. BS 4142: 1997: Method for rating industrial noise affecting mixed residential and industrial areas, BSI.
19. C. Karatsovis and S J C Dyne, "Instrument for soundscape recognition, identification and evaluation: an overview and potential use in legislative applications", in *Proceedings of the Institute of Acoustics,* 2008, Vol. 30, Pt.2.