

Maurice Fallon

Acoustic Source Tracking using Sequential Monte Carlo

This dissertation is submitted for the degree of
Doctor of Philosophy

June 6, 2008



Darwin College
University of Cambridge

Abstract

Particle Filter-based Acoustic Source Localisation algorithms track (online and in real-time) the position of a sound source — a person speaking in a room — based on the current data from a microphone array as well as all previous data up to that point.

The first section of this thesis reviews previous research in this field and discusses the suitability of using particle filters to solve this problem. Experiments are then detailed which examine the typical performance and behaviour of various instantaneous localisation functions.

In subsequent sections, algorithms are detailed which advance the state-of-the-art. First an orientation estimation algorithm is introduced which uses speaker directivity to infer head pose. Second an algorithm is introduced for multi-target acoustic source tracking and is based upon the Track Before Detect (TBD) methodology. Using this methodology avoids the need to identify a set of source measurements and allows for a large saving in computational power.

Finally this algorithm is extended to allow for an unknown and time-varying number of speakers. By leveraging the frequency content of speech it is shown that regions of the surveillance space can be monitored for activity while requiring only a minor increase in overall computation. A variable dimension particle filter is then outlined which proposes newly active targets, maintains target tracks and removes targets when they become inactive.

Preface

This thesis has been the significant focus of my life for the past three and half years. There are a number of people, without whom it could not have been completed and to whom I offer my gratitude.

I would first like to address my thanks to my supervisor, Professor Simon Godsill, for his support and feedback throughout the last three years as well as his patience in considering and developing my often poorly formed ideas into some structured and successful research. My gratitude also goes to Microsoft Research (Cambridge) for the provision of funding.

I would like to thank Jonathan Cameron for his help and patience with the many practical issues I faced, be it with the motion capture system or numerous minor technical issues. In addition I would like to thank Adam Johansen and Edmund Jackson for their helpful advice, and often instruction, at the outset of my studies. Many thanks also to Richard Wareham, Simon Hill and Paul Peeling for their helpful comments and proofreading of this thesis.

Outside of my studies I would like to acknowledge my friends — in particular my friends from Darwin College and the members of Cambridge GAA — for providing me with some separation and diversion from my studies.

To Ellie, I would like to give a special *thank you*. At many stages during the research and writing stages you gave me the strength and focus I needed to continue. I am so lucky to have you.

Finally to my parents and family I would like express my grateful thanks for their help and assistance. I would like to acknowledge the sacrifices that my parents, Anne and Bernard, made throughout my upbringing to give me many opportunities to express and further myself. *Go raibh maith agaibh go léir.*

Declaration This dissertation is the result of work carried out by myself between October 2004 and May 2008. It includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. This dissertation has not been submitted in whole or in part for a degree at any other university. It contains approximately 56,000 words and 48 figures.

Maurice Fallon

Cambridge

June 2008

Contents

1	Introduction	1
Part I Literature Review		
2	Acoustic Source Tracking	7
2.1	Problem Framework	7
2.1.1	Human Hearing and Localisation	8
2.1.2	Recording Environment	9
2.2	Signal Model	9
2.2.1	Complicating Factors	10
2.2.2	Simplified Signal Model	11
2.3	Measurement Functions	13
2.3.1	Generalised Cross-Correlation	14
2.3.2	Steered Beamforming	20
2.4	Other Localisation Methods	24
2.5	Recordings and Simulations	27
2.6	Conclusions	29
3	Sequential Monte Carlo Methods	31
3.1	Bayes' Theorem	31
3.1.1	Bayesian Parameter Estimation	32
3.1.2	Model Selection	32
3.2	Monte Carlo Integration	33
3.3	Recursive Bayesian Estimation	34

3.3.1	Kalman Filtering	35
3.3.2	Extended Kalman Filtering	36
3.3.3	Grid-based Methods	37
3.4	Sequential Monte Carlo	38
3.4.1	Importance Sampling	38
3.4.2	Sequential Importance Sampling	39
3.4.3	Bootstrap Filtering and Resampling	39
3.4.4	Other Advances	41
3.5	Conclusions	41
4	Bayesian Acoustic Source Tracking	43
4.1	Use of Kalman Filtering	44
4.2	General Framework	44
4.2.1	State Vector	45
4.2.2	State Dynamical Model	45
4.3	Measurement Models	46
4.3.1	GCC-based tracking	47
4.3.2	SBF based tracking	48
4.3.3	Other Measurement Functions using within the Literature	50
4.4	Behavioural Advances	50
4.4.1	Alternative Dynamical Models	51
4.4.2	Voice Activity Detection	53
4.4.3	Target Track Initiation and Removal	55
4.5	Related Tracking techniques	56
4.5.1	Audio Video Fusion	56
4.5.2	Multi-target Acoustic Source Tracking	58
4.6	Conclusions	59
Part II Contributions		
5	Experimental Analysis of Acoustic Source Tracking	63
5.1	Experiments with the Generalised Cross-Correlation	64

5.1.1	Speech Directionality and Microphone-to-Source Relative Angle	64
5.1.2	Distance from Source to Microphone Pair	69
5.1.3	Distance Separating Sensors	69
5.1.4	Distribution and Bias versus Background Noise and Reverberation	70
5.1.5	Two Source Experiments	72
5.2	Experiments with the Steered Beamformer Function	73
5.2.1	Characterising the SBF Distribution	73
5.2.2	Frequency Integration Range and Density of SBF Grid ..	76
5.2.3	Source Movement	78
5.2.4	Ambient Noise Level	79
5.2.5	Two Source Experiment: Interference between Sources ..	82
5.3	Conclusions	83
6	Orientation Estimation and Tracking	87
6.0.1	Previous Approaches	87
6.1	Estimating Orientation using Speech Directionality	88
6.1.1	Problem Framework	89
6.1.2	Identifying Source Peaks	89
6.2	Dynamical Models	90
6.3	Measurement Likelihood Model	92
6.3.1	Localisation Likelihood Functions	92
6.3.2	Orientation Likelihood Functions	93
6.3.3	Other Likelihood Functions	96
6.4	Experiments	98
6.4.1	Experimental Setup	98
6.4.2	Typical Tracking Examples	98
6.4.3	Monte Carlo Simulations	100
6.5	Conclusions	101
7	Track Before Detect Acoustic Source Tracking	103
7.1	Observations drawn from Recorded Audio Data	104

7.2	Accommodating Physical Observations	105
7.3	Proposed Framework	106
7.3.1	Source Dynamical Model	106
7.4	Track Before Detect	107
7.4.1	Adapting TBD to Acoustic Source Tracking	108
7.4.2	Magnitude Mapping	109
7.4.3	Activity Indicator Variable	112
7.5	Experimental Performance Testing	114
7.5.1	Comparison with other algorithms	115
7.5.2	Variable Optimisation	119
7.5.3	Monte Carlo Simulation Results	125
7.6	Multi Target Acoustic Source Tracking	127
7.6.1	Multi Target Track Before Detect	127
7.6.2	Joint Tracking: Widely spaced sources	130
7.6.3	Tracking more than one closely positioned source	130
7.6.4	Transition between tracking mechanisms	132
7.7	Multiple Target Tracking Results	133
7.7.1	Illustrative Results	133
7.7.2	Performance Evaluation	135
7.8	Further Work	135
7.9	Conclusions	137
8	Multiple Speaker Tracking for an unknown number of targets	139
8.1	Tracking target activity	139
8.2	Existence Grid	141
8.2.1	Design Choices	141
8.2.2	Desirable traits of the Existence Grid	143
8.2.3	Evaluating the Grid	143
8.2.4	Existence Grid Likelihood Functions	145
8.2.5	Target Proposal Mechanism	145
8.2.6	Existence Grid Optimisation	147
8.3	Tracking Frameworks	149

8.4	Framework 1: Defined Target-Source Assignment	151
8.5	Framework 2: Undefined Target-Source Assignment	153
8.5.1	Data Model	154
8.5.2	Proposal Mechanism	156
8.5.3	Importance Weights	157
8.5.4	Discussion	159
8.6	Experiments	160
8.6.1	Intermittent Single Speakers Examples	160
8.6.2	Two Speakers Examples	162
8.6.3	Monte Carlo Simulations	165
8.7	Conclusions	168
Part III Conclusions		
9	Conclusion	171
9.1	Contributions	171
9.2	Future Direction	174
Part IV Appendices		
Practical Implementation Details		177
A.0.1	Ground Truth	179
A.1	Real-time Implementation	179
Notation		181
B.1	Terms and Abbreviations	182
B.2	Measurement Metrics	183
Details of Alternative Multi-target Tracking Algorithm		185
C.1	Framework 1: Defined Target-Source Assignment	185
C.1.1	Initialising and terminating target tracks	188
C.1.2	Particle Update Procedure	188
C.1.3	Discussion	191
References		193

Introduction

This thesis presents a number of novel Sequential Monte Carlo-based algorithms to track the locations of human speakers in a typical room environment. This problem, commonly known as Acoustic Source Tracking (AST) or Acoustic Source Localisation (ASL), has application in improving speech acquisition, audio-video conferencing, automatic meeting analysis and remote computer interfacing. It is a problem, that, despite being seemingly straightforward, is complicated by many factors which a human listener would disregard. These factors include signal corruption due to noise sources and additional speakers in the recording environment; the non-stationarity of human speech and the effect of reverberation.

The algorithms presented operate online and may be implemented in real time on a typical home computer. The thesis is divided into three main parts.

Chapters 2 to 4 provide an introduction to the problem of Acoustic Source Tracking and some background regarding Sequential Monte Carlo (SMC) methods before discussing previous research in which particle filters (as SMC methods are often called) were used to track moving speech sources. Some discussion is given to the applicability of such a model-based approach to this problem.

The second parts, Chapters 5 to 8, introduces some experimental analysis of the problem and a number of novel techniques to solve the acoustic source tracking problem. First, experimental analysis of speech in a realistic recording environment is carried out in Chapter 5. The behaviour of two common

instantaneous localisation methods, the Generalised Cross-Correlation (GCC) and the Steered Beamformer (SBF), are studied while varying a number of the problem parameters. These parameters include the distance between the sensors and the source, the level of background noise, the effect of speaker motion as well as the presence of an interfering source. The correct choice of design parameters specific to each method is also examined. This analysis is then used to motivate the proposal of algorithms in the subsequent chapters.

Chapter 6 proposes an algorithm which estimates orientation by considering the effect of speaker directivity. This algorithm utilises the observation that recordings of the speaker from a pair of microphones placed in front of a speaker are expected to exhibit greater signal-to-signal correlation than those from a similar pair placed behind the speaker's head. Joint orientation and location tracking for a moving source is then demonstrated.

Chapter 7 reforms the Steered Beamformer-based likelihood function presented in [93] within the Track-Before Detect framework. In doing so the proposed algorithm recognises that the gradient of the SBF surface is directly related to the wavelengths of the recorded speech signal. A discretisation of this surface is then proposed which efficiently utilises the available computing power. This approach also avoids the formal assignment of position measurements to a particular source and in turn allows a simple extension of the algorithm to track two or more simultaneously active sources.

Chapter 8 extends this multi-target tracking algorithm to probabilistically monitor the surveillance region for intermittent speakers using a variable dimension particle filter drawn from the general tracking literature. The algorithm uses an *existence grid* to monitor the surveillance region for changes in activity and to propose new source positions. Meanwhile an importance weighting mechanism incorporates hypothesised prior behavioural information, the previous particle positions and the current measurement data to create a corrected weighting of the particle likelihoods. This particle distribution can then be used to infer the number and location of the active speech sources.

Finally in Chapter 9 some conclusions and a unified summary of the results of this thesis are presented. Suggestions as to promising areas for future research in this field are also given.

Part I

Literature Review

Acoustic Source Tracking

Tracking of acoustic source locations, be they military vehicles [26] or people speaking [28] is an established field within both the tracking and speech signal processing communities. From tracking military vehicles using a wireless ad-hoc microphone system dropped from a aeroplane onto a battlefield, to precisely localising the mouth of the driver in a moving car for hands-free audio capture, the environments and limitations are often very different. Because of this vast variety of situations, the approaches taken are often radically different. Hence the first step in this thesis is to better define the specific acoustic source tracking problem which will be considered.

2.1 Problem Framework

Consider N_m spatially distributed microphones surrounding the centre of a typically noisy and reverberant room — with no particular physical layout. There will be no microphone spacing constraints, however, we will simplify the problem by limiting this study to localisation in the two horizontal dimensions only, the $\mathcal{XY}$ -plane. Tracking in the third dimension is indeed possible without a great increase in complexity, but is not considered to be imperative at this stage.

The room will be assumed to be a normal medium-sized office or living room. As such typical dimensions of the room will be in the order of 7m by 7m by 3m. The noise and reverberation characteristics of the room are assumed to

be stationary over time, as is the speed of sound at 343m/s. The microphone positions will be known and fixed during all experiments.

For further motivation, we will now discuss an expected application for acoustic source tracking. It is envisaged that a solution should be implemented using a reasonably small number of microphones so as to allow for easy installation in a conference or office room. The solution is intended to be implemented using a real-time online system and it is desirable to minimise computation where possible, so that any such algorithm can be implemented on a typical computer or low-powered device. Furthermore the general solution will utilise cheap low-quality microphones (typically electret microphones) which will be chosen to be as omni-directional as possible, again to maintain flexibility.

Experiments have been carried out using the so-called Huge Microphone Array (HMA) at Brown University [84] which contains as many as 512 microphones randomly positioned on a set of panels surrounding the room and carried out beamforming and localising in real time. The Large Acoustic Data Array (LOUD) at MIT [94] produces in the range of 50MB of audio data every second and uses an experimental *scalable polymorphic computer architecture* called RAW. The system utilises a single rectangular array of 1024 microphones in total. However both systems require dedicated computer architectures and I/O schemes to analyse the vast amounts of data generated. Despite all of this equipment, the expenditure and the expertise that has gone into implementing the algorithms, one may recall the localising ability of just one pair of sensors: the human ears.

2.1.1 Human Hearing and Localisation

Though not the focus of this work, it would be remiss not to mention human auditory perception and the way in which the human hearing system performs localisation. As discussed by Moore [66] the human observer uses both time and intensity differences between the two ears to localise speakers. Moore goes on to state:

Experiments confirm that the extent to which the cues of inter-aural time and intensity differences are used in localisation of pure tones is strongly related to what would be predicted from the physical nature of these cues.

From this we can draw reassurance that a signal content-based approach to this problem is not at odds with the lessons learned by evolution. However we must recognise that higher-order processes within the brain go far beyond the complexity of what digital signal processing has achieved.

Moore goes on to recognise the effect of *head-shadow*¹ on the spectral content of the incoming signals, which combined with the possible movement of the head itself, are influential cues for source localisation. These effects will of course not be available in the envisaged recording setup.

A final piece of information discussed by the author is that the human hearing system has improved localisation at the onsets and offset of transient signals. This effect has not been leveraged by those researching Acoustic Source Tracking (AST) as yet.

2.1.2 Recording Environment

From this discussion, it would seem reasonable to focus on using relative time delay information between signals recorded by a pair of sensors — in this case two ears — to determine source direction. To localise equally well across a typical room requires an array with a minimum of approximately 12 microphones. The array of microphones and the experimental setup used in this thesis can be seen in Figure 2.1.

2.2 Signal Model

So as to build up a basic signal model, we will initially assume that only one source is active initially. The relationship between the source signal and the signal received at sensor m , $x_m(t)$, is as follows

¹ Head-shadow is the absorption of sound waves of wavelengths less than the dimensions of the human head.



Fig. 2.1. *The Experimental Recording Laboratory*

$$x_m(t) = h_m(t) * s(t) + e_m(t) \quad m \in \{1, N_m\} \quad (2.1)$$

where $s(t)$ is the source signal (that is the speech content spoken by a person). The acoustic room impulse response (RIR) between the source and the microphone m is denoted $h_m(t)$, while $e_m(t)$ represents additive noise recorded at that microphone.

2.2.1 Complicating Factors

In theory, if it was possible to estimate this impulse response for each recorded signal, the acoustic source localisation problem could easily be solved. The direct path component of the response would indicate the distance between the microphone and the source's position, then using a combination of two or more such direct path estimates the source location could be found. However the room impulse response can be very challenging in even a moderate noisy environment. A typical RIR function is shown in Figure 2.2. It is a complex function: rapidly changing and very difficult to estimate. Indeed experimentally estimating RIR functions requires precise calibration of the behaviour of the recording instruments themselves — something we do not envisage.

Furthermore as the impulse response between the source and the microphone encompasses the response of the speaking person's head and the microphone response functions further variables are introduced.

A common empirical quantification of a room’s acoustic behaviour, related to the RIR, is the Room Reverberation Time. This is defined as the amount of time taken for a sound to die away (due to absorption) to a specific level — usually set to be 60dB below the original energy level. Typically the 60db reverberation time, RT_{60} , of an office room is of the order of 0.5 seconds.

Meanwhile, it is often assumed that speech is stationary for short segments of the order of 20 to 30ms and analysed in frames of this length. As a result a typical far-field recording² will contain portions of the direct path signal mixed with the reverberant signal from a previous utterance. While the addition of such reverberation can be reconciled by complex processing within the human ear, the effect it has on a signal processing applications should not be underestimated.

A final complication is that a speaker’s orientation and position can change quickly causing drastic changes to the RIR in even a very short period of time. This is further reason to use short frame-lengths³.

For these reasons we will choose to simplify the signal model. We will also make the obvious observation that, regardless of the localisation function chosen, in periods of speech inactivity (typically pauses between sentences and words) no method could provide a location estimate. Hence the remit of the localisation function used herein can best be described as *a function which gives an accurate estimate of the source location, during source activity*.

Excepting this argument, in Section 2.4 we will discuss some methods which have attempted to approximate the direct path portion of the RIR.

2.2.2 Simplified Signal Model

Continuing this argument, the signal model will be simplified to the direct path model — essentially ignoring the existence of reverberation. The received

² Far-field recordings encompass all recordings made at a substantial distance from the speaker - as opposed to near-field recordings such as those from clip-on lapel microphones.

³ Related to the difficulty of estimating RIR functions is the problem of realistically simulating RIR’s and resultant signals (for use in algorithm testing). This is discussed in Section 2.5

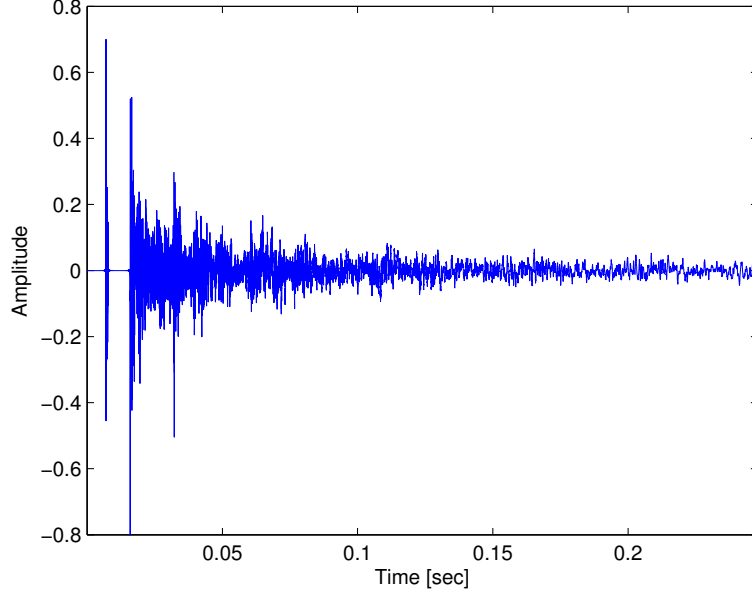


Fig. 2.2. *Room Impulse Response of a medium damped room (courtesy of [52]). Note that this plot represents only the first approximately 10% of the entire response.*

signal is simplified to be the transmitted signal simply delayed by the time taken for the resultant pressure waves to propagate from the source position to the microphone

$$x_m(t) = a_m s(t - \tau_m) + e_m(t) \quad (2.2)$$

where attenuation and delay parameters at a particular microphone are a simple function of the distance between source and sensor, d_m , as follows

$$a_m \propto \frac{1}{d_m} \quad \tau_m = \frac{d_m}{c} \quad (2.3)$$

where c is the speed of sound. The amplitude parameter, a_m , will not be considered for estimation because of the high variability of the overall speech energy level. Estimating the delay parameter τ_m for each microphone or, as will be the case, estimating a derived function of this parameter will be the focus of this work.

Frame-by-frame estimates

To allow for online operation of the algorithm and frequent location estimate updates, the recorded data will be batched into frames of samples. Assuming a frame of synchronised data of L samples is available at time frame k from sensor m :

$$\mathbf{x}_m(k) = [x_m(kL), x_m(kL + 1), \dots, x_m(kL + L - 1)], \quad (2.4)$$

we form a N_m by L matrix of data from all the sensors for this frame, viz

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_1(k) \\ \vdots \\ \mathbf{x}_{N_m}(k) \end{bmatrix}. \quad (2.5)$$

In the following section a number of measurement or *localisation* functions will be introduced. They attempt to make a transformation between the audio frame data and a location estimate $\hat{l} = \mathbf{f}(\mathbf{X}_k)$.

2.3 Measurement Functions

Delay path measurement models are typically divided into two groups. The first group are those that provide individual indirect measurements from each microphone or microphone pair (such as the Generalised Cross Correlation), which are then combined in some manner to give an overall location estimate.

A second group uses the entire microphone data frame to make a single estimate of overall received signal correlation at a particular physical location having hypothesised that the source was located at that location such as the Steered Beamformer (SBF). This correlation measure, called the Steered Response Power (SRP) can then be used as an estimate of how likely a particular location originated the sound.

Each method has its advantages and disadvantages, which will be discussed in the following sections.

2.3.1 Generalised Cross-Correlation

The Cross-Correlation Function can be used as a relative Time Delay of Arrival (TDOA) estimator. An approximation to the function, as derived in Chapter 4 of [28], is expressed as

$$C_{x_1x_2}(\tau) \approx c_N \int_{\Omega} X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega \quad (2.6)$$

where Ω is the range over which the integration is carried out, c_N represents a scaling constant related to the integration range and $X_i(\omega)$ represents the Fourier transform of signal i . $X_1(\omega)X_2^*(\omega)$ is taken as an approximate estimate of the cross power spectral density, $G_{x_1x_2}(\omega)$. The function estimates the amount of sample-by-sample correlation between the signals as a function of the relative delay between them.

Assuming the delay-only signal model, the cross correlation of two signals shifted by a relative delay of τ_{12} , evaluated at a the time τ , is given by

$$C_{x_1x_2}(\tau) = \mathbb{E}[x_1^*(t)x_2(t - \tau)] = a_1a_2C_{ss}(\tau - \tau_{12}) + C_{e_1e_2}(\tau). \quad (2.7)$$

where source signal auto-correlation is $C_{ss}(\tau)$ and the noise signal cross-correlation is $C_{e_1e_2}(\tau)$. If the noise processes are uncorrelated, $C_{e_1e_2}(\tau) = 0$, leaving just the shifted noise-free auto-correlation function which will exhibit a peak at the correct relative delay.

However the shape of this auto-correlation function is dependent on the frequency content of the source signal. For example, if the source signal is dominated by a single frequency (such as voiced speech) then the resultant correlation function will suppress the effect of weaker frequencies and will principally be a periodic function of this dominant frequency and exhibit peaks other than the true TDOA, making estimation of the correct relative delay difficult. This effect is illustrated in Figure 2.3. To mitigate against this problem, characteristics of the signal or noise may be levered by pre-filtering.

Knapp and Carter [53] introduced a general framework for pre-filtering processors used to emphasise the true delay peak and to degrade the effects of signal structure or noise. The Generalised Cross-Correlation (GCC) was proposed as

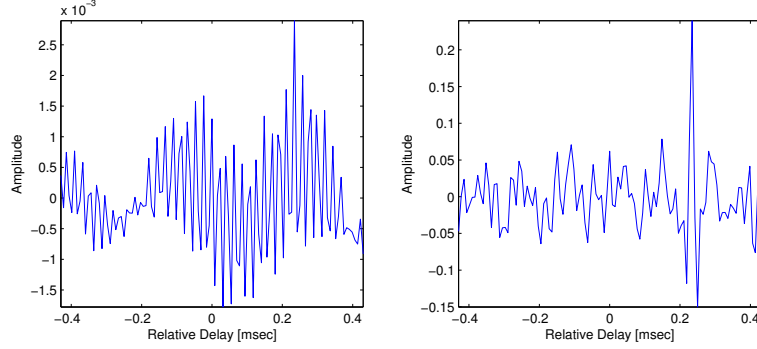


Fig. 2.3. Regular CCF (left) and PHAT-GCC function (right) for the same segment of recorded voiced speech — illustrating how the PHAT-GCC has removed the signal frequency content giving a clearer relative delay peak

$$R_{x_1x_2}(\tau) = \int_{\Omega} \Psi(\omega) X_1(\omega) X_2^*(\omega) e^{j\omega\tau} d\omega. \quad (2.8)$$

where $\Psi(\omega)$ is the filtering function, where again $X_1(\omega)X_2^*(\omega)$ is an approximate estimate of the cross power spectrum of the signals. The cross-signal power spectral density (CPSDF) of the GCC is related to that of the basis cross-correlation as follows

$$\hat{G}_{x_1x_2}(\omega) = \Psi(\omega) G_{x_1x_2}(\omega) \quad (2.9)$$

where $\Psi(\omega)$ is the pre-filtering function.

The authors go on to identify a number of optimal processors which can be used when knowledge of the frequency content of either the noise or source signal is available, such as the Eckart and ML filters. We, however, will assume that no such information is available. Instead we will concentrate on the Phase Transform (PHAT) introduced therein which uses a frequency weighting term to reduce the effect of signal periodicity.

Phase Transform

First note that the CPSDF of the ideal direct path signals in Equation 2.6 is as follows

$$\begin{aligned} G_{x_1x_2}(\omega) &= \mathbb{F}[C_{x_1x_2}(\tau)] \\ &= a_1 a_2 G_{ss}(\omega) e^{j\omega\tau_{12}} + G_{e_1e_2}(\omega) \end{aligned} \quad (2.10)$$

where a delay in the time domain is equivalent to a phase shift in the frequency domain⁴, $\delta(t - \tau) \Leftrightarrow \mathbb{F}^{-1}[e^{-j\omega\tau}]$. $G_{ss}(\omega)$ is the source power spectral density while $G_{e_1e_2}(\omega)$ is the noise cross-power spectral density. Assuming that the noise signals are uncorrelated this density will simplify to zero once more. As such the exponential delay factor, $e^{j\omega\tau_{12}}$, contains the delay information we are interested in. We wish to suppress all other components of this equation (i.e. the effect of the signal frequency content $G_{ss}(\omega)$) to isolate the delay information.

The Phase Transform normalises the amplitude component of each frequency of the cross-correlation and is as follows

$$\Psi(\omega) = \frac{1}{|G_{x_1x_2}(\omega)|}. \quad (2.11)$$

This allows the generalised cross-correlation between two ideal signals to be simplified as follows

$$\begin{aligned} \hat{G}_{x_1x_2}(\omega) &= \Psi(\omega)G_{x_1x_2}(\omega) \\ &= \frac{[a_1a_2G_{ss}(\omega)e^{j\omega\tau_{12}} + G_{e_1e_2}(\omega)]}{|G_{x_1x_2}(\omega)|} \\ &= \frac{a_1a_2G_{ss}(\omega)e^{j\omega\tau_{12}}}{|a_1a_2G_{ss}(\omega)e^{j\omega\tau_{12}}|} \\ &= e^{j\omega\tau_{12}} \end{aligned} \quad (2.12)$$

so that the corresponding function becomes simply a delta function at the correct delay

$$R_{x_1x_2}(\tau) = \mathbb{F}^{-1}[\hat{G}_{x_1x_2}(\omega)] = \delta(t - \tau_{12}). \quad (2.13)$$

This function is much more robust to the periodicity of the input signal than the simple CCF, however it can be regarded as a whitening filter as it destroys the amplitude information of each frequency component. As intended this has the effect of de-emphasising the strong frequency components. However it will also amplify the components with little or no signal energy — components whose phase is likely to be erratic — perhaps uniformly distributed in the interval $[-\pi, \pi]$ — thus reducing the performance of the overall function. While there have been attempts to modify the weighting function to exclude frequency samples with little or no energy and to improve performance in noisy

⁴ where $\mathbb{F}[\cdot]$ denotes the Fourier transform.

environments, the PHAT transform has remained popular, mainly because of its ease of use and simplicity.

Performance of the PHAT-GCC

In unchallenging recording conditions, the delay corresponding to the direct path delay difference is expected to be the peak of the GCC. However, this will not always be the case in situations of considerable reverberation, multiple sources and, of course, speaker silence. In such a scenario the peak of the GCC may instead be corrupted by environmental noise or may even be entirely erroneous leading to a completely incorrect estimate of the source position for this frame.

Figure 2.4 shows a sequence of GCC functions formed using real audio recordings. It represents a speaker moving across a room in front of a microphone pair. We will refer to this representation as a *GCC trace* although it has previously been called a *delayogram* [85]. Gaps in the trace are caused by silent pauses between words resulting in incorrect estimates.

Note that this function is unrelated to the concept of a *Global Coherence Field* or acoustic map, which is the surface formed when combining the GCC function contributions from each microphone pair for each point on a grid within the surveillance region [12, 13]. Such a field is equivalent to the SBF surface discussed in Section 2.3.2.

Champagne et al. [17] went further and studied the performance of TDOA estimation as a function of increasing reverberation time, RT_{60} . The authors showed that when reverberation time reaches a certain threshold the proportion of erroneous estimates abruptly increases until the method becomes useless for RT_{60} greater than approximately 0.6ms. As a result speech recorded in environments with long reverberation times can not be localised using this method.

A further complication is the effect of speech directivity, caused by the human head transfer function [6, 34]. When a speaker is facing away from a microphone pair, the resultant GCC function will exhibit lower performance due to a decreased portion of direct signal energy to reverberant signal en-

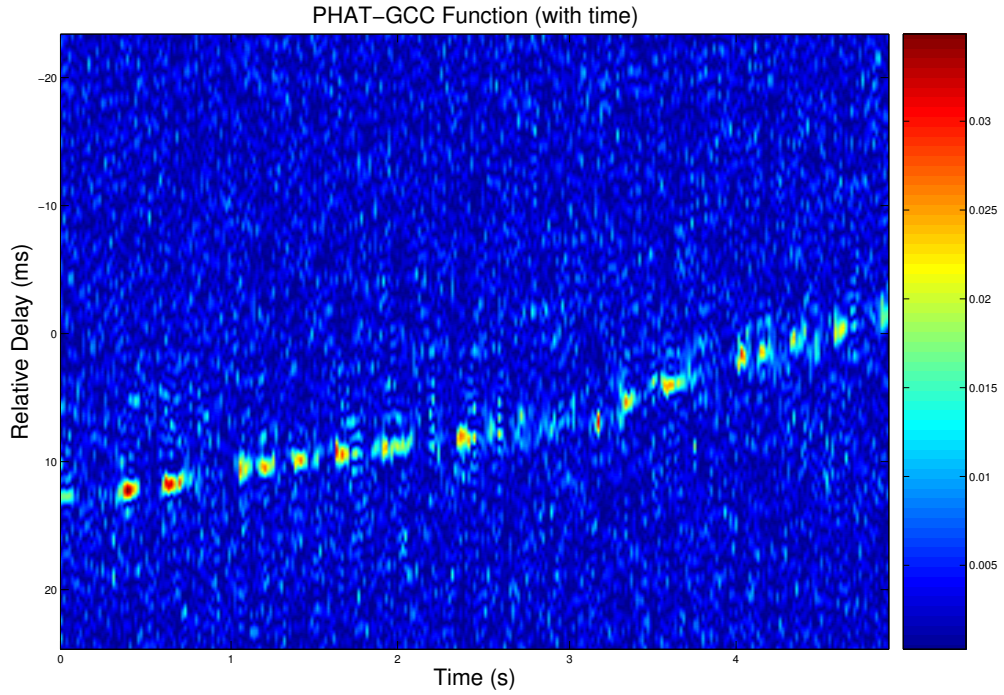


Fig. 2.4. *Evolution of the PHAT-GCC function for a real moving human speaker. Note the silence gaps that exist in the function over time.*

ergy at the microphone. Hence a quickly turning and fast moving source will greatly complicate any GCC-based source tracking algorithm. The effect of orientation and how it may be estimated is further discussed in Chapter 5 and Chapter 6.

Localisation using the PHAT-GCC

Related cross-correlation methods have used the harmonic structure, pitch or frequency content of the speech signal to improve performance of the TDOA estimator.

Raykar et al [77] propose improving delay estimation for voiced speech by using concepts from Linear Prediction analysis. Each signal is first transformed, via the tenth-order LP residual and the Hilbert enveloping function into a function whose periodicity will be a more robust than that of the ac-

tual speech signal. The cross-correlation of the resultant output (from two microphones) was shown to be more accurate than that of the PHAT-GCC.

Also focusing on the voiced portion of speech Brandstein [10] utilises a Multi-Band Excitation (MBE) Speech Vocoder to determine the fundamental pitch of a frame of audio data. Having determined the pitch and its harmonics, an alternative weighting of the pre-filtering weighting function in Equation 2.9 is applied which attenuates the portions of the spectrum not deemed to be active.

This and other approaches [11, 24, 25] assume accurate knowledge of the signals' spectra so as to determine the maximum likelihood (ML) weighting for the pre-filtering weighting function. The typical assumption is that in periods of silence the spectra of the background noise can be evaluated. Later when the speaker is active, the signal can be optimally amplified/attenuated in the frequency domain. However, this algorithm is susceptible to the effects of reverberation [10].

Finally the relative amplitude of the two microphones is typically not used for measuring the time delay of arrival, as mentioned in Section 2.2.2. However Chen et al [21] combined an average magnitude difference function (AMDF) with the usual GCC function to estimate both parameters of the direct path model. The performance improvement demonstrated by the authors was not substantial. Readers are directed to [9] and [22] for a more complete review of the various advances discussed above.

Combining Estimates to Localise Speakers

The shape of the relative delay function for a fixed delay is hyperbolic and is illustrated in Figure 2.5 for two dimensions, although in the far field the function is typically approximated as a bearings-only estimator. Delay estimates from individual microphone pairs must be combined in some way to give the full two or three dimensional estimates of a source's location while coping with the effects of noise and reverberation — which may mean that some of the individual delay estimates may be entirely erroneous.

A number of instantaneous frame-by-frame approaches have been put forward, including Least Squares in two dimensions [9] and a 3-D simplex search

[96]. The reader is directed to the former for a more complete discussion and comparison of non-model-based pairwise-TDOA localisation strategies. However, as recognised by a number of the researchers in the field, the dominant usage of individual GCC functions in the late 1990's was mainly motivated by the computationally complexity of other methods.

However the computational expense required for these methods is no longer considered to be prohibitive. Following this train of thought leads one to methods which instead combine all the incoming data to create stand-alone 3-D estimates without triangulation.

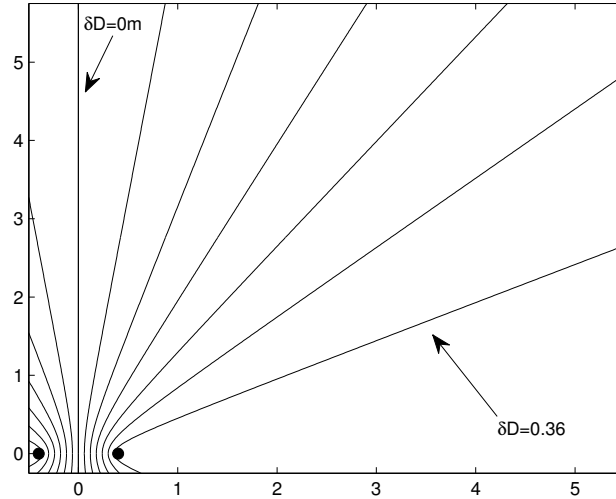


Fig. 2.5. Graph showing hyperbolic lines of equal relative distance between two markers spaced 80cm apart for a range of relative distances.

2.3.2 Steered Beamforming

The Steered Beamformer (SBF) function is a measure of correlation across a batch of signals for a set of relative delays and is often seen as an indirect measure of how likely it is that the full batch of audio recordings, from a microphone array, originated at a specific location. The delay-and-sum beam-

former⁵ steered to the physical location $l = [x \ y]$ is given by

$$\mathcal{S}(l) = \int_{\Omega} \left| \sum_{m=1}^{N_m} S_m(\omega) W_m(\omega) e^{j\omega T_m(l)/c} \right|^2 d\omega \quad (2.14)$$

where the measured quantity itself is known as the Steered Response Power (SRP). The Euclidean distance between the steering location and the known position of microphone m is $T_m(l) = \|l - l_m\|$.

The frequency range over which the integration is carried out is denoted Ω . A number of design choices must be considered when determining this range. These choices are discussed and experimented with in Chapter 5. Finally the weighting function, $W_m(\omega)$, will, again, be chosen as the phase transform $W_m(\omega) = (|S_m(\omega)|)^{-1}$, which in this case is individual to each signal.

Relationship with GCC

To better emphasise the relationship between the GCC and the SBF, we will briefly sketch the derivation of the SBF in terms of the GCC. A more detailed version of the proof can be found in Section 6.3 of [28].

In the frequency domain the filter-and-sum beamformer is defined as

$$Y(\omega, \tau_1, \dots, \tau_{N_m}) = \sum_{m=1}^{N_m} G_m(\omega) X_m(\omega) e^{-j\omega\tau_m} \quad (2.15)$$

where in our application, the steering vector, $(\tau_1, \dots, \tau_{N_m})$, will typically be the set of delay vectors for a particular physical location. Substituting this into the general representation of the SBF we obtain

$$\begin{aligned} S(\tau_1, \dots, \tau_{N_m}) &= \int_{\Omega} Y(\omega, \tau_1, \dots, \tau_{N_m}) Y^*(\omega, \tau_1, \dots, \tau_{N_m}) d\omega \\ &= \int_{\Omega} \left(\sum_{l=1}^{N_m} G_l(\omega) X_l(\omega) e^{-j\omega\tau_l} \right) \left(\sum_{q=1}^{N_m} G_q^*(\omega) X_q^*(\omega) e^{-j\omega\tau_q} \right) d\omega \end{aligned}$$

After rearrangement and making the assumption that the signals have finite energy, the SBF is represented in terms of GCC functions

$$S(\tau_1, \dots, \tau_{N_m}) = \sum_{l=1}^M \sum_{q=1}^M R_{x_l x_q}(\tau_q - \tau_l). \quad (2.16)$$

⁵ For continuity we will maintain the same notation used by Lehmann and Johnansson, [58]

The steered response power is thus the sum of all possible pairwise GCC crossings time shifted for the set of steering delays.

Finally, it should be noted that, while the SBF is a combination of the individual GCC functions, it is still questionable as to whether it is appropriate to compare the correlation between two signals recorded at opposite sides of a room and thus in opposite directions from the speaker. Two such signals would be expected to be largely uncorrelated meaning that the overall SBF function will be little effect by these components.

Computation and Implementation Considerations

The task of implementing a multi-channel recording system, in real-time, with an array of in the region of 10-16 microphones is not to be underestimated. Recording across all these channels and delivering the recorded audio to the signal processing algorithm on a digital signal processing chip with uniform latency is imperative for any localisation algorithm. As most recording equipment pairs channels along a stereo line, one must be careful when comparing inter-pair channels. This data collecting complication has also contributed to making pair-wise GCC microphone setups more common than cross-array SBF setups.

Another reason is the computation necessary to implement the SBF. Each steered response power evaluation involves complex multiplications for each microphone sample and then full calculation of the entire surface (in either two or three dimensions) at a sufficient density is necessary to determine all regions of high SRP — in real-time. A number of researchers recognised this to be prohibitively expensive [30, 49] and non-model based search approaches often avoided using the SBF entirely until recently [22, 96].

While one comparison between a simple grid-based SBF method and a search-based GCC method [96] illustrates the former outperforming the GCC, it also requires four orders of magnitude more arithmetic operations. Although it should be noted that the system used a very large array of microphones and it is not possible to assert from the paper the amount of computation necessary to achieve *similar* performance.

Alternatively it has been suggested [28] that, for an 8-element array, a SBF-based method requires 23 times the computation required by the GCC-based method. The author also suggests that *“to achieve similar results [to that achieved with the SBF], GCC-PHAT requires a significant increase in data requirements (over 10 times the data)”* (Chapter 8, [28]).

Finally as an alternative inverse approach for searching the SBF grid, Dmochowski et al [29] instead evaluates the SBF — not in terms of location — but rather in terms of relative delays between microphone pairs. The SBF is then built up using a set of basis functions whose weights are determined by the level of cross-correlation between the microphone pairs. The authors suggest that this results in a reduction in computation by an order of magnitude without significant loss in accuracy.

Despite such advances, real-time operation of a complete SBF search of the full dimension problem is still a prohibitive operation.

Localisation using the SBF

A number of frame-by-frame strategies have attempted to minimise overall computation by evaluating the SBF function not over the entire surface but only in regions of interest or by evaluating it on a limited resolution grid [28] or by using multi-stage strategies [30, 97]. Another strategy [97] suggested implementing a type of gradient descent-based genetic algorithm.

In general these strategies involve repeatedly evaluating the steered response power with the aim of finding the global SRP maximum. The final location is then chosen to be the source position at that time. However taking a different approach — a model-based approach — involves proposing likely solutions based on past estimates and then determining the function in only these likely functions. Such methods will be discussed in the Chapter 4.

2.4 Other Localisation Methods

In this section we will review a number of alternative localisation methods suggested in the literature, in this and other fields, and discuss their applicability to our problem.

Phase Unwrapping

One of the earliest TDOA methods proposed was Phase Unwrapping, [9, 84]. As illustrated in Equation 2.12, the complex phase difference between two ideal signals at a particular frequency is simply the product of that frequency and the relative time delay between the signals, $\angle(S_{x_1x_2}(\omega)) = \tau\omega$. Thus, dividing this factor from the product of the fourier coefficients gives a delay estimate at each frequency. A least squares estimate of the overall signal delay can then be formed over all the shifts.

While this property holds true at low frequencies; at higher frequencies the phase will be wrapped around. Noting that higher frequencies contain more precise delay information than low frequencies, this issue would be detrimental to fully accurate localisation. To overcome this problem, the authors use an initial delay estimate at the lower frequencies to *unwrap* the higher frequencies and then fit a secondary delay to the estimates, weighted by the frequency magnitude.

Although this method is intuitively simplistic, illustrations from [84] demonstrate instability at higher frequencies and the method has not been used recently.

Independent Component Analysis-based Delay Estimation

As mentioned above meaningful estimation of the phase is only possible at a particular frequency, ω , if

$$|\omega\tau| < \pi \quad (2.17)$$

Another approach is to instead reduce the maximum possible delay between the microphones. For two microphones separated by d , this delay is $\tau_{\max} = d/c$ where c is the speed of sound. Yilmaz and Rickard [95] suggest placing

the microphones with a separation of $d \leq 8.60\text{cm}$. As a result all frequencies below 4kHz will not suffer phase wrap-around — regardless of the position of the source.

The authors go on to implement the DUET algorithm — which independently estimates the relative delay between each time-frequency bin of the recorded signals' respective Short Time Fourier Transforms (STFT). The algorithm then clusters the bins with similar delay which assumes that the audio portion in the bin originated from the same source. These clusters are finally used to partition the STFT between different sources, so as to perform blind source separation for instantaneous (non-reverberant) speech.

While an extension of this algorithm for localisation in an anechoic room has been implemented, [65], the previous publication demonstrated that delay clustering performance plummets when tested within a typical office environment.

Linear Spatial Correlation

Another approach recognises that delays across a number of microphone pairs must agree for resultant estimates to be feasible. For example consider a set of three microphones m_1, m_2 and m_3 . The relative time delay from a source to m_1 compared to m_2 is τ_{21} and similarly τ_{31} for microphones 1 and 3. This implies that the third relative delay, between m_2 and m_3 must be

$$\tau_{32} = \tau_{31} - \tau_{21} \quad (2.18)$$

Using a linear array of microphones with uniform separation Chen et al [19, 20, 23] propose utilising this redundant information via a *Spatial Correlation Matrix*. Proposing an initial relative delay, the signals are uniformly shifted to correct this delay. The determinant for the shifted signal frame is then evaluated. If the proposed shift is close to the correct delay, the determinant will be large. Maximising the determinant for all possible delays will give the best time delay estimate.

This proposed algorithm has the ability to improve delay estimation when compared with the GCC. However to estimate source location in two or three

dimensions it would be required to combine estimates from more than one multi-element array.

Adaptive Eigenvalue Decomposition

Huang et al [5, 46] proposed a so-called adaptive eigenvalue decomposition algorithm (AEDA). This algorithm attempts to estimate the direct delay path of a signal without resorting to a delay-only signal model.

The authors recognised that for two recorded signals, $x_1(k)$ and $x_2(k)$, of a single source, $s(k)$, with respective source-to-sensor impulse responses of h_1 and h_2 :

$$x_1(k) * h_2 = s(k) * h_1(k) * h_2 = x_2(k) * h_1 \quad (2.19)$$

which in the noiseless case gives us the equality

$$x_1^T(k) * h_2 - x_2^T(k) * h_1 = 0 \quad (2.20)$$

at time frame k . Using this, they then go onto derive an adaptive LMS algorithm to jointly estimate the two impulse responses, utilising the assumption that the combined impulse response is the eigenvector of the signal covariance matrix whose eigenvalue equals 0.

The authors suggest that convergence of the algorithm and TDOA estimation takes approximately 250ms in demonstrations in a varechoic chamber, [5]. Such performance is not envisaged to be sufficiently responsive to changes in the cross-signal impulse response for a moving source. Also it is not clear how the algorithm might be extended to multi-source scenarios. Furthermore substantial testing and verification of this algorithm has not been forthcoming — suggesting that it is not sufficiently robust or stable in challenging audio environments.

Finally one of the main advantages envisaged by the authors is that the adaptive portion of the algorithm allows the system to ignore non-speech events such as doors opening or keys swinging — essentially because the system is not responsive to these type of events. Such temporal complications can of course be avoided using the model-based approaches discussed in the following chapter. For these reasons this TDOA method has not been considered.

Microphone Position-based Systems

Finally a number of approaches have been implemented to utilise specific microphone layouts or equipment. Teutsch and Kellermann [88] used a circular array of microphones set into a rigid baffle to create specific acoustic wave properties that can be used to estimate delay. Hioka and Hamada [45] utilise cross-signal correlation across a 4-element microphone tetrahedron to estimate location in two dimensions.

Finally an interesting application by Nakadi et al [69] have attempted to simulate the hearing system of a human being by placing two microphones on either side of a robot head made of acoustic absorbing material. This system artificially creates a head related transfer function including inter-aural intensity difference (IID) as well as inter-aural phase difference (IPD). The authors go on to illustrate a joint hearing and vision speaker localisation strategy which mimics the human hearing function.

We have discounted such approaches as they either require expensive equipment, solve a very specific problem or modify the recorded signals such that post-processing tasks, i.e. noise cancellation, would be hampered.

For further information about instantaneous localisation, the reader is directed to the review in [8].

2.5 Recordings and Simulations

Finally, having compared and discussed the various localisation algorithms which have been proposed, it would be remiss not to comment on the data used to examine such algorithms.

A number of the interesting algorithms mentioned above have, unfortunately, only been tested with simulated ‘reverberant’ data. Typically the data used in tests is created using the image method, proposed by Allen and Berkley [2]. This method evaluates the set of source-to-sensor reverberation paths through a simulated room by individually unwrapping each of the paths taken — as seen in Figure 2.6. The total distance travelled is used to deter-

mine the time delay, and by simulating the reflective properties of the ‘walls’, the attenuation.

However to build a model capturing the true complexity of a realistic room would require utilising knowledge of the absorption properties at different frequencies for every single object and alcove of a room. Also as a speaker moves around the room, it would be necessary to recompute the RIR at each time-step.

Combining these individual reflections gives an artificial estimation of the room impulse response which can then be convolved with the audio signals to simulate a recording. While such simulated recordings are indeed useful for testing algorithms in various experimental conditions (particularly with varied levels of background noise and reverberation), the models have not proven to be particularly accurate for estimating cross-signal correlation for recorded speech — particularly struggling to accurately represent the tail of the room impulse response which is often described as a diffuse reverberation tail.

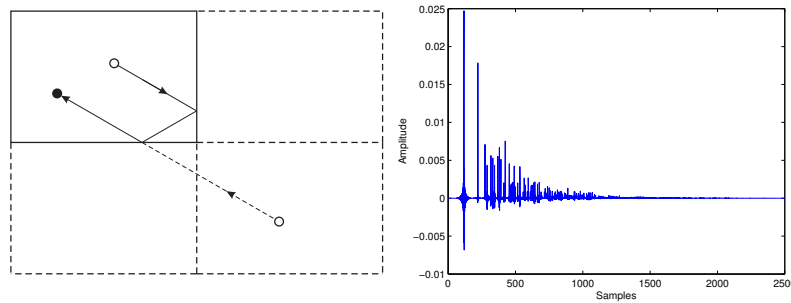


Fig. 2.6. *The image method. Left: the sound path unwrapped between a source (o) and a sensor (●) for two simulated wall reflections. Right: a room impulse response for a box room (with artificial RT_{60} of 0.3sec) generated using the image method.*

In general for this work, we have instead focused on experimental testing using real recorded speech. There does however exist a need for a standardised corpus of recorded test data — such as the NIST database for speech recognition — although it should be noted that recently some meeting data has become more available [16, 55]

2.6 Conclusions

In this chapter, we have introduced and discussed the problem of Acoustic Source Tracking. The problem seems, at the outset, to be quite straightforward, but as illustrated the complications caused by the environment and the source signal make finding an accurate solution for each instantaneous time-frame impossible.

Instead we will consider the problem from a probabilistic model-based point of view. The approach utilises the concept of recursive Bayesian estimation and more specifically Sequential Monte Carlo, which is introduced in the following chapter.

Sequential Monte Carlo Methods

In this chapter we will briefly introduce Bayesian statistics and some of the numerical estimation techniques commonly used in the field.

This field of probability treats the probability of a proposition to be the degree to which one believes that proposition to be true. Also known as subjective probability, Bayesian statistics developed from the eponymous theorem introduced by the English Presbyterian minister Reverend Thomas Bayes (1702-1761).

3.1 Bayes' Theorem

Presented in his posthumously published 1764 work *Essay Towards Solving a Problem in the Doctrine of Chances*, Bayes' theorem was stated as follows:

The probability of any event is the ratio between the value at which an expectation depending on the happening of the event ought to be computed, and the chance of the thing expected upon it's (sic) happening.

In contemporary usage, the theorem is typically presented mathematically as

$$P(X|Z) = \frac{P(Z|X)P(X)}{P(Z)} \quad (3.1)$$

where the terms are as follows

- $P(X|Z)$ — the conditional or posterior probability of X given Z
- $P(Z|X)$ — the conditional probability of Z given X

- $P(X)$ — the prior probability of X
- $P(Z)$ — the evidence or the prior probability of Z

The conditional probability $P(X|Z)$ can be stated orally as *the probability of an event X given that event Z has occurred*.

3.1.1 Bayesian Parameter Estimation

Using the basic principles of Bayesian theory it is possible to evaluate the degree of belief in a particular model or behaviour based on the data that is to hand. Typically one will be presented with a set of observations, $\mathbf{z}$, of a system, which are typically the result of an indirect measurement of a quantity (or quantities) of interest $\mathbf{x}$. This system is also likely to be subject to other influences that effect its behaviour — influences that are either unmeasurable or of no interest to us. We will denote such ‘nuisance parameters’ as θ .

Knowledge about how likely it would for a specific set of parameters to occur is represented as a prior distribution, $p(\mathbf{x}, \theta)$. From experience of how likely a particular set of variables would give rise to a set of measurements, we can form a likelihood function $p(\mathbf{z}|\mathbf{x}, \theta)$. Then using the aforementioned Bayes’ Theorem, it is possible to infer the joint posterior distribution of the parameter of interest along with the other unknowns as

$$p(\mathbf{x}, \theta|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x}, \theta)p(\mathbf{x}, \theta)}{p(\mathbf{z})} \quad (3.2)$$

However to determine the posterior probability of the parameter of interest alone would require one to integrate out the unknown nuisance parameters

$$p(\mathbf{x}|\mathbf{z}) = \int_{\theta} p(\mathbf{x}, \theta|\mathbf{z}) d\theta \quad (3.3)$$

in a process known as marginalisation. This is usually impossible as the integration is typically intractable analytically.

3.1.2 Model Selection

One problem in which such integration is not necessarily required is that of model selection — that is differentiating between a set of models to determine

the most apt model or to evaluate a suitable combination of the models given a set of data.

Consider a set of models, $\mathbf{M}_1, \dots, \mathbf{M}_K$. The posterior probability of model k would thus become

$$p(\mathbf{M}_k|\mathbf{z}) = \frac{p(\mathbf{M}_k)p(\mathbf{z}|\mathbf{M}_k)}{p(\mathbf{z})}. \quad (3.4)$$

To compare two different models, i and j , one could use a ratio of the relative probability of each model

$$\frac{p(\mathbf{M}_i|\mathbf{z})}{p(\mathbf{M}_j|\mathbf{z})} = \frac{p(\mathbf{M}_i)p(\mathbf{z}|\mathbf{M}_i)}{p(\mathbf{M}_j)p(\mathbf{z}|\mathbf{M}_j)} \quad (3.5)$$

such that the evidence term need not be evaluated. The ratio could then be used to determine which model is most suitable or to select a weighted average of the models to best fit the available data if none of the models is believed to be entirely true.

3.2 Monte Carlo Integration

Both of these problems illustrate that when the distribution in question is complex, multi-modal or high dimensional the integration of the marginal posterior distribution is either complex or requires a work around. However it is possible to instead approximate the integral using *Monte Carlo integration*, which is a numerical integration technique.

Monte Carlo integration uses a set of discrete Dirac delta masses, usual called samples, to approximate the distribution of interest. The Monte Carlo representation of the distribution, $p(\mathbf{x}|\mathbf{z})$, is presented as follows

$$P_{N_p}(d\mathbf{x}|\mathbf{z}) = \frac{1}{N_p} \sum_{p=1}^{N_p} \delta_{\mathbf{x}^{(p)}}(d\mathbf{x}) \quad (3.6)$$

where the number of samples used is N_p . Expectations with respect to the distribution can be estimated from this discrete distribution by numerical integration, in this case for the parameter I :

$$\begin{aligned} I(f) &= \mathbb{E}[f(\mathbf{x})] = \int f(\mathbf{x}) P_{N_p}(d\mathbf{x}|\mathbf{z}) \\ &\sim I_{N_p}(f) = \frac{1}{N_p} \sum_{p=1}^{N_p} f(\mathbf{x}^p). \end{aligned} \quad (3.7)$$

If the samples are drawn randomly this estimate will be unbiased. Furthermore from the strong law of large numbers this estimate converges (almost surely) to the correct estimate as the number of samples used increases

$$\lim_{N_p \rightarrow \infty} I_{N_p}(f) \rightarrow I(f) \quad (3.8)$$

The variance of the estimator is related to the number of samples used to estimate it

$$\text{var}(I_{N_p}(f)) = \frac{\sigma_f^2}{N_p} \quad (3.9)$$

The rate of convergence of such an estimate is thus independent of the dimension of the problem unlike deterministic numerical interaction techniques.

Finally note that Monte Carlo integration is only feasible when it is possible to sample from the distribution — a limitation which is discussed in Section 3.4.1.

Next we will consider a problem for which it is often necessary to resort to Monte Carlo integration — that of Recursive Bayesian Estimation.

3.3 Recursive Bayesian Estimation

Consider a system in which an unobserved signal, $\mathbf{x}_k$, will be assumed to have an initial distribution of $p(\mathbf{x}_0)$ and a transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. A set of observations related to the signal will become available, $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$, up to the time k . These observations are related to the signal according to the measurement distribution $p(\mathbf{z}_k|\mathbf{x}_k)$ which may be nonlinear and effected by non-Gaussian noise processes. We wish to form a probabilistic description of $\mathbf{x}_k$. The solution is to estimate recursively in time the posterior distribution $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$.

Using Bayes' theorem the posterior distribution can be restated as

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})p(\mathbf{x}_{0:k})}{\int p(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})p(\mathbf{x}_{0:k})d\mathbf{x}_{0:k}}. \quad (3.10)$$

Using a straightforward recursion, the related marginal posterior distribution, $p(\mathbf{x}_t|\mathbf{z}_{1:k})$, can be found using the following two step process. First the

prediction step, which uses the transition density to obtain the prior pdf at time k

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \quad (3.11)$$

which is known as the Chapman-Kolmogorov equation. Secondly when the measurement $\mathbf{z}_k$ becomes available the prior distribution will be updated via Bayes' Theorem

$$p(\mathbf{x} | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}. \quad (3.12)$$

This is hence the *update step*, where $p(\mathbf{z}_k | \mathbf{x}_k)$ is the likelihood function.

However the solution of this optimal recursion cannot be determined analytically for all systems. Solutions do exist for certain situations however, including the widely used Kalman filter.

3.3.1 Kalman Filtering

The Kalman filter, [50], is the optimal recursive Bayesian filter for linear systems observed in the presence of Gaussian noise. Drawing on the summary from [3], the two step Kalman filter recursion is briefly sketched as follows.

The behaviour of the system can be characterised as a linear system of equation

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \quad (3.13)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k \quad (3.14)$$

with known dynamical matrices $\mathbf{F}_k$ and $\mathbf{H}_k$. The system and observation noises will have covariances of $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ respectively although here we will make the simplification of assuming them to have zero mean.

Firstly the *prediction step* estimates the effect of the system dynamical model on the state mean and variance

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \quad (3.15)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1} \quad (3.16)$$

Then the *update step* incorporates the effect of the measurement residual to give us an updated posterior estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (3.17)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (3.18)$$

where the covariance of the innovation and the system gain are as follows

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (3.19)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}. \quad (3.20)$$

As all distributions are constrained to be Gaussian, once the conditional probability density function has been propagated, the optimal state estimate can easily be calculated. If the system model (the initial estimates of the values of the state, $\mathbf{x}_{0|0}$, and its covariance, $\mathbf{P}_{0|0}$) is correct then the expected value of the state will have zero mean error, $E[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = 0$ and will be the optimal solution.

However as our specific problem is nonlinear and the measurement noise of the localisation functions (as introduced in Chapter 2) are most certainly non-Gaussian and strongly time varying, the Kalman filter is not suitable for acoustic source tracking.

Many other problems cannot be solved within these limitations. Because of this, since the 1960's, a large body of research has outlined approximations to the Kalman filter. We will now refer to some of these approximation methods.

3.3.2 Extended Kalman Filtering

To accommodate non-linear state transition and observation models, the Extended Kalman Filter [48] implements a local linearisation of the models utilising a Taylor expansion. However when choosing to use the EKF one must decide to what order the Taylor expansion is carried out. Using a higher order expansion will be more accurate but will also add additional complexity. Using a lower order and perhaps modelling the process incorrectly may cause the filter to diverge. The EKF, unlike the Kalman filter, is not an optimal solution.

Furthermore the distributions are still confined to be Gaussian. For problems in which the system is multi-modal or highly skewed such a distribution

can never accurately represent the posterior. In particular the likelihood functions formed to represent the behaviour of the Generalised Cross Correlation in Chapter 4 and elsewhere are obviously non-Gaussian.

Despite these drawbacks the EKF has been widely used in some systems, including GPS and other navigation systems.

3.3.3 Grid-based Methods

Another approach has used a grid-based set of weighted samples to represent the state space. Grid-based methods break down the state space into a uniform grid of cells which are then weighted using a version of the Bayesian recursion procedure to estimate a non-Gaussian representation of the state.

For such a system to give an accurate representation of the underlying state space careful tuning of system parameters, as well as assumptions about the measurement process, is necessary. Furthermore many such approaches are often victims of increased computational cost when the dimensionality of the state space increases¹. Indeed some current acoustic source tracking solutions currently focus on finding two dimensional solutions, assuming that an extension to a full three dimensional solution can be trivially formed, which is not necessarily the case.

With the great increase in computational power in the 80's and 90's other numerical integration techniques which were, until then, infeasible have brought about rapid progress towards providing general solutions to the recursive estimation problem for non-linear and non-Gaussian cases. These Sequential Monte Carlo (SMC) methods are briefly introduced in the following section.

¹ Note that this work (Chapter 8) and other works [62, 63] have used grided measurement spaces as part of a particle filtering solution. This is distinctly different to the Grid-based methods described here.

3.4 Sequential Monte Carlo

As briefly discussed in Section 3.2, by using Monte Carlo integration with a large number of samples drawn from a distribution, such as the posterior, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, it is possible to numerically approximate integrations that are otherwise intractable analytically.

However, it is usually impossible to sample directly from the posterior distribution — possibly due to it being a multi-variate non-standard distribution. An indirect method known as *Importance Sampling* can be used.

3.4.1 Importance Sampling

Consider a novel distribution function $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. Following from Equation 3.7 we can recast the parameter estimation problem as follows

$$\begin{aligned} I(f) &= \int f(x)p(dx) \\ &= \int f(x)\frac{p(x)}{q(x)}q(x)dx \end{aligned} \quad (3.21)$$

If the support of $q(\cdot)$ includes the support of $p(\cdot)$ ($\text{supp}(q) \supseteq \text{supp}(p)$) then sampling from this distribution with an *importance weight* of

$$w(\mathbf{x}_{0:k}) = \frac{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \quad (3.22)$$

will be equivalent to sampling from the original distribution in the limit as N_p tends to infinity. Thus if we wish to estimate an expectation of the system such as $I(f)$ using N_p such samples this can be done using

$$I_{N_p}(f_k) = \sum_{p=1}^{N_p} \tilde{w}_k^{(p)} f_k(\mathbf{x}_{0:k}^{(p)}) \quad (3.23)$$

where $\tilde{w}_k^{(p)}$ is a normalised importance weight. Asymptotically this estimate tends to the true value, $I(f_k)$, as the number of samples tends to infinity.

Alternatively one could say that the posterior distribution has been approximated by a set of discrete samples

$$\hat{P}_{N_p}(d\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \sum_{p=1}^{N_p} \tilde{w}_k^{(p)} \delta_{x_{0:k}^{(p)}}(d\mathbf{x}_{0:k}) \quad (3.24)$$

and that parameter estimated in Equation 3.23 is simply the function $f_k(\mathbf{x}_{0:k})$ integrated with respect to it

$$I_{N_p}(f_k) = \int f_k(x_{0:k}) P_{N_p}(d\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) \quad (3.25)$$

However this system requires that the importance weights are recomputed anew each time new data becomes available. Hence it is required that all data is stored for all time, while also the computational complexity increases as time goes on.

3.4.2 Sequential Importance Sampling

To overcome these problems the method must be modified so as not to require previous data or estimates to update the current estimate of the posterior distribution $\hat{P}(d\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$. This can be done by updating the previous posterior distribution using a marginal distribution chosen to factorise as

$$q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) = q(\mathbf{x}_{0:k-1} | \mathbf{z}_{1:k-1}) q(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) \quad (3.26)$$

Following the derivation explained in Equations 45-47 of [3] the importance weights can be evaluated or rather updated using a recursive update equation

$$\tilde{w}_k^{(p)} \propto \tilde{w}_{k-1}^{(p)} \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(p)}) p(\mathbf{x}_k^{(p)} | \mathbf{x}_{k-1}^{(p)})}{q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(p)}, \mathbf{z}_k)} \quad (3.27)$$

A special case of this sequential importance sampler adopts the prior distribution as the importance sampling distribution. This common simplification reduces the weight update equation to $\tilde{w}_t^{(p)} \propto \tilde{w}_{k-1}^{(p)} p(z_k | y_k^{(p)})$. While this simplification is commonly used — including in some of what follows — it is not the most general case.

Finally note that weighted Sequential Monte Carlo samples are often referred to as *particles* and the field in general as *particle filtering*. Both terms will be used, often interchangeably, in what follows.

3.4.3 Bootstrap Filtering and Resampling

While the Sequential Importance Sampling approach relieves the algorithm from the problem of ever increasing computation another major issue is still to

be considered. As time increases these importance weights becomes more and more skewed as a small number of particles will contain most of the overall importance weight. Eventually after a few time steps a single particle will dominate with all other particle containing little or no weight. This is known as particle degeneracy and is unavoidable [32], consequently the algorithm fails to represent the posterior distribution adequately.

While a good choice for the importance density function will help to reduce the rate at which degeneracy occurs it is nessary to add a resampling step to the SIS algorithm. All such resulting algorithms are known as Sequential Importance Sampling with Resampling (SIR).

Resampling was first demonstrated by Gordon et al [44] and the procedure facilitated the first successful implementations of the particle filter. Resampling is, essentially, the elimination of particles with low weights and the multiplication of particles with larger weight to reduce particle impoverishment.

Systematic resampling has been implemented in this work. While other resampling strategies exist including stratified resampling and residual resampling, the choice of this method was not deemed crucial for our chosen application. Further discussion about resampling procedures can be found in [31].

When and how often resampling is carried out is also open to discussion. One method used in this thesis resamples only when a significant degeneracy is detected. While the details will be excluded here degeneracy can be estimated using the *effective sample size*,

$$N_{\text{eff}} = \frac{1}{\sum_{p=1}^{N_p} (w_k^{(p)})^2} \quad (3.28)$$

which effectively measures the proportion of the particle meaningfully contributing to the posterior distribution. Typically, should this parameter fall below some threshold, a new set of particles are drawn in accordance with the previous importance weights. A new set of uniform weights ($w_k^{(p)} = 1/N_p$) will then be assigned to each new particle for the next iteration.

Alternatively one could resample using a fixed timescale, however doing so may greatly reduce the diversity of the particle cluster. Note that both approaches are used in the following chapters where appropriate. Particle Filtering with an specified resampling step is illustrated in Figure 3.4.2.

3.4.4 Other Advances

The SIR bootstrap filter is the most basic implementation of the particle filter. Much research has gone into developing more complicated models which allow for performance improvements and the solution of problems that were until recently out of reach. A number of reviews of the various improvements have been produced, [3, 33, 40], and theoretical work continues on open problems. Some recent advances have been modified and implemented as part of this work. These methods are introduced and commented upon throughout the text.

Meanwhile another branch of the field continues to determine convergence limits and to prove the bounds of performance, [27]. While this thesis does not attempt to advance such theoretical research and is focused on the application of SMC methods, it is pertinent to recognise this work regardless.

3.5 Conclusions

Sequential Monte Carlo methods have the ability to solve highly complicated non-linear estimation problems, while also maintaining the flexibility to be applied to unrelated fields — without sacrificing generality of the underlying framework. With the vast increase in available computational power these methods have recently found applicability in a number of different fields.

In Chapter 4 we will outline how particle filtering has been applied to the problem at hand, acoustic source tracking, as well as identifying the limitations and constraints of current approaches.

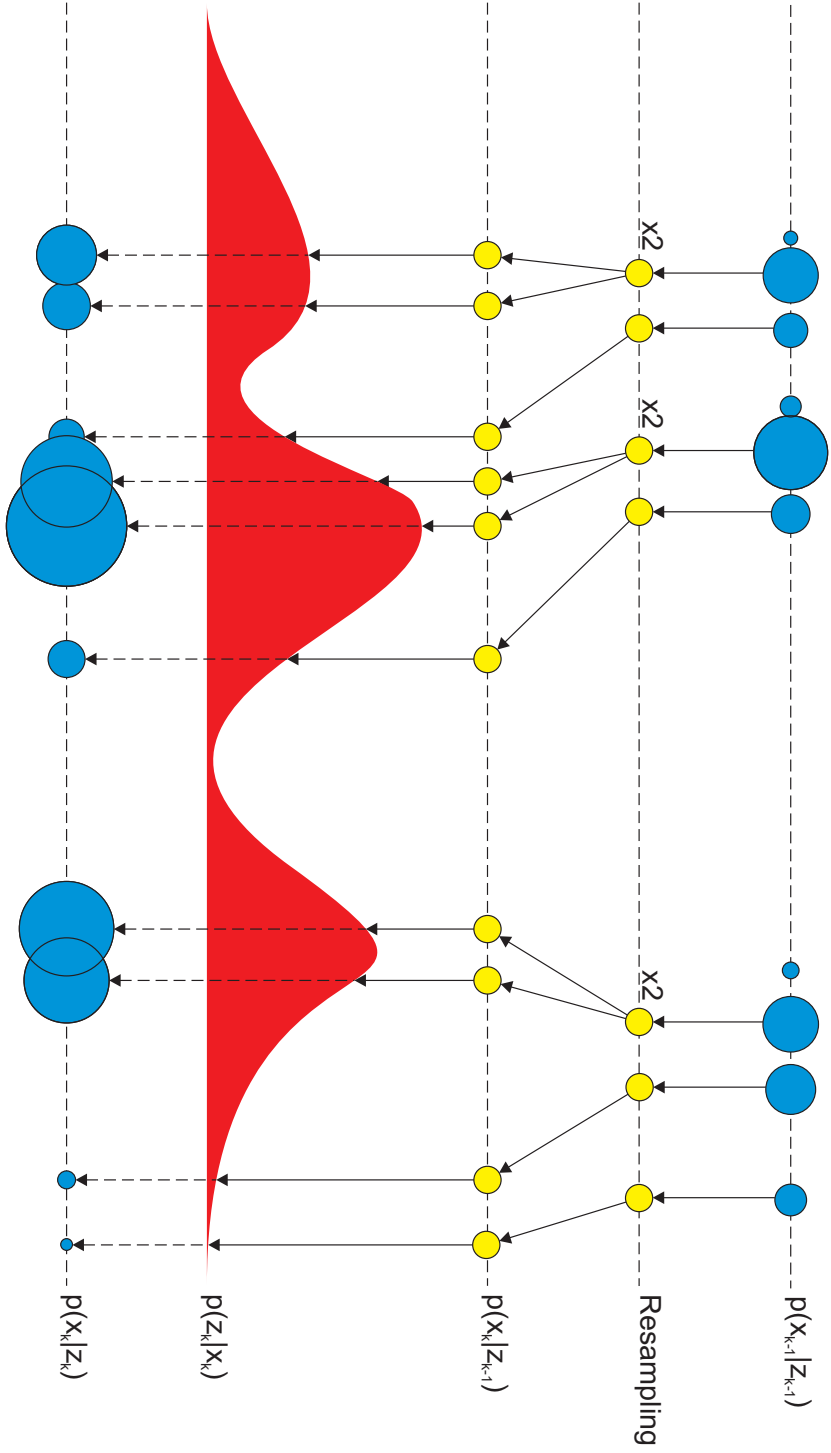


Fig. 3.1. An iteration of a 10 particle bootstrap filter. First the particle set representing the previous posterior distribution is resampled. The fittest particles are retained for the forthcoming iteration and the remainder are removed. Next the prediction step utilises the dynamical prior of the system while also adding random perturbation to gives us an estimate of the prior distribution $p(x_k|z_{k-1})$. Finally the likelihood and prior are combined using Bayes' Theorem to gives us a Monte Carlo estimate of the posterior distribution at time k .

Bayesian Acoustic Source Tracking

Bayesian Acoustic Source Tracking has recently attracted increasing attention as the approach discussed in Chapter 3 have become more widely disseminated, in parallel with the increase in available computational power. It is now possible to implement Bayesian AST algorithms in real-time using a typical PC and low cost microphone equipment.

However because AST is an applied problem whose performance and applicability varies depending on the number of sensors used, their layout, as well as the type of speech being tracked, so too have the layout and nuances of the solutions put forward. While the research communities of other signal processing tasks, such as blind source separation and speech recognition, have benefited from relatively limited problem definitions and organised performance evaluation tests to compare the relative merits of algorithms, the AST task is still characterised by different research groups working separately to solve stably different problems using their own equipment setups.

Bearing this in mind, this chapter attempts to provide a summary of the development of those Acoustic Source Tracking techniques that have used Bayesian methods to filter, identify and track targets. Section 4.2 introduces the problem framework which is largely common to the various applications. Section 4.3 details the various likelihood functions which utilise the measurement functions introduced in the previous chapter. Section 4.4 details various improvements that have drawn on the characteristics of how people speak

and move as well as the physical characteristics of the spoken word to improve tracking.

Finally, in Section 4.5, a number of advanced applications — such as the fusion of audio and video cues for joint tracking and the extension of algorithms to track more than one audio source — are discussed.

4.1 Use of Kalman Filtering

Initial model-based Bayesian AST methods, during the 1990's, used Kalman filtering. As Kalman filtering does not suit the problem at hand a number of modifications (mainly concerning movement models) were necessary.

A mechanism for switching between a stationary source model and moving source models was proposed for this application in a early paper by Sturim et al. [86]. The so-called Interacting Multiple Model (IMM) system maintained two or more Kalman filters running in parallel for each source and used a switching mechanism which attempted to match the most suitable movement model to the observations. While target initialisation and removal was carried out in quite an ad hoc manner, the framework suggested did keep track of activity in the different regions of the tracking space.

Unfortunately these Kalman filtering solutions have generally been unsuccessful [83]. While the multiple Kalman filter approach has been reused by some more recent work, [68, 76], albeit with a different localisation functions and hybrid microphone arrays, it has been mostly recognised that such an approach is essentially a halfway house between Kalman Filtering and particle filtering.

4.2 General Framework

A number of particle filter solutions have been proposed for AST recently. These strategies have typically built upon a basic particle filter framework. The framework will be introduced in this section and retained (with modifications) throughout the remaining text.

4.2.1 State Vector

Firstly a state vector will be introduced to represent the behaviour of the source(s) we wish to track. The state vector will be proposed as follows

$$\alpha_k \triangleq (x_k, \dot{x}_k, y_k, \dot{y}_k) \quad (4.1)$$

where, for example, $(x_k, \dot{x}_k)$ are the source position and velocity terms, in the $\mathcal{X}$ -dimension at time k . Estimation of this state vector for all time is essentially the aim of all algorithms in this field. Other variables and indicators such as for acceleration, orientation, activity and stationary/moving monitoring can be added to this basic vector. Multiple sources can also be represented by concatenating two or more such vectors to create a larger overall state vector.

Our understanding of the source behaviour is represented by a discrete probability distribution, using a set of particles. The p th particle, $\alpha_k^{(p)}$, will be paired with an importance weight, $w^{(p)}$ giving the pairing $\{\alpha_k^{(p)}, w_k^{(p)}\}$. Note that the particle number will often be dropped for clarity in this text.

4.2.2 State Dynamical Model

Source movement in the $\mathcal{X}$ and $\mathcal{Y}$ dimensions will be assumed to be independent and can be decoupled as a result. State dynamics will be modelled by a first-order Langevin Markov process whose specifics were first proposed by Vermaak [90] and retained by Ward et al. [93]. The model will be specified by its initial state and state transition distributions which are of the form $p(\alpha_0)$ and $p(\alpha_k|\alpha_{k-1})$ respectively. The discrete time equations for the $\mathcal{X}$ dimension of the source state will be

$$\begin{aligned} \dot{x}_k &= a_x \dot{x}_{k-1} + b_x F_x \\ x_k &= x_{k-1} + \Delta T \dot{x}_k \\ a_x &= e^{-\beta_x \Delta T} \\ b_x &= v_x \sqrt{1 - a_x^2} \end{aligned} \quad (4.2)$$

where $F_x = \mathcal{N}(0, 1)$. A suitable choice of parameter values for β_x and v_x will allow us to simulate realistic human motion. These transition equations can also be represented in matrix form as

$$\alpha_k = \begin{bmatrix} 1 & 0 & a_x \Delta T & 0 \\ 0 & 1 & 0 & a_y \Delta T \\ 0 & 0 & a_x & 0 \\ 0 & 0 & 0 & a_y \end{bmatrix} \cdot \alpha_{k-1} + e_k \quad (4.3)$$

where the noise variable is as follows

$$e_k \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b_x^2 \Delta T^2 & 0 & 0 & 0 \\ 0 & b_y^2 \Delta T^2 & 0 & 0 \\ 0 & 0 & b_x^2 & 0 \\ 0 & 0 & 0 & b_y^2 \end{bmatrix} \right) \quad (4.4)$$

While publications discussed in Section 4.4.1 have proposed alternative dynamical models, the choice of model is not thought to be a key algorithm factor. Next we will focus on the various measurement model and equipment layout combinations which have been proposed.

4.3 Measurement Models

The microphone equipment layout chosen for a specific application usually defines the measurement function used to generate the particle likelihoods and hence the algorithm structure. The following is a list of typical systems that have been implemented:

- Constrained arrays such as circular array (with a diameter in the order 15cm and placed at the centre of a meeting table) or linear arrays (typically installed onto/into a computer monitor) to estimate location in one angular dimension [41, 68, 91]. These setups typically use TDOA estimators — most commonly the GCC.
- Closely spaced microphone pairs (or simulated pairs) scattered throughout the room [64, 74, 90, 92]. These setups typically use TDOA estimators.
- Dispersed arrays with a set of individual microphones scattered throughout a room [42, 63, 86]. Typically the microphones are scattered around the outside of the room — envisaging that eventually they would be installed in the walls. This also helps to avoid front/back ambiguity. The number

of individual sensors using in this setup has varied from as few as 6 to approximately sixty. The Steered Beamformer is typically preferred for this setup when using a model-based approach, although earlier non-model methods often used pairwise GCC functions because of the computation requirements of searching the SBF surface.

- Combinations of the above have been experimented with. One such setup used a mobile robot equipped with a circular array moving in a room containing banks of microphones installed on the rooms walls [70].

Setups utilising joint video and audio tracking have also been proposed and are discussed in Section 4.5.1.

4.3.1 GCC-based tracking

The first particle filter-based AST algorithm used the GCC as the localisation function [89] although it had been used previously in a Kalman filter solution [86]. Typically, having isolated a useful set of GCC function peaks, a likelihood function is evaluated for each microphone pair across the relative delay space for that pair. A normal distribution combined with a noise clutter floor was used by [90]. For example (for microphone pair r)

$$p(T_r|\alpha_k) = q_0 + \sum_{t=1}^{N_t} q_t c_\alpha \mathcal{N}(\tau_t; \tau_\alpha, \sigma_g^2) \quad (4.5)$$

where c_α is a truncation constant, τ_α is the delay corresponding to the state vector α_k and $T_r \triangleq (\tau_1, \dots, \tau_{N_t})$ is the set of useful GCC peak delays for that pair. q_0 is a constant which reflects the proportion of clutter measurements while the set of prior belief probabilities, numbered q_t , will typically be set equal to one another such that $q_0 + \sum_{t=1}^{N_t} q_t = 1$.

Then the final particle likelihood will be the product over all pairs as follows

$$p(\mathbf{T}|\alpha_k) = \prod_{r=1}^{N_r} p(T_r|\alpha_k) \quad (4.6)$$

Because the solution is model-based, rather than data-based, the computationally intensive triangulation step is neatly avoided — a feature of all subsequent methods. A similar method is used in a multi-target framework [64] and for joint audio-video tracking [41].

Related algorithms have attempted to use the magnitude of the GCC function peak to provide information, a so called *pseudo likelihood* [93]. However the relative magnitude of the GCC is a poor indication of source position likelihood.

Finally, the feasibility of using the GCC for tracking of more than one source is discussed in Section 7.6.

4.3.2 SBF based tracking

As mentioned in Section 2.3.2 implementing the SBF is not as straightforward as the GCC. Nonetheless having implemented the function, the SBF provides a single measurement at the location of interest for the correlation across all the recorded signals. However a source of complication is that peaks in the SBF are hidden unless the entire function's continuous surface is evaluated — which is a computationally intensive task.

A number of algorithms have attempted to bypass this problem by using the magnitude of the SBF function at the particle locations as a likelihood measure with a bootstrap particle filter. This was dubbed the *pseudo likelihood* by [93] and algorithms using the function have been seen to operate admirably when compared with GCC-based methods [57]. However as noted therein this density cannot be properly normalised while the distribution of SBF magnitudes was not studied. Use of the peak magnitude is likely to be useful only as a simple active/inactive designation.

An interesting hybrid system consisting of a distributed array of sixty sensors combined with an eight channel circular array placed at the centre of a room was proposed in [70, 71]. The authors combine a SBF-based likelihood function drawing on audio recorded by the distributed array and another using audio from the circular array which utilises the GCC function. However the authors chose to combine the final likelihood measurement together without discussing the relative importance of each component.

Importance Sampling Methods

The bootstrap particle filter is, of course, the most crude and simplistic of particle filtering strategies. The algorithm is essentially that which was initially proposed by [44]. The proposal strategy is simply the original particle set from the previous time step propagated using a model of the particle dynamics. Importance Sampling, as discussed in Section 3.4.1, is a more advanced proposal strategy and would be expected to improve performance by proposing particles in more apt areas of interest. However, the question of how one might design an importance sampling function for AST is not straightforward.

Lehmann and Williamson, [62, 63], proposed an importance sampling strategy in which the proposal method uses a version of the SBF integration over a small set of low frequencies — the authors chose a range of 100Hz–400Hz — to propose new particle positions. The SBF function evaluated over this frequency range will then give rise to broader SBF peaks and hence the density of measurements need not be as high to capture all peaks. The authors used a grid resolution of 0.1m, which was deemed sufficient to observe all of the functions peaks. Because this low density function is evaluated in its entirety it can then normalised and used as a proper sampling distribution.

The hybrid approach which follows draws heavily on I-Condensation [47]. A proportion of particles were proposed using this proposal mechanism. Other particles were proposed using a reinitialisation scheme, to promote diversity across the surveillance region, while the remainder (and typically the majority) were proposed using the usual bootstrap model from the previous particle set. Having determined the particle set, the importance density of each particle is then evaluated and an overall posterior distribution for the particle set can be determined.

However it must be noted that this importance sampling algorithm required fine tuning of a number of different variables — including the proportion of particles proposed by each method (which also depends on the SBF surface shape) and a number of variables within the importance sampling transition stage. How successful this strategy would be within complex and challenging recording environments remains to be seen.

Furthermore there appears to be no obvious way to extend this method to multi-target tracking as there isn't a measurement set which can be assigned to either source (as the final likelihood function is only evaluated at the actual particle locations).

Despite these problems, the authors made some important observations regarding the choice of SBF frequency range. They showed that by limiting the frequency range to a smaller band of low frequencies computation of the SBF function was vastly reduced. This property of the SBF is studied in Section 5.2.2 and novel algorithms drawing on the results of this analysis are proposed in Chapters 7 and 8.

4.3.3 Other Measurement Functions using within the Literature

In Section 2.4 a localisation method using a *spatio covariance matrix* for localisation is detailed. Checka et al [18] go on to use the estimates generated by the measure with the additional aid of two cameras to track three sources using a particle filter. While the localisation method is indeed novel, examples illustrated by the authors suggest that the video tracking portion of the algorithm is highly dominant while the sound information is mainly used to determine which of the present speakers is actually active during a conversation.

Finally a related phase unwrapping method, also discussed in Section 2.4, has been used to track two speakers in [76]. The performance of the technique, which used a Kalman filter, is difficult to benchmark because results were tested using simulated recordings.

4.4 Behavioural Advances

Of the two main design decisions for a particle filter, the most important is the design of the likelihood function and how it represents our belief in the measurements. This portion of the algorithm has been discussed widely above.

The second design decision — the choice of dynamical model — as well as other behavioural-based innovations will be discussed in this section.

4.4.1 Alternative Dynamical Models

The Langevin dynamical model first used in this field by [90] is the commonly used source dynamical model. While a mild modification of the model introduced additional acceleration components [89], typically researchers in the field have assumed that the choice of model is not crucial. If the number of particles in the system and the constants in Equation 4.2 are chosen so as to spread a particle cloud over the entire region in which the source could plausibly be located, the only benefit garnered would be a mildly lower average location error or an ability to operate the filter with a reduced number of particles.

However recently the first attempt to examine a full range of dynamical models for acoustic source tracking has been published [59, 61]. The authors broke the set of models into two main types:

Coordinate Uncoupled Models:

All parameters in the $\mathcal{X}$ and $\mathcal{Y}$ dimensions are treated separately and each has an associated velocity component. This is in Equation 4.1. Four Cartesian dynamical models were discussed:

- A random walk process for each of the velocity components
- A random walk process with an addition acceleration components for each of the velocities
- A time-correlated process in which the correlation between the velocity terms and subsequent values falls when the time-step between iterations increases
- The previous Langevin process

Curvilinear Models:

The parameters for the velocity terms are represented via a polar coordinate system. This model was also used by [71]. In its simplest form the state vector is as follows

$$\alpha_k = [x_k, y_k, v_k, \theta_k] \quad (4.7)$$

where v_k represents the source velocity magnitude and θ_k the orientation angle or direction of movement. The decomposition of the velocity components is illustrated in Figure 4.1. The source position at each time k is updated indirectly as follows

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \Delta T \cdot v_k \cdot \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix} \quad (4.8)$$

where the current values of v_k and θ_k are found using one of the models described below. Note that state position novelty is only introduced indirectly via the velocity component.

In total five curvilinear-based dynamical models were proposed including

- A random walk process for each of the orientation and velocity components
- A random walk process with an addition acceleration component for the velocity
- A time-correlated process as above

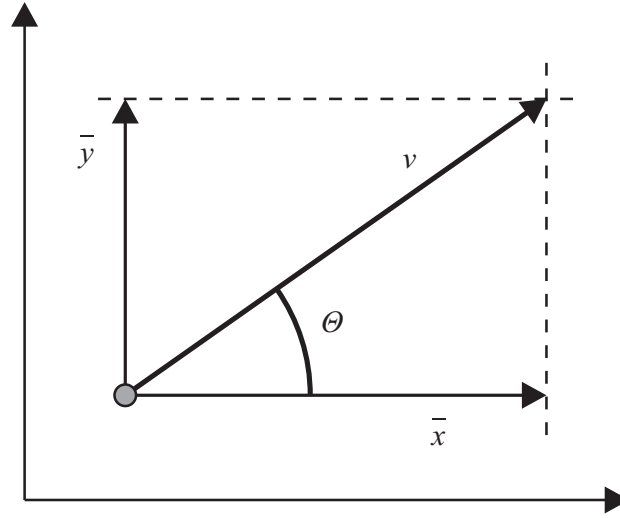


Fig. 4.1. *Illustration of velocity variables used in the coordinate systems in Section 4.4.1*

In testing the various algorithms the authors proved that both tracking systems were suitable for acoustic source tracking. The algorithms were tested

against a piece of audio in which the source moved in a linear path, alternating between silence and activity. As one would expect the models tuned to tracking such drifting targets performed best. The tests illustrated do not suggest however that these models would provide any significant performance improvement in general.

Also using a model over-tuned to a particular audio sample could possibly lead to lower stability in general. The Langevin model remains the preferred model for generic movement.

Finally, it should be noted that in Chapter 6 a source orientation estimation algorithm is proposed. While the algorithm utilises the uncoupled dynamical system for location dynamics, a parameter for orientation, similar to the proxy parameters discussed above, is also added to the state vector. This parameter is independent of the Cartesian coordinate system and has its own unrelated dynamical model.

4.4.2 Voice Activity Detection

Rudimentary Voice Activity Detection (VAD) was used in some of the earliest research in Bayesian AST field [86]. Typically these ad-hoc algorithms were of the form “*If there is some activity in a certain region for X of the last Y frames, then initialise a tracking algorithm in that location.*” While algorithms for introducing and removing source tracks are discussed in the subsequent section, in this section we will look at how the activity of sources should effect the source behaviour.

Activity can be measured either indirectly¹, that is from a measurement other than those used in the localisation algorithm, or directly from the measurement function itself. Indirect VAD algorithms measure activity from the incoming speech signal and could allow for the application of a generic VAD to this field.

¹ It is a moot point as to whether voice activity detected from the actual speech signal should be labelled direct or indirect activity detection, as we are defining our system from the viewpoint of tracking rather than speech analysis.

A study of indirect VAD's for AST is introduced in [60]. Having determined when a source is inactive they proposed that active tracking should cease and instead the particle cluster should drift according to the source dynamical model alone — a quite logical approach.

The method initially proposed [49] summed the energy level across a set of frequency bands before making a binary decision. In frames which are deemed silent all particles were assigned equal weight — essentially drifting freely. Subsequently this decision was softened by comparing the energy level to the background noise to determine a signal to noise ratio. Using a sliding scale, the particle likelihoods are then found by a combination of a SBF-based likelihood function and a uniform distribution.

The authors illustrated examples of a source alternating between activity and inactivity, showing how instability between speech events can be avoided.

Alternatively, other target activity detection methods measure activity from the localisation function, such the SBF surface. Because these methods use the same modality that is used in the tracking algorithm, we are guaranteed correlation between the set of frames deemed active and the frames from which we will aim to garner tracking information. This is not the case with an indirect VAD method. In addition this tracking method should have the ability to determine the activity of more than one simultaneously active source and possibly the approximate location of any activity.

Gatica-Perez et al. [43] proposes a method which evaluates the set of SBF peaks in all frames over the duration of a typical phoneme and clusters nearby peaks. All the frames belonging to significant clusters of peaks are defined as speech, otherwise frames are labelled as non-speech. However this algorithm does not take a probabilistic viewpoint, making a hard distinction between activity and inactivity for the purpose of target initialisation/removal only.

This approach is discussed further in Chapter 8 in which a multiple target activity detection algorithm is proposed.

4.4.3 Target Track Initiation and Removal

As previously mentioned, a number of the previous algorithms in this field assume knowledge of the initial source position(s) and also require that activity of the source(s) be maintained from start to finish of the tracking segment [76, 90, 91, 93]. Generally this assumption was made to implement and illustrate novel tracking algorithms with the acknowledgement that later a module for the initiation and removal of targets would be required to make an implementation fully feasible.

Other approaches tracking non-overlapping speakers in a conversation [60, 91] have essentially been single target trackers with modules to allow switching between source according to activity. This so-called re-initialisation step has the ability to quickly switch between different areas of the surveillance region but does so without any acknowledgement of the existence of more than one target in the environment or any attempt to determine which of the targets is currently speaking.

As was mentioned previously Sturim et al. [86] first introduced explicit birth/death modules based on regional activity. While the algorithm took an ad-hoc approach to the decision making, subsequent proposals have followed the general idea of monitoring the surveillance region and proposing new particles or even entirely new particle filters when deemed necessary.

Another proposal, [43], also determines the number of targets and any newly active targets at the outset of the iteration. The authors argue that unless there is significant ambiguity as to the number of targets present then it is best to determine it at the outset and that otherwise the particle set will evolve to contain particles with a varying number of targets and that particles with the ‘wrong’ number of targets will effectively be wasted. Instead a set of likely face-type shapes are probed. (The algorithm is an audio-video fusion). The shapes are ranked according to how likely they are to represent a speaker and the best location configuration is used in the prior of the filter. The audio data plays no part in this proposal mechanism.

Alternatively one could incorporate the number of targets into the probabilistic framework. Pham et al. [74] evaluates a spontaneous birth intensity

grid across the surveillance region. Summing the values for each cell gives an instantaneous estimate of whether or not a new target should be added. This is then used to update the probability distribution for the number of active sources. When new particles are proposed they will be added with a number of targets according to this distribution. This leads to the situation whereby the particle set will contain differing numbers of targets, although the most likely state is expected to predominate and to match the correct number of active targets. In Section 4.5.2 multi-target tracking with this approach is further discussed.

4.5 Related Tracking techniques

4.5.1 Audio Video Fusion

As mentioned previously Audio-Video conferencing for remote collaboration is an envisaged scenario in which acoustic source tracking could be used. A number of approaches have fused the two measurement modalities to complement their respective strengths. While joint tracking is not within the remit of this thesis, we will briefly mention current research.

Face tracking algorithms can keep track of a number of people moving in a image, but struggle to determine which of the people to focus on at any one time. Lip tracking, for example, is a weak measure of a particular speaker's activity — becoming confused by such things as participants chewing gum or covering their mouths.

Alternatively audio tracking can be used to accurately track active speakers but, of course, cannot track when the source is silent. Should the source move from one location to another before continuing to speak segment-to-source assignment will be lost (unless diarisation techniques can be used to match audio segments together).

The simplest combined layout, consisting of two microphones and a single camera mounted around a computer monitor, was proposed by [91]. The system was used to track sources in the two dimensions of the camera plane with the microphone pair provides delay estimates for the horizontal dimension.

The camera, via a face outline model, provides estimates in both the horizontal and vertical dimension. The likelihood function proposed was simply a product of each of the individual likelihood functions. This simple method was tested with two speakers taking part in a conversation while sitting in front of the equipment. While results showed the tracker alternating between the two speakers, it is apparent that the audio measurements are mainly used for re-initialisation of the filter when the speakers alternate and not central to the tracking process itself.

Gatica-Perez et al. [41] proposed a very similar tracking system (with desktop circular 8-microphone array replacing the two element array) to track in the same dimensions. The contribution of the microphone array was limited to accurate azimuthal estimates as the elevation estimates were of high variance.

The authors then went on to greatly expand their system, [43], to analyse speaker activity during a four person conversational meeting. This system was made up of three cameras as well as the circular microphone array and provided estimates in all three dimensions. The addition of four people as well as three cameras causes a large increase in the dimensionality of the state space which would require a vast increase in the number of samples to adequately sample regions of high likelihood. Instead the authors utilise a hybrid Markov Chain Monte Carlo Particle Filter (MCMC-PF) which uses a burn-in phase at each iteration to detect regions of high likelihood.

Finally an unrelated system proposed by Nakadai et al, [51, 71], develops a speaker-facing mechanism using a camera and a pair of microphones. The entire sensor system is mounted on a rotating plate which can be commanded to turn — like the turning of a human head — to face an object of interest. This non-Bayesian system determines the direction in which the robot should turn to using an Expectation-Maximisation algorithm using face detection and two-microphone TDOA.

4.5.2 Multi-target Acoustic Source Tracking

Multi-target tracking of acoustic sources is an emerging part of the AST field. Two particular applications have been considered in the literature, the majority of which has focused on a conversational situation with 2–3 speakers.

Audio-only applications have typically made the implicit assumption that no two speakers may be active simultaneously and speaker activity is strictly non-overlapping [63, 91]. The resultant algorithms make not attempt to recognise or acknowledge the existence of more than one speaker — essentially tracking whatever source is active regardless of speaker identity. While audio-video fusion approaches have been able to relax that limitation somewhat, as mentioned above applications have typically been dominated by the superior accuracy and stability of the video-based measurements [41].

Alternatively, simultaneously active speakers within a cocktail party-type environment has also been considered. Initial attempts have typically assumed that each of the sources be active from start to finish, such as the Kalman-filter based approaches by Potamitis et al [76] and Murase et al [68]. The authors assigned peak measurements to sources based on proximity and assumed sources were well separated. Thereafter the targets were dealt with independent of one another using two separate filtering systems. The latter system was later extended by the implementation of a particle filter to track two sources, [70, 71], but again assumes constantly active sources. This application used a 72 channel recording system, modelled around Honda’s ASIMO humanoid robot.

However the first attempts to track a time-varying number of targets probabilistically has appeared much more recently. Ma et al [64, 92] introduced an application of random finite set (RFS) theory while a related work has also been published by Pham et al. [74]. The resultant RFS particle filter was shown to track two simulated (omnidirectional) moving speakers using a TDOA-based measurement function. However as the authors recognise the set of TDOA measurements “*deteriorate due to mutual interference between the two signals.*” How robust this method would be in the face of speakers crossing in front of microphone pairs and realistic speaker models remains

to be seen. Further discussion about multi-target tracking can be found in Chapter 8, where an alternative algorithm is proposed.

4.6 Conclusions

Attempts to use particle filters to track moving speakers have been discussed in this chapter. Previous work within the field has progressed considerably from the initial paper by Vermaak and Blake, [90], but still there exists a number of problems and limitations with these tracking systems. How best to solve these challenges is open to discussion, but drawing observations from realistic recorded data should at least provide an indication of the direction in which research should move.

In the following chapter a number of experiments are carried out to quantify the many parameters and behavioural trends, which have thus far only been empirically observed. Drawing from these observations a number of contributions are put forward in subsequent chapters.

Part II

Contributions

Experimental Analysis of Acoustic Source Tracking

To further understand the Acoustic Source Tracking problem, it is important to first experimentally determine the typical behaviour that is expected for sources speaking and moving in the proposed environment.

For this reason a series of experiments were carried out using different combinations of sensors, sensor geometries and sources positions (stationary and moving) in the proposed recording environment. While what follows may stand alone as an experimental study, a number of the results presented in this chapter are also utilised in subsequent chapters.

In Section 5.1 we present the results of a number of experiments with the Generalised Cross-Correlation. We experimented with the following system parameters:

- Speaker directionality and the relative angle between the microphone and the source.
- The distance between the source and the sensors.
- The distance separating the sensor pair.
- Interference effects between two simultaneously active sources.

The conclusions drawn from these experiments motivate our interest in using the Steered Beamformer as the localising measurement model instead. We go onto study the SBF in Section 5.2. Again a series of parameters were studied:

- The characteristic behaviour of the distribution of the SRP values.
- Frequency range of integration.
- Density of the SBF grid.

- The effect of source movement.
- Ambient noise level.
- Interference effects between two simultaneously active sources.

5.1 Experiments with the Generalised Cross-Correlation

First we will carry out experiments to identify the performance limitations of the PHAT-GCC measurement function. These experiments will illustrate some oft ignored physical complications as well as motivating our decision to use the steered beamformer as our main localisation measurement function in Chapters 7 and 8.

5.1.1 Speech Directionality and Microphone-to-Source Relative Angle

Speech Directionality is the effect of non-uniform radiation of sound from the mouth and its absorption by the head and torso. At its most simple one notices its effect in a car — speakers sitting in the front seats of the car are often unintelligible to those sitting in the rear.

Previous Experimentation

Directionality was experimentally studied as early as 1938 by Dunn and Farnsworth [34]. The authors observed that for human speakers there exists a 15dB attenuation at high frequencies for microphone recordings taken behind the speaker when compared to recordings carried out by sensors placed at the front of the speaker. Their published results are presented graphically in Figure 5.1.

Further Experimentation

Within the field of coherent signal processing and using the far-field assumption, it is typically assumed that sensor arrays are clustered close to one another relative to the distance to the signal source. This means that the source

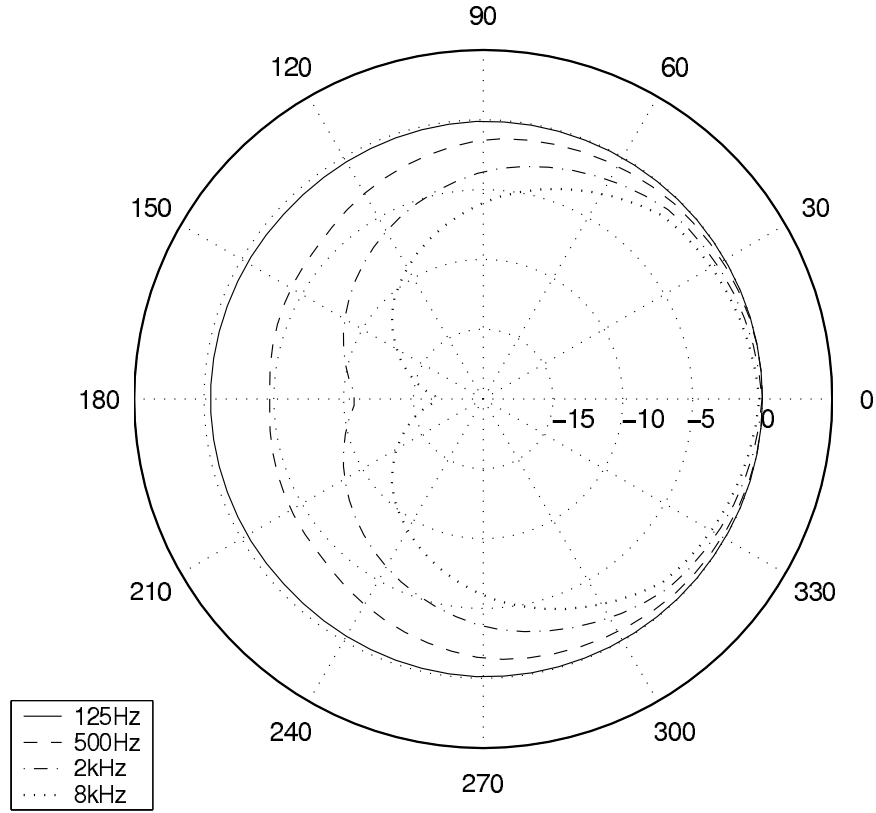


Fig. 5.1. Human speaker directional response (dB) obtained by least squares fitting of data from [34]. Graph taken from [6] with permission.

aperture angle is very small and that the directionality of the source(s) can be assumed to be negligible. Many audio signal processing algorithms make these assumption while enforcing strict microphone layout limitations [78–80].

However if one instead considers a sparsely distributed sensor network, the effect of directionality becomes very relevant indeed. Many of the recording setups described in the literature fall into this category [63, 93, for example], often surrounding the source with sensors. This effect causes reduced correlation between signals recorded at sensors positioned at different angles relative to the mouth.

There are two reasons for this. First, the attenuation of sound varies greatly with frequency because of the non-uniform absorption of the sound by the source’s head. For example, a 600Hz speech signal has a wavelength of 0.57m,

while speech at 3000Hz has a wavelength of 0.11m — which is less than the dimensions of the human head. This results in higher frequencies being attenuated more than lower frequencies.

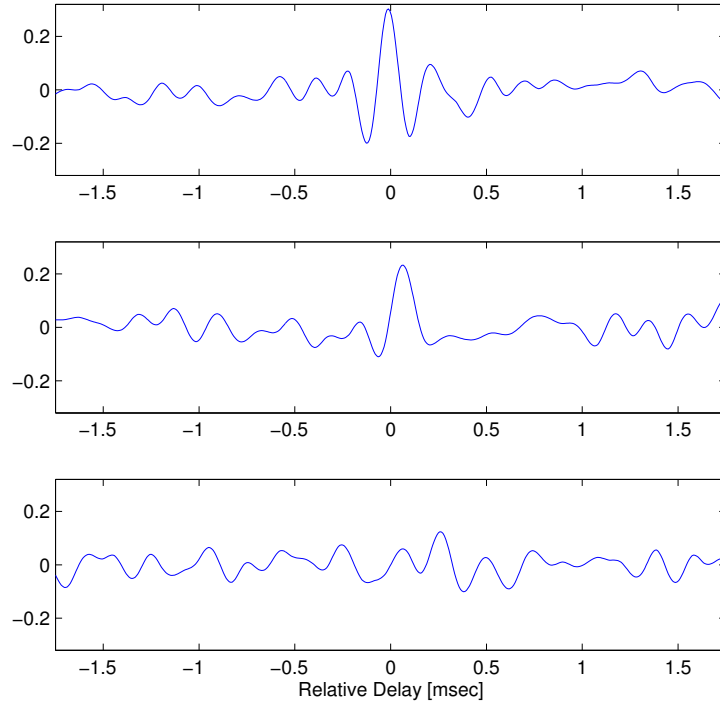


Fig. 5.2. *Effect of speaker directionality on signal-to-signal correlation: GCC functions formed by recording the same source signal at microphone pairs situated at 0° (top), 90° (middle) and 180° (bottom) relative to source orientation. In each case, the true source delay was zero, the x-axis is measures the relative delay between the recorded signals (in milliseconds), while the y-axis measures is the cross-correlation magnitude at that delay.*

Second, as the relative angle between source and sensor increases, the signal energy of the direct path portion of the signal falls when compared to that of the multi-path portion (caused by reflections). This has the effect of increasing the relative amount of reverberation of the signal recorded by the sensor.

While it is very difficult to fully quantify the effect of speech directionality and absorption because of these numerous external factors, the effect that it has on signal correlation is illustrated in Figure 5.2 for three different recordings. See how the clarity of the GCC function peak is hampered by the increased relative angle between the sensor pair and the speech source.

We now want to illustrate and quantify the effect that increasing the relative angle between the source orientation and the microphone pair has on the recorded signal cross-correlation in comparison with that of a microphone pair located behind the speaker. In other words, a microphone pair positioned in front of a speaker's head will record the source signal with greater signal correlation.

Figure 5.3 presents the results of a series of recordings of white noise. Each radial segment of the plot represents the distribution of the maximum value of the GCC function formed by recordings made at microphones placed at a particular relative orientation (which is represented by the orientation of the segment). Three recordings (each of 40 seconds duration) were carried out at 16 increments of 22.5 degrees, while the distance between the source and sensor pair was 1.5m, which is relatively small. This resulted in approximately 3500 data samples per distribution.

The segment directed to the right represents the distribution for zero relative angle — that is that the speaker is facing the microphone pair directly, while the segment directed to the left represents the measured distribution when the speaker is facing directly away from the microphone pair.

As would be expected the figure is symmetrical about zero degrees, although in practice this functional behaviour could be effected by the layout of reflecting surfaces or noise sources within the room. For each incremental increase in the relative angle, the distribution of GCC peaks has a lower mean. This behaviour is quite stable and is used as the basis of a speaker orientation estimator in Chapter 6. However beyond a relative orientation of 90 degrees it can be seen that the mean of the distribution of peaks converges. There is little correlation between recorded signals when the source is now facing away from the sensors.

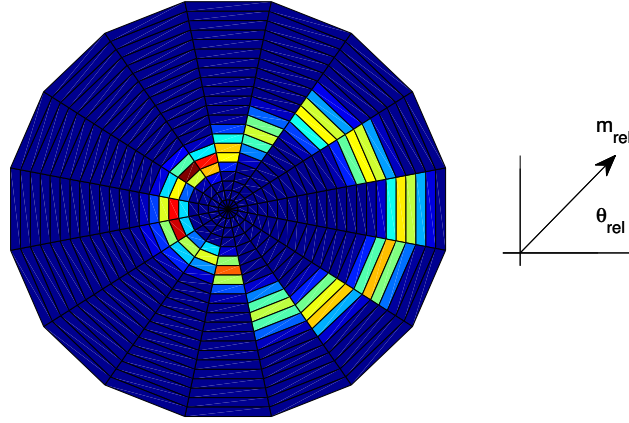


Fig. 5.3. Plot of the distribution of GCC peak magnitudes, m_{rel} , versus relative angle between source heading and microphone pair, θ_{rel} , illustrating that there is greater signal correlation when the relative angle is small. The source signal was white noise and the environment was a typical office room. See Section 5.1.1 for more details

However beyond estimating speaker orientation, this parameter is a substantial complicating factor for source tracking algorithms. Microphone arrays must be distributed around the room to provide reasonable coverage of this non-uniformly emitting source or else sacrifice tracking performance for certain source positions and orientations. Furthermore, because a source may be observable in the cross-correlation function of a particular microphone pair but not in another, this leads to measurement association complications where there is more than one single active source. This specific problem is further discussed in Section 7.6.

Finally, it must be pointed out that this effect has been either ignored or not recognised explicitly in the majority of previous research¹ within this field — despite the awkward limitations it places on both the data capture and signal processing portions of any AST solution.

¹ One exception is an implementation of Acoustic Beamforming by Betlehem and Williamson [6].

5.1.2 Distance from Source to Microphone Pair

The second parameter we will consider is the distance between the source and the microphone pair for a fixed sensor separation. To do this the sensors were positioned 0.6m apart and a sample of speech was recorded while varying the source-to-sensor distance. The source faced the microphones in each recording. The experimental results are presented in Figure 5.4.

The results show, as one would expect, that as the separation is increased the level of correlation between the recorded signals falls (and does so approximately linearly). This rate of decline is quite gradual and at a distance of 4m there still exists a substantial peak (relative to the RMS GCC function values). However these recordings were carried out in unchallenging recording conditions with low noise and reverberation levels. Furthermore the issue of speaker orientation is not a further detrimental factor in these recordings. In practice, there exists a limit to the source-to-sensor separation in which correlation can be expected, however this is typically beyond the dimensions of a normal office or household room.

Note that in this and subsequent experiments the RMS of the entire GCC function (averaged over the entire recording) is displayed. While this measure gives an indication of the relative signal amplitude from one recording to the next, it does not indicate a floor for the mean peak magnitudes (as the GCC is zero-mean function).

5.1.3 Distance Separating Sensors

A similarly straightforward experiment was carried out by varying the separation of the microphone pairs. The source was placed directly in front of the mid-point between the two sensors, at a distance of 2.5m. The experimental results of analysis of the recordings are presented in Figure 5.5.

Again one can see that as the sensor separation increases the mean cross-correlation falls gradually. This parameter is of specific interest when considering the layout of the recording sensor system; while placing sensors far from one another provides greater resolution of time delay estimation, there is a price to be paid by way of reduced correlation performance.

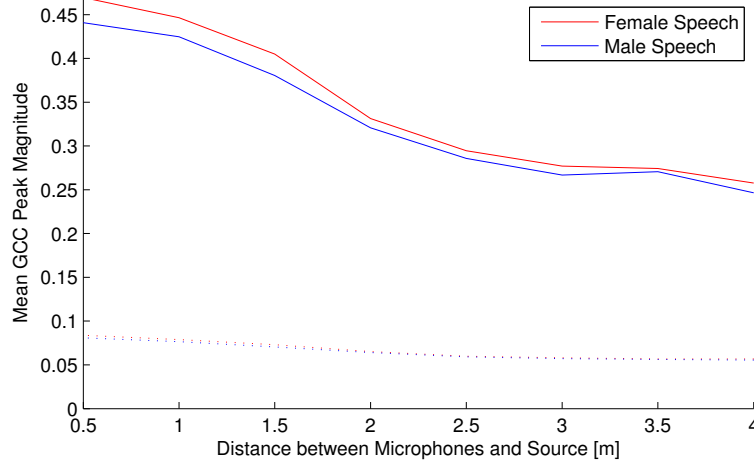


Fig. 5.4. Mean GCC function peak magnitude for two speech samples evaluated while varying the distance between the speech source and microphone pair. Also indicated (with dashed lines) is the mean RMS value of the entire GCC function for each sample.

The recordings at a separation of 0.6m correspond to those carried out in the previous experiment for the source-sensor separation of 2.5m. Note that some inconsistencies in the recording process have led to the small rise seen for the male speech recording at 1m separation.

Finally it should be pointed out that for the largest sensor separation (3.5m), the relative source-sensor angle *for each microphone* is approximately 35 degrees, meaning that the effect of speech directivity will also have a contributing effect — a fact which illustrates the interrelation of all of these system parameters.

5.1.4 Distribution and Bias versus Background Noise and Reverberation

Previous Experimentation

Champagne et al. [17] first studied the distribution and bias of the peak values of the GCC function. Although the authors utilised the ML weighting function

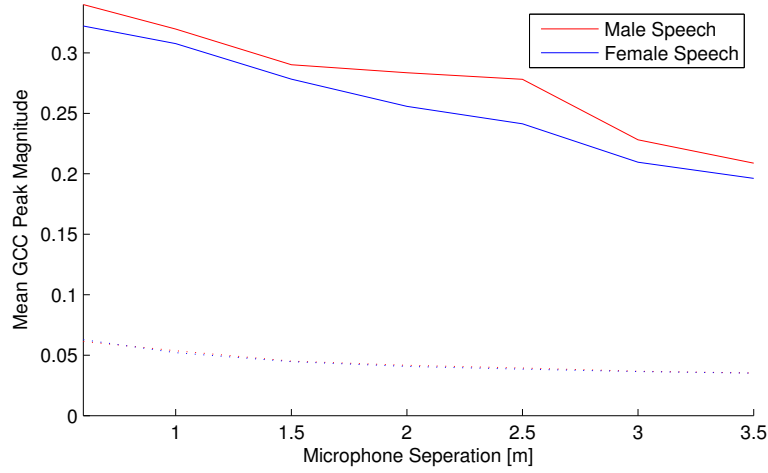


Fig. 5.5. Mean GCC function peak magnitude for two speech samples evaluated while varying the distance between the recording microphones. Also indicated (with dashed lines) is the mean RMS value of the entire GCC function for each sample.

rather than the phase transform (PHAT) and then simulated recordings (of white noise source signals) using the image method [2], some useful conclusions can still be drawn from their work.

First, the authors demonstrated that with increasing (simulated) reverberation one can expect an increasing proportion of anomalous TDOA estimates from the GCC. While they demonstrate a performance collapse for a reverberation time of greater than 0.3 seconds, such a conclusion may not necessarily transfer to the PHAT-GCC function, with speech signals recorded in a real room.

Champagne et al. then went on to demonstrate that non-anomalous GCC function peaks are distributed with increasing variance as reverberation time increases — hypothesising that the reverberation tail of the room impulse response becomes much more significant as the reverberation time increases. Finally they identify a bias specific to the strong initial echoes in the RIR which might be avoided with the judicious placement of microphones distant from reflective surfaces.

Later Vermaak [89] carried out experiments which instead used the PHAT-GCC function and tested real speech samples — however the image method was used to simulate results once more. For a series of noise and reverberation levels the author illustrated that TDOA measurements are roughly distributed according to a zero-mean Gaussian function. This reasonable assumption has since been implicitly made by much of the research which utilises this measurement function. It was also demonstrated there exists no specific bias for different relative microphone delays (referred to as a *source bearing* therein).

As these parameters have been widely studied, we will instead consider the effect that two active sources have on the ability of the GCC function to localise them both.

5.1.5 Two Source Experiments

To demonstrate the unsuitability of using the GCC function to perform the tracking of two sources a series of experimental recordings were carried out with two speech sources. The sources were placed in front of and 2m distant from the microphone pair (with inter-pair separation of 0.6m). The sources were one metre apart. For the first recording both sources faced the microphones while in subsequent recordings the male speaker was turned incrementally until facing away from the sensor pair in the final recording. Note that the actual intensity of the male speech was below that of the female speech.

The results of the experiments are demonstrated in Figure 5.6. The mean GCC function peak in the region surrounding the true source delay (which was 0.77msec and -0.77msec respectively) was evaluated for both the male and female speaker.

The mean GCC function peak magnitude functions presented here (for each source) are comparable to those presented in previous sections but were, however, calculated in a slightly different manner. Here, the mean peak magnitudes for each source were evaluated using the portion of the GCC function $\pm 0.4\text{msec}$ either side of the true source delays while for the signal source experiments the corresponding value was determined over the entire GCC function.

First, compared to the single source experiments carried out in the previous section, these results show that the localisation of each of the two active sources is severely compromised by the corrupting presence of the other source. The mean peak values for each source are significantly lower than those for a single active source recording, especially when compared to the respective RMS noise floor.

When the male source turns to face away from the microphone pair the mean values of the peak of the GCC function converges at the value 0.08 (for 60 degrees and above) — indicating that it has reached a noise floor. Beyond this angle, this particular microphone pair will be of no benefit in localising this source. This is the fundamental behavioural challenge that Ma et al’s multi-target AST algorithm [64] does not consider (as the algorithm was tested using omnidirectional simulated speech samples). However in principle their approach could be adopted to use the SBF measurement function instead.

Because of these issues we will instead turn to the Steered Beamformer function to provide us with more reliable measurement data.

5.2 Experiments with the Steered Beamformer Function

In this section we will experiment with the performance of the SBF function while varying the added noise level, whether or not the source is moving and the presence of another active source. We will also experiment with the shape and density of the grid lattice used to discretise the function surface and the frequency range over which the SBF integration is carried out.

5.2.1 Characterising the SBF Distribution

As previously observed [60] speech is a highly non-stationary signal whose frequency content and activity varies widely from one frame to the next. This means that successive SBF frames may provide clear distinct location measurements that are useful for source localisation while others may only contain clutter measurements. In the following two sections the distribution

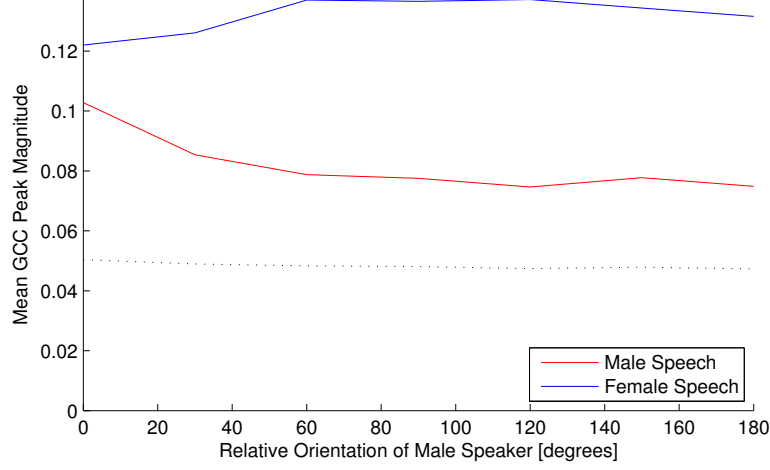


Fig. 5.6. Mean GCC function peak magnitude for two *simultaneously* active speakers, evaluated while varying the relative orientation between the male speaker and the recording microphones. Also indicated (with a dashed line) is the mean RMS value of the entire GCC function for each sample. See Section 5.1.5 for details of how these values were determined.

of accurate source measurements and of nuisance clutter measurements is examined.

Clutter Distribution

Figure 5.7 illustrates the distribution of all the steered response power values for a fully evaluated 4x4m SBF function grid for 8 minutes of continuously active speech taken from different speakers as well as different paths and trajectories. The distribution illustrated in red corresponds to all grid values **more than 30cm** away from the true source location i.e. the noise clutter distribution. The distribution illustrated in blue corresponds to the SBF function peak values i.e. the source distribution. Note that it is common to normalise the steered response range by dividing by $N_m^2 N_{\text{freq}}$; this has not been carried out in this thesis.

Next we will consider what defines the behaviour of the portion of distribution corresponding to noise clutter. The phasor portion of Equation 2.14,

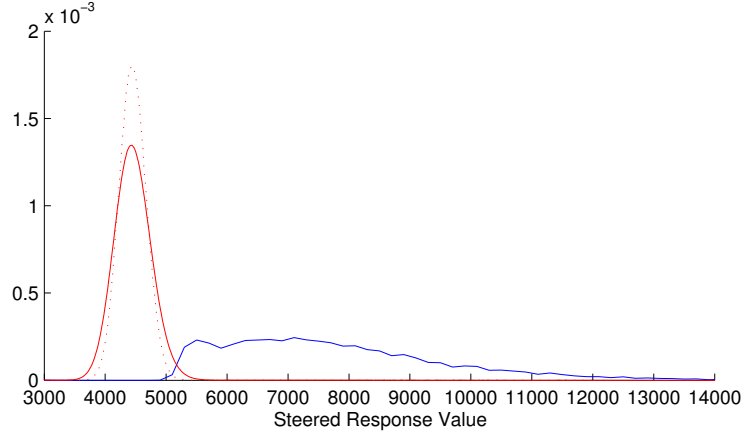


Fig. 5.7. Signal and noise distributions of SBF values (evaluated on a grid of values) for 8 minutes of speech. Blue: signal distribution. Red: noise distribution. Dotted red: simulated noise distribution. See Section 5.2.1 for more details.

pertaining to microphone m , denoted as follows

$$R_m = S_m(\omega)W_m(\omega)e^{j\omega T_m(l)/c}, \quad (5.1)$$

which is defined to have unit magnitude when using the phase transform. Should each of the N_m signal phasors be entirely uncorrelated with one another, the phases will be uniformly distributed in the range $[0, 2\pi]$, as follows

$$\begin{aligned} R_m &= a_m + jb_m \\ u &\sim \mathcal{U}(0, 2\pi) \\ a_m &\sim \cos(u) \\ b_m &\sim \sin(u). \end{aligned} \quad (5.2)$$

In this case, the distribution formed by the Equation 2.14 (across each of the microphones) is an intractable non-standard distribution. However simulations have shown that the statistical characteristics (i.e. mean, mode) of the distribution are directly dependent on the number of frequencies and microphones used in the integration. A simulated noise distribution, with $N_m = 12$ and $N_{\text{freq}} = 371$, broadly corresponds to the noise distribution and is illustrated with a dotted red line in Figure 5.7.

Distribution of Signal Measurements

Next, the distribution of SBF peak values versus clutter values was studied while varying the **density** of a discrete SBF grid. We examined the proportion of signal frames in which a particular SRP threshold was exceeded for (i) the area within 0.3m of the true source location (deemed correspond to signal measurements) and (ii) the remainder of the surveillance region (deemed to correspond to clutter measurements). This threshold was varied in the range 5000–6000 SRP units. The results are represented in Figure 5.8.

Consider the measurements for a grid density of 0.1m. For an SRP threshold of 5500 units, it can be seen that approximately 62% of frames contain clutter measurements while 81% of frames contain signal measurements for this value. Furthermore, it can be seen that beyond this magnitude there is a dramatic fall in the proportion of clutter measurements while the proportion of signal measurements follows only gradually.

For this reason the non-linear CDF-based mapping suggested in Chapter 7.4.2 uses 5500 units as its mean value.

5.2.2 Frequency Integration Range and Density of SBF Grid

In Chapters 7 and 8 algorithms are proposed which implicitly rely on discretising the underlying continuous steered beamformer surface. As mentioned in Section 2.3.2 and implicitly discussed by Lehmann [56], the minimum density at which this surface must be implemented to avoid aliasing is defined by the range of integrated frequencies.

Figure 5.9 represents the results of a study of the grid density parameter in which the frequency integration range and SBF grid density were experimented with. Consider the plot of the data points (upper plot): each data point represents the ratio of the SRP peak value at a particular grid density compared to the SRP peak value for the highest grid density, 0.025m, averaged over many frames. A single data point on this figure can be read as thus: the typical SBF peak magnitude integrated over the frequency range of 100-200Hz and evaluated on a grid of 0.8m cells is approximately 75% of

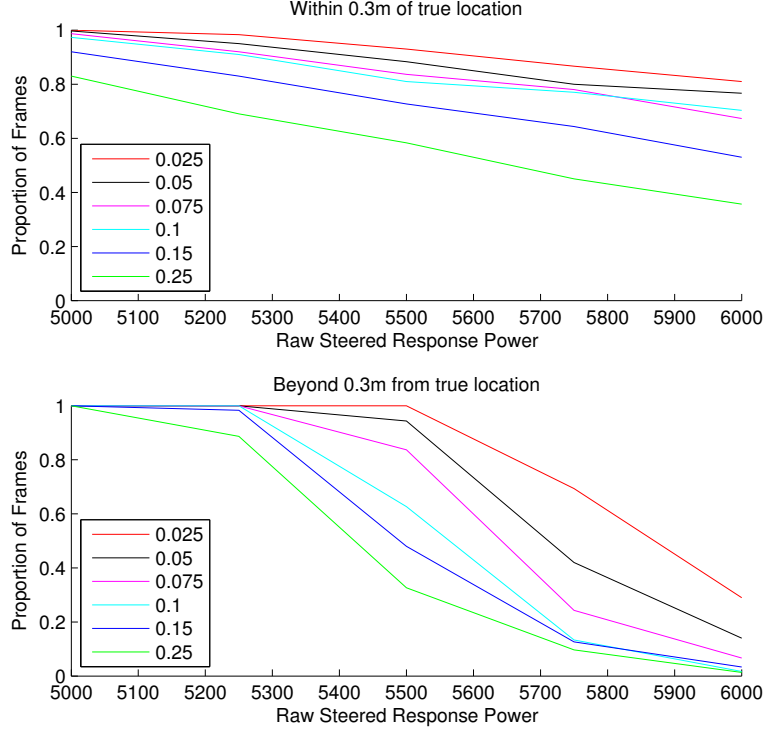


Fig. 5.8. *Proportion of frames in which a particular SRP value was exceeded for points within 0.3m of the true source location (corresponding to signal measurements, upper plot) and points beyond that distance (corresponding to clutter measurements, lower plot). The experiments were repeated for a set of SBF cell grid densities between 0.025m and 0.25m.*

the same function instead determined on a grid with 0.025m density (which requires 1600 times more evaluations points).

Each line on the figures illustrate the performance gain that is to be achieved from increasing the grid density while the SBF frequency range is held constant. Some important trends observed from this function are as follows:

- Most importantly: as the maximum frequency range of integration is increased; the density of grid points must also be increased so as to accurately observe the global maximum of the surface. Roughly speaking: to expect

95% accuracy for the highest frequency (of up to 3000Hz) would require a grid density of about 0.06m, while for the range up to 400Hz a grid of density 0.2m would suffice.

- When using the frequency range 100-400Hz there exists little benefit in increasing the grid density beyond 0.2m cells.
- For the higher ranges (2000Hz and 3000Hz) little benefit is gained from using a 0.02m density rather than a 0.04m density grid. This suggests that above 2000Hz there is little significant localising information and indicates an upper bound on the accuracy of any acoustic source localisation algorithm using the steered beamformer.
- Very similar results were found upon repeating the experiment with a moving source. It was not deemed beneficial to present these results. Any comparison of the differences between the two scenarios would require further testing on a larger set of samples.

It should be pointed out that the measurements were carried out using typical samples of speech — containing both active and inactive portions. The effect of this is that the trends observed are expected to be more severe for a set of samples purely containing high amplitude active speech.

5.2.3 Source Movement

An important assumption of many AST algorithms is that the duration of time-frames is small when compared to the duration of events such as a speaker turning their head or quickly moving — known as stationarity. We assume that the movement of the source is not significant enough to effect the room-impulse response significantly *during* a time-frame.

To verify that this assumption holds for the SBF measurement function, the next experiment (following the same format as in Section 5.2.1) examined the proportion of frames containing measurements as compared to the proportion containing clutter measurements. The results, presented in Figure 5.10, do not show any substantial difference when compared to the stationary source results in Figure 5.7 although the speed of movement, 0.2m/s, was relatively low.

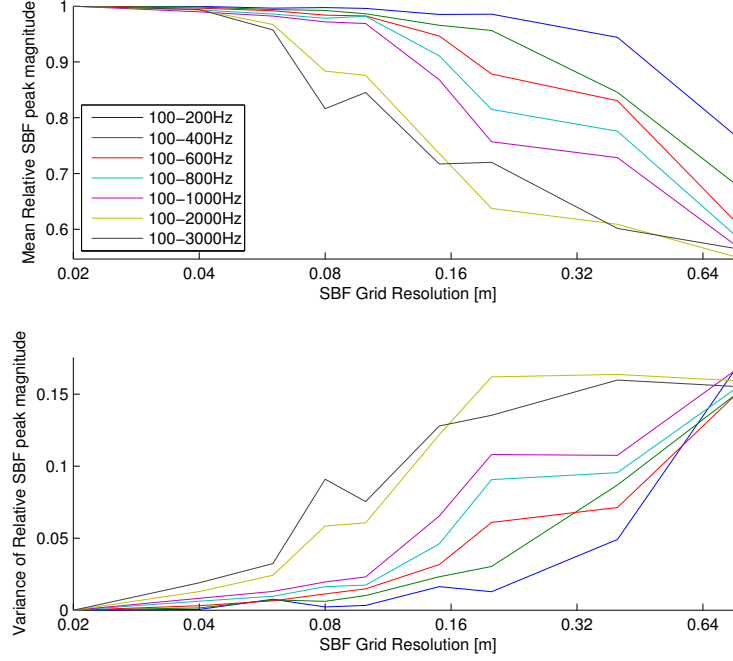


Fig. 5.9. Experiment to study the SBF grid for different frequency ranges and grid densities for a stationary source. The upper plot represents the relative mean SBF peak magnitude for an SBF grid at a specific grid density, when compared to the mean peak magnitude of a similar grid implemented for the most dense grid (with spacings of 0.02m). The lower plot illustrates the associated variances. Note that the grid density axis is measured on a logarithmic axis.

5.2.4 Ambient Noise Level

To examine the robustness of the (discretised) SBF measurement function to different added noise levels, this type of experiment was repeated while varying the level of white noise added to an original recording. The results presented in Figure 5.11 are of a similar format to that previously used, however instead of repeating the experiment for different SBF grid resolutions, this was instead fixed at a cell size of 0.1m for all experiments.

While the distribution of frames containing clutter measurements (in the SRP range of 5000-6000 units) remains quite stable as the level of added

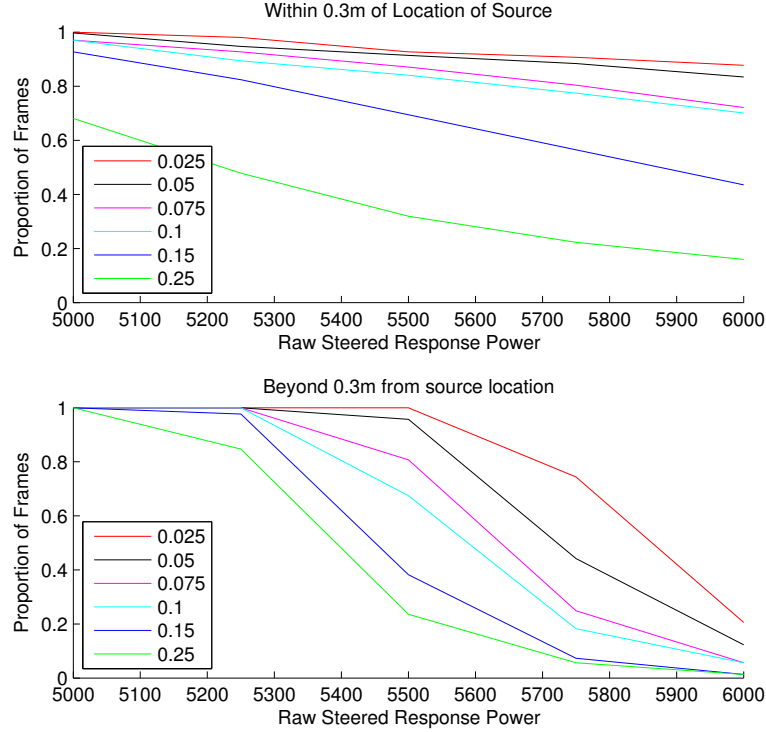


Fig. 5.10. *Proportion of frames in which a particular SRP threshold was exceeded for points within 0.3m of a **moving** speech source (upper plot) and for the set of points outside of each region (corresponding to clutter measurements, lower plot). The experiments were repeated for a set of SBF cell grid densities between 0.025m and 0.25m. Note that the larger than expected performance reduction for a grid density of 0.25m can be attributed to the 0.3m threshold for nearby measurements.*

noise increases, the proportion of useful source measurements falls off heavily between 5-10dB. The mean of the clutter/noise portion of the distribution is defined by the number of the microphones and the number of frequencies involved in the integration. As the level of added noise is increased, previously useful localisation frames tend to this distribution — although the clutter distribution, itself, does not change substantively.

Because the clutter distribution is unchanged, the behaviour at the previously discussed threshold of 5500 SRP units remains approximately unchanged

— regardless of the level of added noise. This means that modification of resultant algorithms is not required for different noise levels.

While this experiment indicates that there is a noise level at which tracking will no longer be possible, it is expected that the tracking performance of a related algorithm will only gradually deteriorate with increasing noise level. Comparison between these tracking modalities is carried out in Section 7.5.1 and accurate SBF-based tracking with a SNR of -5dB was demonstrated.

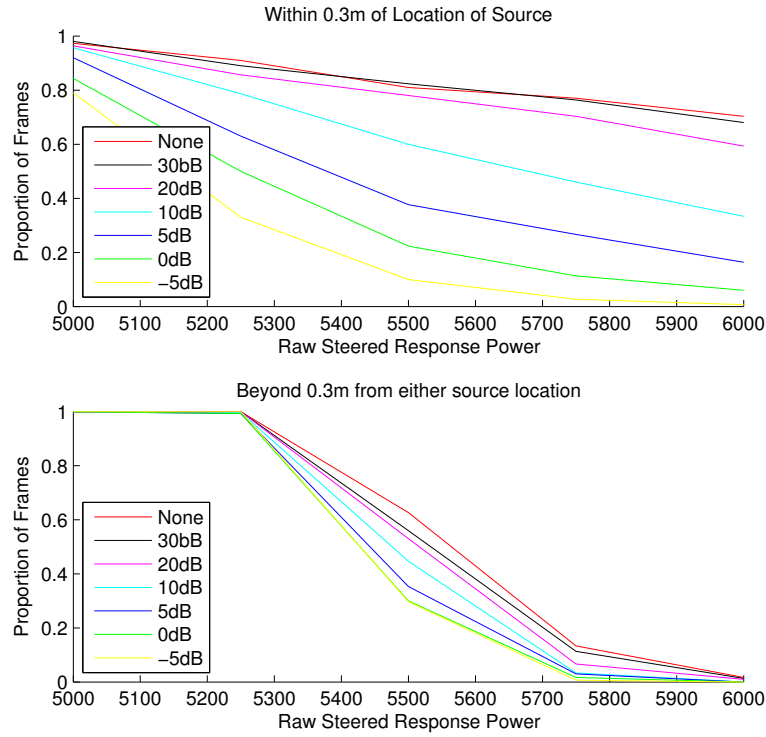


Fig. 5.11. *Proportion of frames in which a particular SRP threshold was exceeded for points within 0.3m of a speech source (upper plot) and for the set of points outside of each region (corresponding to clutter measurements, lower plot). The experiment was repeated with white noise added to the original signal in increments of 30dB to -5dB. A grid density of 0.1m was used throughout.*

5.2.5 Two Source Experiment: Interference between Sources

In general terms, speech and music are said to be *sparse* — meaning that much of a signal’s energy can be concentrated in a small portion of the coefficients of a transform space (such as the Modified Discrete Cosine Transform or the Short Time Fourier Transform) with little or no energy in the remaining coefficients. This characteristic of speech and music has been the motivation for a wide variety of audio research including analysis [38, 78, 87], blind source separation [95], coding [1] and music analysis [39].

Typically this type of research requires anechoic recordings of music or speech. Reverberation, however, causes an effect known as *convolutive spectral smearing* [66] in which the speech content of each source is smeared across time. As a result recordings made in reverberant conditions are strongly affected by the presence of other sources and sparsity cannot be leveraged in the same manner.

The frequency and magnitude of SBF source location measurements is affected by the activity of the speech sources. In some circumstances, useful SBF measurements from a particular source may be non-existent for a significant amount of time (e.g a couple of seconds) despite the speaker talking. The performance of any SBF tracker should reflect this reduction in the proportion of useful location measurements.

This effect was examined in much the same way as the experiment in Section 5.2.1. A 40 second sample of speech in which two people were simultaneously speaking was examined in the SBF domain on a full grid of size 4x4 metres. The two sources were 2.6m apart. Figure 5.12 represents the results of the experiments.

Consider the results presented for a grid density of 0.1m. For an SRP threshold of 5500 units, it can be seen that approximately 60% of frames contain clutter measurements (measurements above 5500 SRP units and beyond 0.3m from the sources) while 46% and 65% of frames contain signal measurements (measurements above 5500 SRP units and within 0.3m of the

source) for source 1 and source 2, respectively². This compares with the single source scenario in Figure 5.7 in which 81% of frames contained source position measurements.

There is a substantial fall in the proportion of source position measurements when more than one source is simultaneously speaking. As a result the parameters of the multi-target tracking algorithms in Chapters 7 and 8 were changed to reflect this. These changes, however, are the only fundamental problem that complicates multi-source tracking versus single source tracking.

In summary, this section has indicated that the maximum number of simultaneously active sources which may be tracked using the SBF measurement function is quite low (in the region of 2–3) unless more fundamentally different approaches are taken — specifically pre-filtering the signals using source separation.

5.3 Conclusions

In this chapter we examined a number of the parameters of the GCC and SBF localisation functions. We observed that the effects of source directionality are significant — yet regularly overlooked. We then examined the effect that this parameter has on signal-to-signal correlation.

Other experiments illustrated that two speech sources disastrously interfere with one another when simultaneously active — thus causing much complication for GCC function-based tracking algorithms. This motivates our consideration of the Steered Beamformer function.

The distribution of measurements drawn from the SBF was also examined. A set of different parameters including noise, movement and inter-source interference were studied. The frequency of accurate source measurements was seen to be stable in the face of changes to these parameters and that sufficient measurements were available for both one and two source tracking.

² Note that typically the threshold levels for source 1 were below those of source 2, this is because the actual speech intensity of that source was lower.

Further experiments on the SBF function illustrated that to minimise computation while avoiding aliasing, one must consider the frequency range of integration and the grid cell density as which it is implemented jointly.

Note that no consideration was given to a fast **approximation** to the SBF surface. If such an approximation could evaluate the SBF surface to sufficiently low deviation from the actual surface with low computation; this could, perhaps, alter the focus and approach that is currently considered in many of the algorithms discussed previously.

In summary, the results presented in this chapter will be utilised in the following chapters, we will first consider speaker directionality to estimate orientation in Chapter 6.

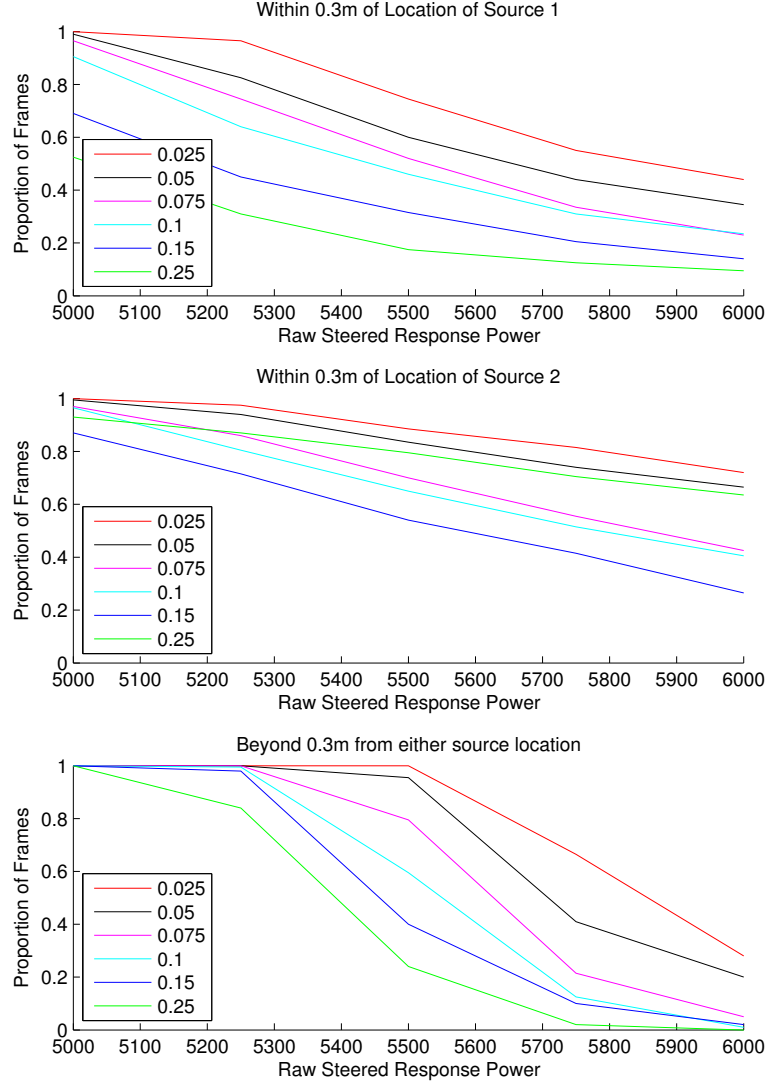


Fig. 5.12. *Proportion of frames in which a particular SRP threshold was exceeded for points within 0.3m of source 1 (upper plot), for source 2 (centre plot) and for the set of points outside of each region (corresponding to clutter measurements, lower plot). The experiments were repeated for a set of SBF cell grid densities between 0.025m and 0.25m.*

Orientation Estimation and Tracking

In the previous chapter we demonstrated that speaker directivity and orientation can have a substantial effect on localisation measurement functions — particularly the GCC function formed from individual microphone pair recordings. Using this effect it is possible to estimate and track the source orientation using correlation measurements.

In this chapter we will introduce an algorithm to track speaker orientation. The algorithm neatly fits within the existing AST framework, illustrating how easily extensible the particle filtering framework is. In Section 6.1 we will modify the state vector by introducing speaker orientation variables, while a novel likelihood function for speaker orientation is then formed in Section 6.3.2. Finally in Section 6.4 experimental results using real audio recordings are detailed.

The development of this work was first presented in [37].

6.0.1 Previous Approaches

Previous attempts to estimate speaker orientation, or more generally head pose, have focused on instantaneous frame-by-frame techniques.

An approach by Sachar and Silverman [81] presents many useful experimental observations. The authors consider the importance of a number of parameters including sound transit time around the head, the non-absorption of low frequencies by the head, the frequency profile of the background noise,

the (lack of) directivity of the reverberant portion of the audio as well as implementing a correction for the speaker-to-microphone attenuation path.

Their proposed method essentially estimates the weighted mean of each signal from each of the 512 elements of the Huge Microphone Array [84] which surround the speaker. This gives an approximate speaker directivity pattern from which the orientation can be estimated. Although this approach is simple, logical and straightforward, the number of microphones is indeed substantial. Correlation information between microphones was not considered.

A much less data intensive approach [12] considers signal correlation instead. The frame-by-frame orientation estimation algorithm presented therein takes a related approach to the likelihood function presented in this chapter and uses a similar experimental setup. Essentially, the proportion of the contribution from a particular microphone pair to the Global Coherence Field (a pairwise version of the SBF function surface) for a region (a 5cm circle) surrounding the assumed speaker position, is assumed to represent the likelihood that the speaker is facing in that direction. An extension of this algorithm [13] combines orientation estimation with location estimation to marginally improve the localisation accuracy.

Finally a recent publication [14] has combined orientation estimates from a distributed microphone array with video information garnered from 4 cameras to form a multi-modal particle filter. The audio-only algorithm presented therein focuses on the frequency dependent nature of speaker directivity (Figure 5.1) rather than on correlation. Interestingly their proposed audio-only approach outperforms a video-only approach. As expected, combined audio-video orientation estimation was more accurate than either individual modality.

Each of these approaches have utilised a surrounding microphone array.

6.1 Estimating Orientation using Speech Directionality

The experiments of Section 5.1.1 showed that when the speaker is not facing directly at a microphone pair one can expect a reduced peak magnitude in

the GCC function. The amount that this peak is reduced is directly related to the relative orientation between the speaker and the microphone pair. This was best depicted in Figure 5.3.

In this chapter we will form a probabilistic representation of this information.

6.1.1 Problem Framework

The proposed algorithm will utilise the particle filter framework outlined in Section 4.2. We will first modify the state vector by introducing the specific orientation parameters which are to be estimated.

The source state vector at time k will be subdivided into its location and orientation components as follows

$$\alpha_k \triangleq (\alpha_{l,k}, \alpha_{o,k}) \quad (6.1)$$

which are in turn defined as

$$\begin{aligned} \alpha_{l,k} &\triangleq (x_k, \dot{x}_k, y_k, \dot{y}_k) \\ \alpha_{o,k} &\triangleq (\theta_k, \dot{\theta}_k) \end{aligned} \quad (6.2)$$

where the former will retain the variable definitions from Section 4.2, while the latter will be made up of the orientation of the speaker, θ_k , and the rate of change of orientation, $\dot{\theta}_k$.

6.1.2 Identifying Source Peaks

At time frame k the PHAT-GCC function will be evaluated for the r th microphone pair from the current frame of audio according to Equation 2.8.

Next a set of N_c promising candidate time delay estimates, $T_k^{(r)} = (\hat{\tau}_1^{(r)}, \dots, \hat{\tau}_{N_t}^{(r)})$, will be isolated from this GCC function. The entire set of delay measurements for all N_r microphones, at time frame k , is thus $\mathbf{T}_k = (T_k^{(1)}, \dots, T_k^{(N_r)})$.

In parallel to this we will also retain the corresponding set of GCC function *magnitudes*. For a single peak of the GCC function, $\hat{\tau}^{(r)}$, the magnitude corresponds to

$$\hat{m}_t^{(r)} = R_r(\hat{\tau}_t^{(r)}). \quad (6.3)$$

which is the value of the GCC function, $R_r(\tau)$, evaluated at the relative delay $\hat{\tau}_t^{(r)}$, as defined in Equation 2.8.

In the same way that the delay measurement vector for microphone pair r was constructed above, the magnitude measurement vector for the microphone pair is $M_k^{(r)} = (\hat{m}_1^{(r)}, \dots, \hat{m}_{N_t}^{(r)})$ and the overall set of magnitude measurements for a particular time frame k will be $\mathbf{M}_k = (M_k^{(1)}, \dots, M_k^{(N_r)})$.

Finally the overall combined measurement vector will be a combination of the two vectors, $\mathbf{D}_k = (\mathbf{T}_k, \mathbf{M}_k)$, in which the pairs of elements correspond to the relative delay and magnitude of a particular peak from a particular GCC function.

How one decides upon which observations to consider is open to some discussion and is in many ways heuristic. Thresholding of the GCC function, a simple and commonly used method, was used here. Function peaks above a correlation magnitude are retained, typically we used a correlation threshold of 0.09 units (which is a normalised value, as we are using the PHAT weighting function).

It is also useful to limit the number of candidate peaks to a reasonable maximum using a constant $N_{t,max}$. Regardless of the mechanism used, the set of candidate measurements should contain all peaks likely to be due to a speech source but may well include incorrect noise measurements.

6.2 Dynamical Models

The next step is to outline dynamical models for orientation and location which will be used to propagate the particle positions. The dynamical models for orientation and location will be assumed to be independent. This essentially assumes that the direction that a person is facing will not provide any further information about the direction we would expect the speaker to walk, although in practice this assumption is not necessary true.

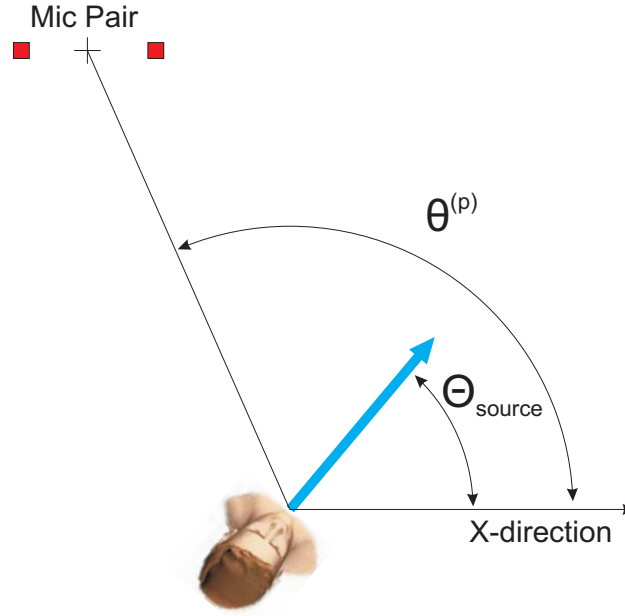


Fig. 6.1. Overhead illustration of the speaker position and orientation relative to a microphone pair.

Localisation Movement Model

Motion in the $\mathcal{X}$ -coordinate will be modelled by the second order Langevin motion system described in Equation 4.2. The parameter values will be set to $\beta_{\mathcal{X}} = 10\text{Hz}$ and $\bar{v}_{\mathcal{X}} = 0.7\text{m/sec}$. Movement in the $\mathcal{Y}$ -coordinate will be independent of this, but will use an identical framework and parameter values.

Orientation Movement Model

We will retain the Langevin motion model when propagating orientation dynamics. The model parameters, β_{θ} and $\bar{v}_{\theta}$, will be set to be 10Hz and 1rad/sec respectively, which were observed to give reasonable performance for the envisaged type of motion.

Note that in the following, particle orientations are limited to the range $[-\pi : \pi)$ with wrap-around applied to all particles propagated beyond these bounds.

6.3 Measurement Likelihood Model

A likelihood function for joint orientation and location tracking is proposed in this section. As movement in each domain is assumed to be independent the overall likelihood function will simply be the product of the individual likelihood functions

$$\begin{aligned} p(\mathbf{D}_k|\alpha_k) &= p(\mathbf{D}_k|\alpha_{l,k}, \alpha_{o,k}) \\ &= p(\mathbf{T}_k|\alpha_{l,k})p(\mathbf{M}_k|\alpha_k). \end{aligned} \quad (6.4)$$

Note that to track the source location alone, one need only set $p(\mathbf{M}_k|\alpha_k) \propto 1$ and to remove all orientation parameters from the state vector¹. Conversely, orientation-only tracking (using a known source location) is possible by setting $p(\mathbf{T}_k|\alpha_{l,k}) \propto 1$ and removing the location parameters. This is further discussed in Section 6.3.2 and experimentally illustrated in the Section 6.4.

For the remainder of this section the time index k has been suppressed for ease of reading.

6.3.1 Localisation Likelihood Functions

The localisation likelihood function will follow a similar form to that introduced in [90] and also discussed in Chapter 4. This framework allows two hypotheses: either that one of the measurements, t , is due to the source and the rest are due to clutter, $\mathcal{H}_t$, or alternatively all of the measurements are due to clutter, $\mathcal{H}_0$.

This approach is essentially as used for tracking in clutter (RADAR, SONAR), see [4] for more details. The basic framework is represented here for completeness.

Hypothesis $\mathcal{H}_t$: For microphone pair r , if the t th candidate delay measurement, $\hat{\tau}_t^{(r)}$, is due to the true source we will represent the resultant likelihood function using a normal distribution as follows

¹ The orientation likelihood is dependent on the source location vector, $\alpha_{l,k}$, and the source orientation vector, $\alpha_{o,k}$.

$$p(\hat{\tau}_t^{(r)}|\alpha_l, \mathcal{H}_t) = \begin{cases} c_{\alpha_l} \mathcal{N}(\hat{\tau}_t^{(r)}; \hat{\tau}_\alpha, \sigma_l^2) & \text{for } |\hat{\tau}_t^{(r)}| \leq \tau_{\max}^{(r)} \\ 0 & \text{otherwise} \end{cases} \quad (6.5)$$

where c_{α_l} is a normalising constant due to the limited admissible delay region. The TDOA is assumed to be corrupted by Gaussian observation noise with variance σ_l^2 . The normalising constant is given by

$$c_{\alpha_l} = 2 \left(\operatorname{erf} \left(\frac{\tau_{\max} - \tau_\alpha^{(r)}}{\sqrt{2}\sigma_l} \right) - \operatorname{erf} \left(\frac{-\tau_{\max} - \tau_\alpha^{(r)}}{\sqrt{2}\sigma_l} \right) \right)^{-1} \quad (6.6)$$

where $\operatorname{erf}(x) \triangleq \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$ is the normal error function.

Hypothesis $\mathcal{H}_0$: The likelihood of one of the measurements being associated with clutter is given by a uniform distribution within the admissible interval

$$p(\hat{\tau}_t^{(r)}|\alpha_l, \mathcal{H}_0) = \mathcal{U}_{\mathcal{D}}(\hat{\tau}_t^{(r)}). \quad (6.7)$$

These $N_t + 1$ hypotheses are summed together to give a final likelihood function for microphone pair r

$$p(T^{(r)}|\alpha_l) = \sum_{t=0}^{N_t} q_t p(\hat{\tau}_t^{(r)}|\alpha_l, \mathcal{H}_t). \quad (6.8)$$

where q_t are the prior hypothesis probabilities which are chosen to reflect confidence in the measurements. The measurement hypotheses probability will be held equal while the clutter hypothesis probability (q_0) is chosen to reflect the frequency of clutter measurements. In what follows q_0 will be set to 0.7 and $q_c = 0.3/N_c$ for $c > 0$.

Overall Likelihood Function: Finally the overall likelihood function will be the product of the likelihood functions for each microphone pair

$$p(\mathbf{T}|\alpha_l) = \prod_{p=1}^{N_r} p(T^{(r)}|\alpha_l) \quad (6.9)$$

6.3.2 Orientation Likelihood Functions

In this section a novel orientation likelihood function, $p(\mathbf{M}|\alpha_o)$, is proposed. When combined with the localisation likelihood function, joint tracking in both domains is possible.

The likelihood function will be formed using information from the measurement magnitude vector, $\mathbf{M}$, as well as knowledge of the specific particle and microphone locations. The function also requires that the microphones be distributed evenly surrounding the source so as to produce an unbiased estimate in the full range $[-\pi : \pi)$. This limitation is discussed in Section 6.5.

We will assume that the magnitude of the peak of the GCC function formed from the recordings at a particular microphone pair is an measure of how likely it is that the source is facing in that particular direction. When combined across all microphone pairs, this gives an overall estimate of orientation.

The hypothesis structure previously used in Section 6.3.1 is retained; however only the largest peak from each GCC function is considered here. Extension of this approach to the full set of peaks, $T^{(r)}$, is possible but has not been carried out here.

Hypothesis $\mathcal{H}_1$: Measurement due to the source. In this case we assume that the magnitude of the peak of the GCC function is a measure of the amount of correlation between two signals. Assuming all other parameters are held constant, Figure 5.2 implies that greater correlation indicates that the source is more likely to be facing in the direction of this particular microphone pair.

Although the figure suggests that closed form distributions of signal correlation may be determined, it must be noted once more that the radial distributions formed were each drawn from recordings of white noise recorded by microphones uniformly positioned around the source. Because of the huge variability of signal correlation (due to speaker inactivity, signal frequency content and signal interference) greater correlation can only be taken to be *indicative* of the speaker orientation rather than a direct measure of orientation. As a result, the proposed likelihood function focuses on the relative difference between correlation magnitudes instead.

Consider microphone pair r with microphones positions defined as l_r^1 and l_r^2 . The midpoint of this microphone pair is then simply

$$l_r^{\text{mid}} = \frac{l_r^1 + l_r^2}{2}. \quad (6.10)$$

Now consider a particle with position $l = (x, y)$; its position relative to the midpoint of microphone pair r is given by

$$l_r^{\text{rel}} = l - l_r^{\text{mid}}. \quad (6.11)$$

Using this vector the likelihood that the source is facing in direction θ is proposed as follows

$$p(\hat{m}^{(r)}|\alpha, \mathcal{H}_1) = \mathcal{N}(\hat{\theta}^{(r)}; \theta, \sigma_{\hat{\theta}}^{(r)}). \quad (6.12)$$

using a normal distribution with mean and variance determined as

$$\hat{\theta}_r = \angle(l_r^{\text{rel}}), \quad \sigma_{\hat{\theta}}^{(r)} = \frac{a}{\hat{m}_{\text{max}}^{(r)}} \quad (6.13)$$

where a is a experimentally determined constant. As such the mean of the normal distribution will be the relative angle between the source and the microphone pair. The standard deviation is inversely proportional to the **maximum value** of the measurement magnitude vector, $\hat{m}^{(r)}$.

Note that it was later observed that this form can be reconstituted as a gamma-type distribution evaluated at $\hat{m}_{\text{max}}^{(r)2}$ with a scale parameter related to the θ^2 . It is presented as originally proposed so as to illustrate its function.

In essence we propose that the variance of the likelihood function be inversely proportional to this correlation magnitude. In this way the proposed orientation likelihood function will exhibit a peak in the direction of a microphone pair when the related recorded signal show strong correlation.

Hypothesis $\mathcal{H}_0$: Measurement due to clutter. In this hypothesis we assume that the clutter hypothesis for this algorithm is again a uniform distribution formed across the range $[-\pi, \pi)$ for each microphone pair r

$$p(M^{(r)}|\alpha, \mathcal{H}_0) = \mathcal{U}_{\mathcal{D}}(M^{(r)}). \quad (6.14)$$

For this microphone pair the likelihood function is as follows

$$p(M^{(r)}|\alpha) = (1 - q_T)p(M^{(r)}|\alpha, \mathcal{H}_0) + (q_T)p(M^{(r)}|\alpha, \mathcal{H}_1) \quad (6.15)$$

where q_T is the prior probability that the source is present in microphone pair r .

Finally the complete orientation likelihood function is the product of the individual microphone pair likelihood functions

$$p(\mathbf{M}|\alpha) = \prod_{r=1}^{N_r} p(M^{(r)}|\alpha). \quad (6.16)$$

The proposed likelihood functions are presented graphically in Figure 6.3.2 and in Figure 6.3.2.

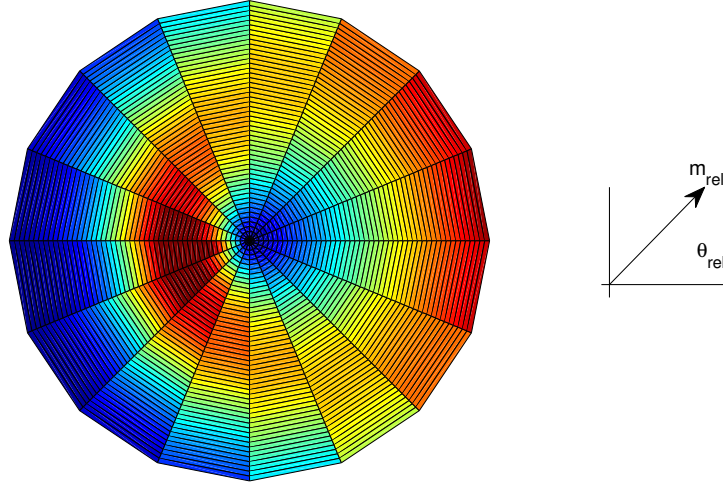


Fig. 6.2. Graphical representation of the proposed orientation likelihood function when varying GCC peak magnitude, m_{rel} , versus relative angle between source heading and microphone pair, θ_{rel} . The figure replicates the form of Figure 5.3 and illustrates that the proposed distribution behaves in a similar manner to the measured distribution. Note these segments are not normalised.

6.3.3 Other Likelihood Functions

Other proposals for a suitable likelihood function were also implemented. The first combined the set of maximum GCC peak magnitude vectors ($\hat{m}_{\text{max}}^{(1)}, \dots, \hat{m}_{\text{max}}^{(N_r)}$) into a single overall orientation estimate which was then used to construct a

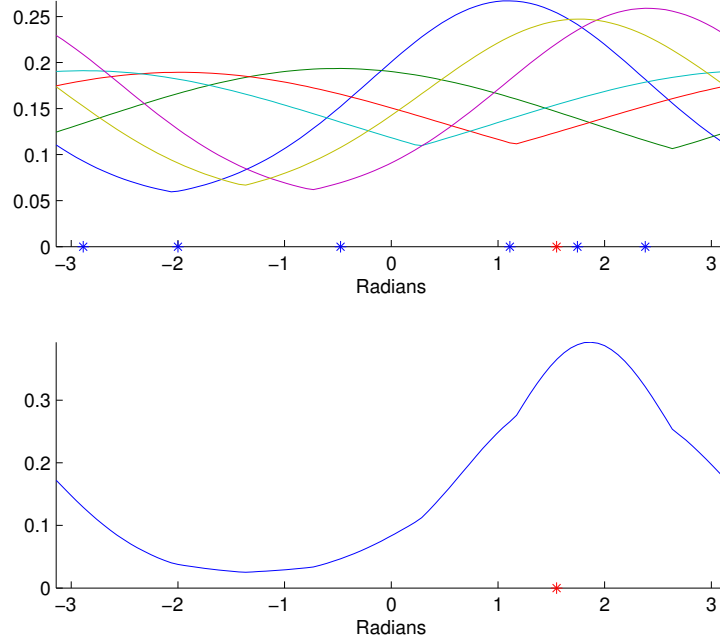


Fig. 6.3. Graphical representation of the proposed orientation likelihood function in use. Upper: Individual microphone pair likelihood functions. Lower: Overall likelihood function — the product of the individual functions. The true source orientation (red *) and the relative orientation of the microphone pairs (blue *) are also illustrated. See Sections 6.3.2 and 6.4 for more details and the specific parameters used.

likelihood function. Another proposal replaced the Gaussian normal distribution used in Section 6.3.2 with a rectangular distribution (so as to approach the distributions suggested in Figure 5.2).

However each of these proposed distributions failed to outperform the method described in Section 6.3.2 and without any further investigation were abandoned. Taking a broader view, more fundamental biasing issues should be corrected before optimising the likelihood function. These issues are discussed in Section 6.5.

6.4 Experiments

6.4.1 Experimental Setup

The recording environment used is that explained in Section 2.1.2 and is also used in all subsequent chapters. The set of 12 microphones were arranged into pairs with a uniform intra-pair spacing of 0.6m.

The algorithm parameters specific to the orientation estimation portion of the algorithm are as follows:

- Number of particles: $N_p = 150$
- Variance constant: $a = 0.4$
- Prior probability of source being present: $q_T = 0.3$

The number of particles is sufficiently small to allow performance many times quicker than real time when implemented with MATLAB on a typical PC (Intel Core Duo 1.2GHz, 2GB RAM) and as such no specific optimisations to reduce computation have been attempted. The algorithm employed systematic resampling [31] to maintain particle diversity.

6.4.2 Typical Tracking Examples

Two examples of typical orientation tracking performance are illustrated in this section. The first example, of orientation-only tracking of a stationary source, is seen in Figure 6.4. The source (a computer speaker) was placed at the centre of a room and turned manually, first anti-clockwise until 7 seconds, and then clockwise. The particle filter is seen to clearly track the source's changing orientation.

The second example illustrated is of the source simultaneously moving and turning around the room. Figure 6.5 shows the results for joint tracking. As expected, the particle filter comfortably tracks the movement and orientation for the first two-thirds of the signal. The remainder of the example illustrates the performance limitations of bearings only tracking.

At this particular source location and orientation, only the microphone pair in the upper right of the room can provide useful bearing estimates.

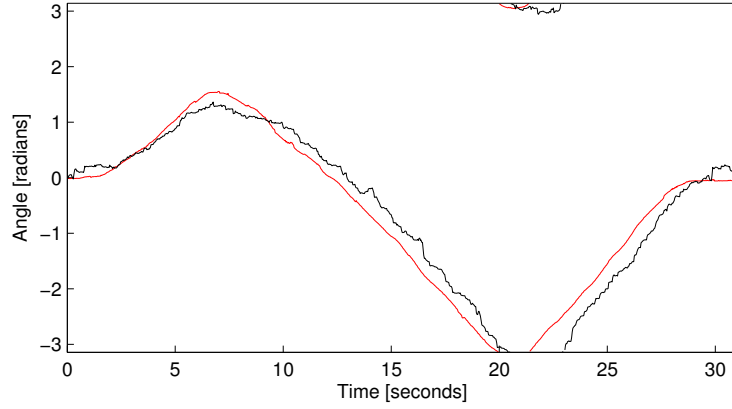


Fig. 6.4. Example of orientation tracking performance (black line) of a real source which is turning, but stationary, at the centre of an office room. The red line represents the ground truth orientation.

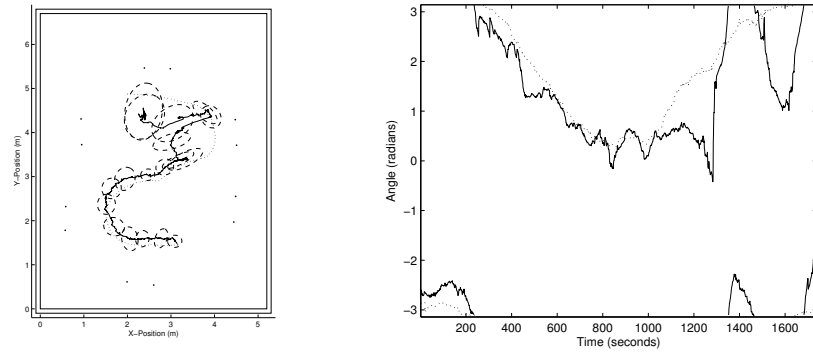


Fig. 6.5. Example of joint location (left) and orientation (right) tracking performance (solid line) of a real moving and turning source (dotted line). Particle position and tracking performance is indicated by the loops of equal variance (dashed lines, top). The microphone positions and the room boundary are also shown.

This causes badly reduced performance radially from the microphone pair as the GCC functions drawn from the other microphone pairs show very little correlation because of source directivity. Also during this final section the source turns quickly anti-clockwise. The combined effect causes the orientation tracker's performance to fail although the particle filter does regain the correct track after a period of time.

Path	1	2	3	4	5
Location RMSE	0.322	0.247	0.114	0.155	-
Location MSTD	0.051	0.070	0.06	0.074	-
Orientation RMSE	0.134	0.313	0.558	0.336	0.381
Orientation MSTD	0.073	0.227	0.115	0.377	0.300

Table 6.1. *Results for experimental tracking for each recording. The units of the performance measures are metres and radians respectively. No location results are presented for Path 5 because the source was stationary.*

6.4.3 Monte Carlo Simulations

The proposed algorithm was tested on 5 different audio recordings using Monte Carlo Simulation to further evaluate typical performance. The algorithm was run 50 times and the statistics were then averaged to give the results in Table 6.1.

The paths the speaker took in four of these recordings are shown in Figure 6.6. The source movement was approximately constant at typical walking speeds. Orientation was maintained in the direction of movement. The fifth recording was of a stationary source located at the co-ordinates (2.7m,3.9m) turning on its axis as shown in Figure 6.4. Because of the nature of the microphone pair spacings and the different paths and source signals used performance is expected to vary from recording to recording. The statistics evaluated are root mean square error (RMSE) and mean standard deviation (MSTD) of the particle cloud (see Appendix B for more details). The first statistic gives an indication of tracking performance while the second is an estimate of tracking stability.

The results in Table 6.1 illustrate that the algorithm can robustly estimate both orientation and location jointly. As expected Path 1 has better orientation estimation performance as it does not change orientation while it moves across the room. Paths 2–5 all show similar orientation estimation performance. Paths 2 and 3 illustrate good tracking performance in the face of changes in both orientation and location.

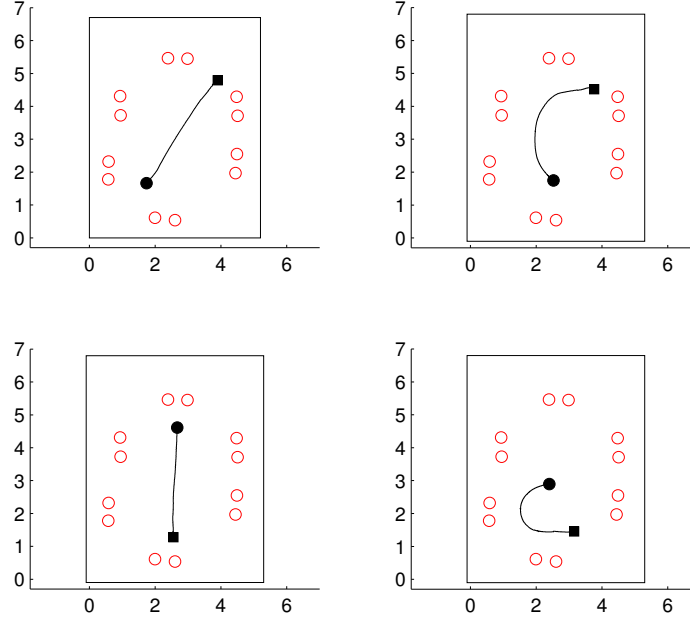


Fig. 6.6. Source paths used to evaluate performance statistics. Each path runs from the square marker to the circle marker and lasts approximately 20-30 seconds, the room boundary and microphone positions are also shown. The paths are numbered 1-4 from the top left, clockwise. The dimensions of each axis are metres.

6.5 Conclusions

In this chapter the problem of joint orientation and location estimation of a moving speech source has been discussed. We introduced an algorithm to estimate and track speaker orientation while maintaining the same framework used in the previous localisation literature. This allows us to simultaneously estimate both parameters jointly.

Results of real audio experiments were promising, however future research could improve performance further. As mentioned in previous sections the likelihood function is by no means perfect and as such a good deal of experimentation is required to achieve optimal performance.

Note that the very accurate tracking performance presented in Figure 6.4 is aided by the positioning of the source at the very centre of the room and uniformly surrounded by microphone pairs.

To improve general performance to this level, it is necessary to correct for the bias caused by an uneven distribution of microphones around the source. Experimentation has shown that differences in direct path between the source and individual microphone pairs produce a constant bias (in other words the likelihood function is biased to face towards the closest microphone pair) which could be eliminated using a signal path attenuation model, for example.

Furthermore it is expected that for many applications there will not exist a uniform distribution of microphone pairs surrounding the room and even if it should exist, there will still be specific positions of the room in which the distribution will be uneven. Another estimation bias is caused by this issue which could possibly be solved by more accurate modelling of the GCC peak distributions — particularly by inferring orientations which are **unlikely** when measured signal correlation is low.

Finally, as mentioned in Section 6.0.1, the frequency attenuation-based orientation function recently proposed by Canton-Ferrer et al. [14] is an interesting advance towards solving this particular issue.

Track Before Detect Acoustic Source Tracking

Published research in the Acoustic Source Tracking field has introduced and compared novel localisation and tracking algorithms while not explicitly recognising the properties of the underlying measurement function or incorporating the behavioural properties outlined in Chapter 5. More specifically, a disproportionate amount of algorithm computation is devoted to the raw evaluation of the SBF function for particles located very close to one another (within 1cm) despite the frequency content of the incoming signals precluding the estimation of the function to such precision. The resultant effect of this approach is that to maintain real-time operation of the algorithm either more computational power or fewer particles must be used.

In this chapter we propose a novel algorithm which utilises the Track Before Detect (TBD) methodology to more evenly distribute computation. This algorithm allows us to retain a large particle set which results in much more stable performance.

Further, as discussed previously, the highly non-stationary nature of speech must also be recognised and acted upon to allow for realistic source tracking during speech inactivity. The method proposed in this chapter will *directly* model activity from the Steered Beamformer function without recourse to typical Voice Activity Detection (VAD) algorithms (which are an indirect measure of the activity of our localisation measures). Furthermore, knowledge of the source's activity is a useful piece of information in itself.

Finally an extension of the framework allows for straightforward multi-target tracking. This extension is presented and further discussed in Section 7.6.

The development of this work was first presented in [35].

7.1 Observations drawn from Recorded Audio Data

A number of parameters of the SBF measurement function and of speech itself were studied in Chapter 5. In this section we will discuss how some of this information might be best integrated into the AST framework, while bearing in mind underlying performance limitations defined due to the frequency content of speech.

Firstly, in Section 5.2.2 we discussed the optimal density of SBF cells as a function of the range of frequencies used in the SBF integration. It was shown that the shape of peaks in the SBF surface is determined by the range of frequencies used to calculate the surface. A band of lower frequencies contribute to broader peaks while higher frequency ranges give sharper peaks.

In more specific terms, the maximum frequency of speech is in the range 4000-6000Hz. For the SBF, integrated up to this frequency, the SBF peak corresponding to a source will have a 3dB width of the order of 5-10cm (Figure 5.9). However should a lower range of frequencies be chosen this peak will broaden.

Secondly, as previously observed [60], speech is a highly non-stationary signal whose frequency content and activity varies widely from one frame to the next. Typically the SBF trace for a speech source will consist of a sequence of useful measurements followed by a sequence of silent or corrupted frames containing no useful measurement. This behaviour is difficult to model as each syllable, word or sentence can vary in length from speaker to speaker and from utterance to utterance. Instead, we will tackle this problem with a data-reactive Markovian activity detector in Section 7.4.3.

Finally interference between two sources simultaneously active in the same acoustic field was discussed in Section 5.2.5. It was observed that the frequency

of useful measurements is greatly reduced when compared to the single source scenario. This is due to signal-to-signal interference. We will adjust the tracking parameters of the multi-target tracking extension proposed in Section 7.6 to account for this; a more explicit solution would be to perform some form of source separation at the outset to suppress other active sources.

7.2 Accommodating Physical Observations

Classical approaches to tracking typically involve an initial step in which a small number of useful position measurements are extracted from the sensor output (e.g. from raw radar scans) using sensor signal processing. However this step usually requires a thresholding process; which as well as often being subjective, leads to a loss of information and limits the generality of the tracking algorithm. While this initial step is often unavoidable in military tracking (for example due to the necessity to anonymise data capture), we have full control of the data collection process in this algorithm.

Secondly for AST, this step would require us to calculate the SBF surface in the full region of interest and to then determine possible candidate peaks from the surface. To calculate this function at a sufficient density of points to guarantee the observation of all candidate source peaks (using the full frequency range of interest) is computationally prohibitive [63, 93].

However it has been noted that it is possible to avoid part of this computation by recognising the behaviour of the measurement function. In Section 4.3.2, the strategy proposed by Lehmann and Williamson [63] was discussed. This work first considered the frequency range used in the evaluation of the SBF function. The authors proposed limiting the frequency range to a small band of low frequencies. This allows the evaluation of the entire surveillance region at a low density (which can then be normalised). This surface is then used to provide particle proposals for an importance sampling algorithm. Later a continuous SBF surface is used to evaluate the particle likelihoods. While the first step has correctly considered the underlying shape of the SBF sur-

face, the second step carries out SBF evaluations despite some particles being, perhaps, fractions of a centimetre away from one another.

If particles are closely positioned and the gradient of SBF surface is constrained by the frequency content of speech, it may be considered unwise to persist in calculating likelihoods what can only be very marginally different. Instead we propose to recognise this limitation and to reduce the resolution of the evaluated SBF surface.

We propose evaluating the likelihood function on the points of a grid which have a carefully chosen density. As a result the likelihood function of neighbouring particles need only be calculated once (and shared for each). The overall number of SBF evaluations required falls dramatically. This alternative approach is introduced in the following sections, drawing on the Track Before Detect (TBD) framework [7, 82].

7.3 Proposed Framework

Firstly, we will maintain the original state variable and naming framework¹ outlined in Section 4.2. To the state variable vector we will append a source activity indicator parameter, λ_k . The state variable vector will thus become

$$\alpha_k \triangleq (x_k, \dot{x}_k, y_k, \dot{y}_k, \lambda_k). \quad (7.1)$$

This parameter, introduced in Section 7.4.3, is a binary variable with states corresponding to

- Source is currently active: $\lambda_k = 1$.
- Source is currently inactive: $\lambda_k = 0$.

7.3.1 Source Dynamical Model

As before, the dynamical model will be the Langevin model discussed in Section 4.2.2. However in subsequent sections the tracking algorithm will be extended to track more than one source and to this end a modification will be

¹ The modifications proposed for the orientation estimation in the previous chapter will be put aside for now.

introduced to the dynamical model, in Section 7.6.3, which adds a repulsive force to a pair of sources should they drift close to one another.

7.4 Track Before Detect

In the field of Electro-Optical sensor-based tracking, it is assumed that at each time step k , a pixel grid of IJ resolution cells is read simultaneously and that an individual pixel (i, j) has an intensity of $z_{ij}(k)$. The complete sensor measurement is denoted

$$\mathbf{Z}(k) = \{z_{ij}(k) : i = 1, \dots, I, j = 1, \dots, J\}. \quad (7.2)$$

Furthermore if a target is present it may only influence the pixel measurement² in which it is located. As a result the likelihood can be represented as

$$\begin{aligned} p(\mathbf{Z}|\alpha) &= \prod_{i,j} p(z_{ij}|\alpha) \\ &= \prod_{i,j \in C(\alpha)} p_{S+N}(z_{ij}) \prod_{i,j \notin C(\alpha)} p_N(z_{ij}) \end{aligned} \quad (7.3)$$

where $C(\alpha)$ is the set of subscripts of pixels affected by the target (with state vector α) and $p_N(\cdot)$ and $p_{S+N}(\cdot)$ are the likelihood functions for pixels in noise and a combination of signal and noise respectively. Using the particle filter technique the update stage of the filter is achieved using weighted resampling in proportion to the particle likelihoods. The resampling weights are thus $q(\mathbf{Z}|\alpha) \propto p(\mathbf{Z}|\alpha)$. However because this weight need only be evaluated up to a scaling factor the likelihood function can be divided by $\prod_{i,j} p_N(z_{ij}|x, y)$ giving a likelihood ratio

$$q(\mathbf{Z}|\alpha) \propto \begin{cases} \prod_{i,j \in C(\alpha)} l(z_{ij}) & \text{for } \lambda = 1 \\ 1 & \text{for } \lambda = 0 \end{cases} \quad (7.4)$$

where

$$l(z_{ij}) = \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})}. \quad (7.5)$$

² More accurate sensor models may allow the target to contribute to more than one pixel, however this possibility is not considered here.

This key step means that this likelihood ratio need only be calculated for the individual pixel in which the particle is located if using a single pixel model (or for the set of pixels located in the immediate vicinity of the particle, if using a more defined pixel model).

Moreover, when tracking accurately the particles will typically form a tight cluster around the true source location. This means that the measurement value for a particular pixel, z_{ij} , may be shared by all the particles located within that pixel. This leads to a dramatic computation reduction as the SBF calculation is the computationally intensive step in this algorithm, since each likelihood only needs to be evaluated once and then stored. The benefit of this is studied and discussed in Section 7.9. Finally, resampling of the particles is then carried out to give a final weighting for this iteration.

7.4.1 Adapting TBD to Acoustic Source Tracking

We will now adapt this framework and apply it to the AST problem. Initially, we will assume that only the discrete cell of the SBF pixel in which the source is located is affected by the source's speech — the central assumption of TBD. However, the entire acoustic field within a room is affected by the activity of a human speaker. Nonetheless, all AST algorithms have made the broad assumption that SBF or GCC correlation at a specific location is a quantitative measure of the likelihood of the source being located there. Note that the SBF is a continuous function and can be evaluated at any continuous location.

As discussed in Chapter 5 and also in Section 7.1, an SBF grid density of 10cm is sufficient to observe the majority of promising peaks (although increasing this density may lead to more accurate results). The SBF grid will thus be discretised with this density for the remaining portion of this thesis.

In standard TBD, Equation 7.5 requires that the SBF values be normally distributed with known mean and variance statistics. However the actual range of the SBF values is not distributed in this manner. Fricative speech, for example, will typically have greater signal-to-signal correlation than that of

a similarly energetic voiced speech sample. As a result a non-linear mapping will be used to adjust the SBF values onto a more balanced range.

7.4.2 Magnitude Mapping

As proposed in the previous section, the particle likelihood function will be based on a measurement function calculated for a set of pixels rather than a continuous function. As such the measurement related to a particular particle is that of the point at the centre of the *pixel* in which it lies,

$$z(x_k, y_k) = z(i\Delta, j\Delta) \quad \begin{array}{l} \text{for } |i\Delta - x| < \Delta/2 \\ \text{and } |j\Delta - y| < \Delta/2. \end{array} \quad (7.6)$$

From the study of the behaviour of the SBF function in Section 5.2.2, it was noted that for a particular recording environment and experimental setup the SBF function results in distributions of signal-and-noise and noise-only measurements with different variance statistics. In an attempt to better define the measurement function we apply a nonlinear mapping to the SBF values, $\mathcal{S}(x_k, y_k)$, as follows

$$z(x_k, y_k) = \Phi(\mathcal{S}(x_k, y_k); \bar{\mathcal{S}}, \sigma_{\mathcal{S}}^2) \quad (7.7)$$

where Φ is a normal cumulative distribution function with mean $\bar{\mathcal{S}}$ and variance $\sigma_{\mathcal{S}}^2$. These parameters are calibrated in advance or online so that the distribution mean, $\bar{\mathcal{S}}$, lies between noise measurements and active measurements — without applying a hard threshold. The choice of parameters for our specific recording setup was studied in Section 5.2.1. As a result the measurements have been mapped onto the range $z \in \{0, 1\}$.

Following the framework proposed by Salmond and Birch [82] we shall assume that the background noise is modelled as a zero mean Gaussian with variance of σ_N^2 for all pixels (i, j) . This gives a noise-only likelihood function of $p_N(z_{ij}) = \mathcal{N}(z_{ij}; 0, \sigma_N^2)$. If however the source is located within the grid pixel the signal-plus-noise likelihood function will be $p_{S+N} = \mathcal{N}(z_{ij}; I, \sigma_{S+N}^2)$ where I is the intensity due to the source. Using the above mapping, the intensity value, I , will be set to unity.

Truncation Constant

Because the measurement range is now truncated with limits of $[0, 1]$, it is necessary to introduce a truncation constant as follows

$$\begin{aligned} l(z_{ij}) &= \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} \\ &= \frac{c_{S+N}\mathcal{N}(z_{ij}; 1, \sigma_{S+N}^2)}{c_N\mathcal{N}(z_{ij}; 0, \sigma_N^2)} \end{aligned} \quad (7.8)$$

The truncation constant for the normal distribution, $N(z_{ij}, 0, \sigma_N^2)$, used to evaluate the noise likelihood function will be:

$$\begin{aligned} c_N &= \left[\int_0^1 p_N(z_{ij}) dz \right]^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{T_{max} - T_{mean}}{\sqrt{2}\sigma_N} \right] - \operatorname{erf} \left[\frac{T_{min} - T_{mean}}{\sqrt{2}\sigma_N} \right] \right)^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{1 - 0}{\sqrt{2}\sigma_N} \right] - \operatorname{erf} \left[\frac{0 - 0}{\sqrt{2}\sigma_N} \right] \right)^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{1}{\sqrt{2}\sigma_N} \right] \right)^{-1} \end{aligned}$$

Meanwhile the truncation constant for the normal distribution, $N(z_{ij}, 1, \sigma_{S+N}^2)$, used to evaluate the signal and noise likelihood function will be:

$$\begin{aligned} c_{S+N} &= \left[\int_0^1 p_{S+N}(z_{ij}) dz \right]^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{T_{max} - T_{mean}}{\sqrt{2}\sigma_{S+N}} \right] - \operatorname{erf} \left[\frac{T_{min} - T_{mean}}{\sqrt{2}\sigma_{S+N}} \right] \right)^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{1 - 1}{\sqrt{2}\sigma_{S+N}} \right] - \operatorname{erf} \left[\frac{0 - 1}{\sqrt{2}\sigma_{S+N}} \right] \right)^{-1} \\ &= 2 \left(-\operatorname{erf} \left[\frac{-1}{\sqrt{2}\sigma_{S+N}} \right] \right)^{-1} \\ &= 2 \left(\operatorname{erf} \left[\frac{1}{\sqrt{2}\sigma_{S+N}} \right] \right)^{-1} \end{aligned}$$

These variances may be chosen to be non-identical and other forms of the likelihood function could be used instead. Modification of the formulation of the likelihood function, while not carried out in this chapter, is discussed and experimented with in Chapter 8.

Simplification for Identical Variances

Should both variances be equal, ($\sigma_{S+N} = \sigma_N$), these constants will cancel out. The likelihood ratio for a pixel will simplify considerably and may be stated as

$$l(z_{ij}) = \begin{cases} \exp \left[\frac{2z_{ij}-1}{2\sigma_N^2} \right] & \text{for } |i\Delta - x| < \Delta/2 \\ & \text{and } |j\Delta - y| < \Delta/2, \\ 1 & \text{otherwise.} \end{cases} \quad (7.9)$$

Example likelihood functions, as well as the steered response power mapping, are illustrated in Figure 7.1. The variance used in this figure, $\sigma_{S+N} = \sigma_N = 0.5$ was used in the experiments carried out in the remainder of this chapter.

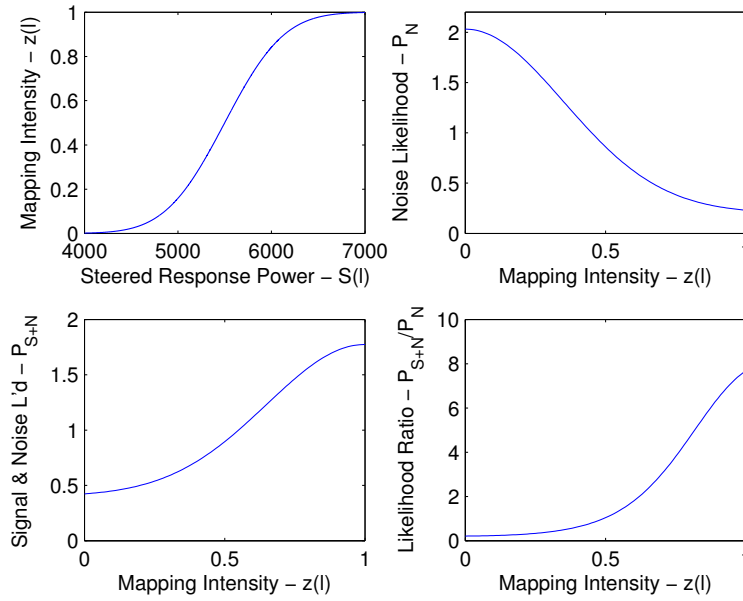


Fig. 7.1. Illustration of the functions used in calculating the likelihood ratio. First the raw steered response values are mapped onto the range $[0 - 1]$ using a normal CDF (upper-left). Then the likelihood ratio (lower-right) is then evaluated as a ratio of the noise-only likelihood function (upper-right) and the signal-and-noise likelihood function (lower left).

7.4.3 Activity Indicator Variable

As discussed previously [60], the temporally discontinuous nature of speech must be recognised to allow for a complete AST system. The authors then went onto introduce a model which uses a direct measure of voice activity as a parameter of the tracking system. Generally, when the source is deemed to be inactive, the particles were allowed to drift according to the dynamical model without recourse to the measurement data (the publication goes on to propose a number of different versions which soften the judgement of speaker inactivity). This approach is reasonable and the behaviour presented when the speaker became silent was as one would logically expect - a gradual increase in location estimate variance.

However the Voice Activity Detector proposed therein operated on the actual recorded speech signal rather than on the measurement function itself. This solution is one degree removed from the level of measurement we ideally require, whether or not the target can be observed by the SBF. Such a system will perform poorly should another source be active simultaneously in the room or if there was a loud noise for a short period elsewhere in the room for example. To counter these failings we propose instead to detect activity directly from the SBF function, while also integrating the proposed detection mechanism within the tracking algorithm.

As mentioned we add an activity indicator variable, $\lambda_k \in \{0,1\}$, to the state vector in a similar way to [82]. This variable will attempt to track the instantaneous activity of the source; that is if the source is *observable* via the measurement function at the time-frame in question. For single source operation it is typically anticipated that this will be broadly analogous to syllable-level activity estimation.

The parameter is not intended to determine overall longer term speaker activity - that is whether a speaker has *completely* stopped speaking or if an individual has begun speaking. An approach to solve that problem is proposed in Chapter 8.

The activity indicator variable will evolve according to a Markovian switching process with pre-determined transition probabilities — where

$$\text{Prob}\{\lambda_{k+1} = \lambda_a | \lambda_k = \lambda_b\} = [II(\lambda_{sk})]_{ab} \quad (7.10)$$

is the probability of a transition between states a and b . This is as was suggested in [82]. The optimisation of these parameters is discussed in Section 7.5.2. Following optimisation, we chose the probability of birth, $P_B = 0.05$, and the probability of death, $P_D = 0.05$ which was seen to perform well. This gives a transition probability matrix

$$[II(\lambda_k)]_{ab} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}. \quad (7.11)$$

Particles with an inactive state will drift via the dynamical model with the likelihood ratio set to unity, so that the final likelihood weighting function will become

$$q(\mathbf{Z}|\alpha) \propto \begin{cases} \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} & \text{for } \lambda = 1 \\ & \text{and } |i\Delta - x| < \Delta/2 \\ & \text{and } |j\Delta - y| < \Delta/2, \\ 1 & \text{otherwise} \end{cases} \quad (7.12)$$

when time index k is omitted for simplicity. When the target is actually speaking, the likelihood ratio, $l(z_{ij}|\lambda_k = 1)$, of particles deemed to be active will typically be greater than one. Meanwhile the ratio for particles deemed to be inactive is defined to be unity. In this way active particles will eventually proliferate upon resampling. In the opposite situation - when the speaker is silent, the likelihood ratio for active particles will be below one and they will be gradually removed.

Pseudo-code for the Track Before Detect AST algorithm is given in Algorithm 1.

Overall Source Activity Estimation

As each particle's activity variable discretely determines the source to be either active or inactive, the overall probability of activity of the source can simply be estimated as the proportion of active particles as

$$p(\lambda_{\text{overall}}|\mathbf{Z}_k) = \frac{\sum_{p=1}^{N_p} (\lambda_k^{(p)})}{N_p}. \quad (7.13)$$

This will allow us to track source activity directly — as distinct from signal energy activity (the typical VAD output).

Furthermore as this activity variable is dependent only on SBF activity in the region of the particle cluster (which coincides with the estimated source location), it is possible to track the activity of multiple sources simultaneously in different regions of the room — something that would not be possible with a generic voice activity detector.

Algorithm 1: *Track Before Detect Acoustic Source Tracking Algorithm*

```

for  $p \in \{1 : N_p\}$  do
    Predict  $\alpha_k^s$  using  $\alpha_{k-1}^s$ 
    Draw a new activity state,  $\alpha$ , using Eq 7.10
    if  $\lambda = 1$  then
        Located the grid cell containing the particle
        Evaluate  $q_k^{(p)}$  using Eq 7.12
    else
        Set  $q_k^{(p)} = 1$ 
    Update weight  $w_k^{(p)*}$  using  $w_{k-1}^{(p)}$ 
For each  $p$  set  $w_k^{(p)} = w_k^{(p)*} / \sum_{p=1}^{p=N_p} \{w_k^{(p)*}\}$ 
    Resample if necessary

```

7.5 Experimental Performance Testing

In this section, using recordings made with the experimental setup described in Chapter 2, the performance of the single target TBD acoustic source tracker is tested and described. Note that the source used was a computer loudspeaker transmitting typical conversational speech.

The metrics used in this section are explained in Appendix B.2.

First we will determine the stability of the proposed algorithm in increasingly difficult circumstances when compared with other particle filter strategies (Section 7.5.1). We will then go on to demonstrate the optimisation of some of the key parameters of the system (Section 7.5.2). Finally in Section

7.5.3 the algorithm is compared with existing algorithms using some common metrics.

7.5.1 Comparison with other algorithms

As mentioned in Chapter 4, a GCC-based measurement function fails to utilise all available signal-to-signal correlation information. This means that particle filter tracking will be unstable — particularly so for certain source positions, paths and recording scenarios. Figure 7.2 illustrates an example path in which the GCC measurements rely on only a single speaker pair; pair number 1, for the first half of the recording and then pair number 4. Because of this, the tracking algorithm becomes unstable as the frequency of useful location estimates from the other pairs is low.

To simulate increasingly challenging recording conditions, white noise³ was added to the recorded audio sample. The average original signal-to-added noise ratio of the samples used were as follows:

1. No noise added
2. 30dB (noise barely noticeable)
3. 20dB (noise becoming noticeable)
4. 10dB (noise level significant)
5. 5db (noise beginning to drown out speaker)
6. 0db (speaker slightly drowned out)
7. -5db (speaker substantiality drowned out)

For each test sample a particle filter was run using each of three measurement functions: the GCC, the SBF using Lehmann and Williamson’s *Pseudo-Likelihood* method [57] and the proposed Track-Before Detect SBF. Each filter utilised identical dynamical model settings, resampling schemes and 250 particles.

Figure 7.3 illustrates the performance of the algorithms for each scenario, averaged over 50 Monte Carlo simulations. Although the SBF frameworks con-

³ The implementation of this test with gradually increasing *reverberation* would, of course, have been more insightful but a varechoic chamber was not available.

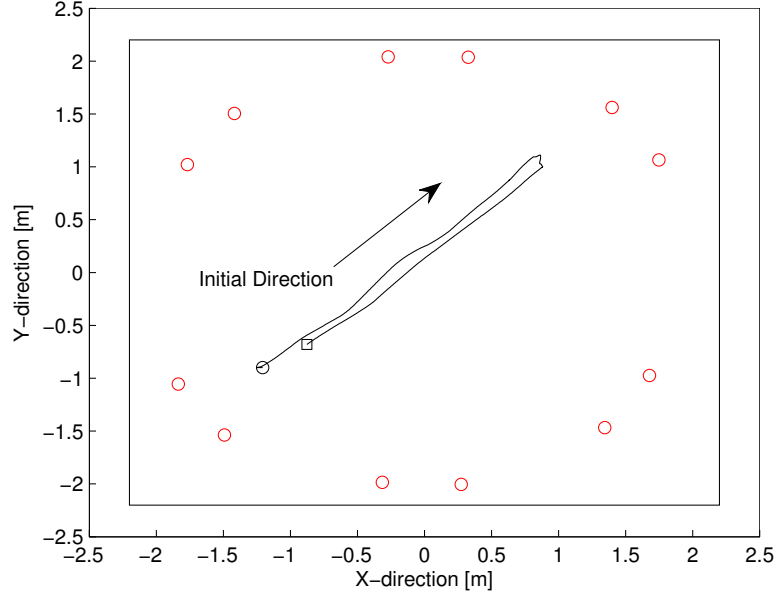


Fig. 7.2. Path taken by a source moving in a room. Recorded data used to evaluate tracking performance in Section 7.3. Note that the speaker begins and ends in the lower left corner and always faces in the direction it is moving. The microphone pairs mentioned in Section 7.5.1 are numbered 1-6 from the upper left corner clockwise.

sistently out-perform⁴ the GCC version, it is the disastrous collapse suffered by the GCC framework when the added SNR falls from 5dB to 0dB that is of note here.

In these samples the secondary microphone pairs fail to provide sufficient bearing estimates to localise in 2 dimensions with the only useful measurements those from pair 1 (and later pair 4) — illustrating the deficiency of the GCC measurement model.

Comparison with other SBF Algorithms: While the performance of the TBD algorithm does marginally outperform the pseudo-likelihood algorithm, it must be mentioned that the parameters of neither algorithm was

⁴ The superior tracking accuracy afforded by the SBF measurement function has previously been identified [93] and was also discussed in Chapter 4.

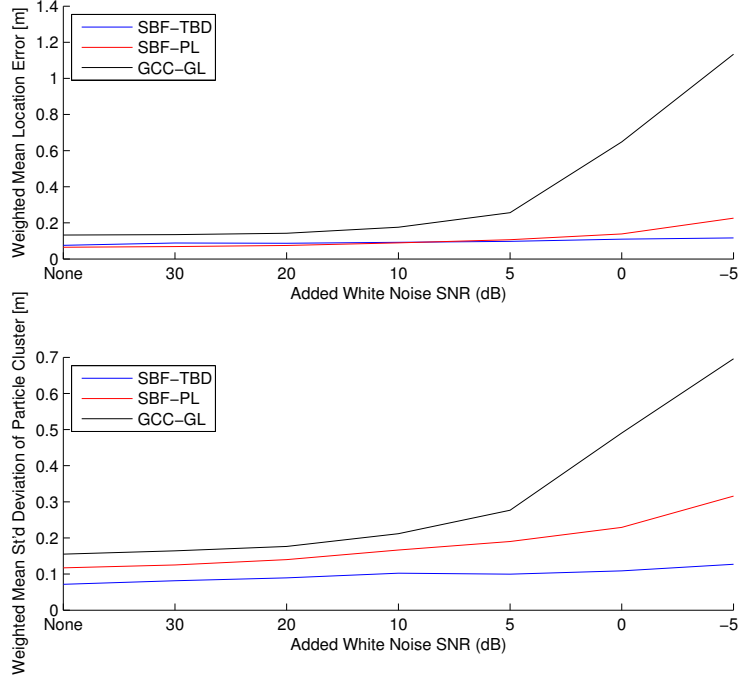


Fig. 7.3. Tracking Performance for three tracking algorithms tracking a speaker in increasing levels of added white noise. Note the collapse of the performance of GCC-based tracker when the SNR falls below 5dB.

specifically optimised for this audio sample. We now highlight algorithmic issues which illustrate the advantages of the proposed algorithm.

First by its nature the distribution proposed by the authors cannot be properly normalised [63]. This means the framework’s treatment of particle weightings in successive frames may not be entirely equitable. This issue has been addressed by the TBD method.

Second the correct implementation of the TBD algorithm allows for a large reduction in SBF computation. When multiple particles are located within the same TBD cell the SBF calculation need only be carried out once and stored for each particle. The computation time required for each of the algorithms (when implemented using MATLAB on a typical desktop PC) was as follows:

- Pseudo-likelihood SBF-based particle filter: 5.94 times real time

- Track-Before Detect SBF-based particle filter: 1.09-1.84 times real time
- (GCC-based particle filter: 1.3 times real-time)

Note that the computation required for the TBD particle filter is variable as the number of iterations will increase when the particle cloud becomes more diffuse, because more SBF likelihood evaluations are required. This occurs during pauses in speech activity and when there is greater speaker location uncertainty. This behaviour is illustrated in Figure 7.4 and in the following section the apt choice of activity variables is shown to remove this instability.

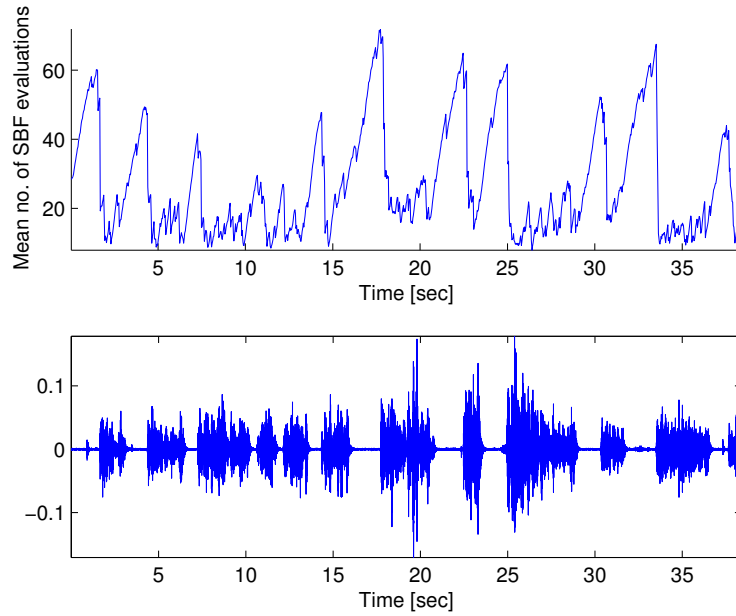


Fig. 7.4. Figure showing the variation in the number of SBF evaluations (upper plot) carried out using the TBD framework over the course of a speech sample (lower plot). Note how the number of evaluations required grows during silent periods.

Regardless of the computation required, to maintain stability of such a tracking algorithm during an extended speaker pause occurs requires a large particle cluster diffuse enough to explore all plausible regions of the state space so as to ensure the target's eventual detection. The Track-Before Detect

framework allows us to do this while balancing the computation of the overall algorithm.

7.5.2 Variable Optimisation

In the following section the optimisation of two of the more important parameters of the TBD algorithm is illustrated.

Density of the SBF Track Before Detect Grid

Varying the density of the SBF TBD grid affects the particle filter tracking accuracy as well as the required computation. In this section the optimal grid density is sought - while balancing these constraints. The experiment is related to but distinct from the experiment in Section 5.2.2. In Section 5.2.2 the minimum grid density required so as to observing the full set of SBF surface peaks was examined for different frequency ranges. No particle filter was implemented.

Instead, the experiment presented in this section a particle filter tracked the speaker path illustrated in Figure 7.8 (left). The results for a set of simulations are presented in Figure 7.5. Each data point is the average of 50 simulation runs, each tested on 30 seconds of data.

The mean estimated source location error (upper plot) was evaluated while varying the SBF TBD grid density. Meanwhile, the lower plot illustrates the mean number of SBF evaluations, per frame, required at each grid density.

The mean location error for the particle filter falls as the cell size is reduced. However it can be seen that there is little improvement gained by reducing the cell size below 10cm. Further, the number of SBF evaluations (and the associated increase in computational expense) required when the density is increased beyond this range quickly increases beyond this. As such this cell size, 10–25cm, represents a *sweet spot* in which accuracy of the particle filter and the computational demand of the beamformer are balanced.

The number of particles used in these experiments was 250. This means that for a grid density of 0.1m, on average, each steered response value was shared across 23 different particles — yet evaluated just once. When the grid

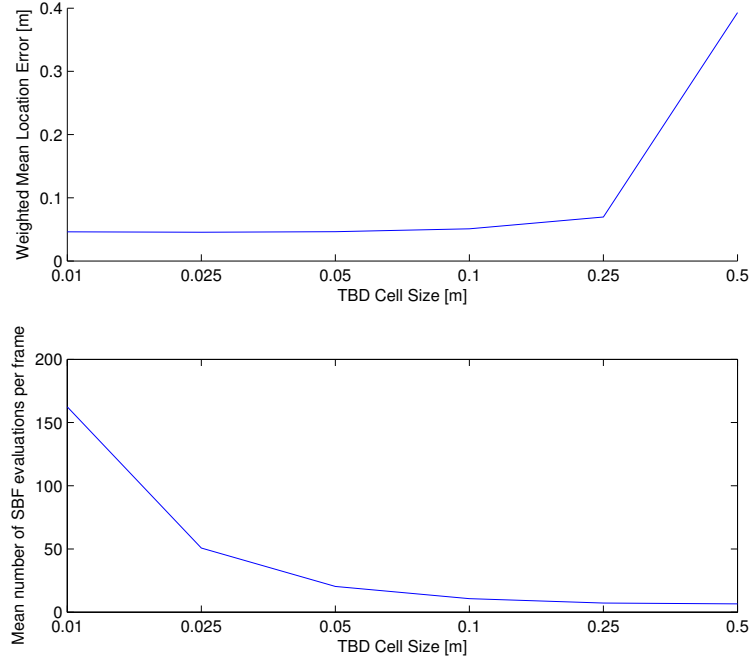


Fig. 7.5. Figure showing the effect of varying the density of the Track-Before Detect grid cells. See Section 7.5.2 for more details.

density is set to 0.01m each evaluation was shared across an average of just 2 particles — a huge reduction in efficiency. The cell size of the TBD SBF grid is set to 0.1m for all other simulations in this thesis.

Finally it must be noted that these tests were carried out using a relatively noise-free sample containing no speech pauses. The typical number of iterations for a particular grid density is likely to increase in more challenging conditions, as the particle cluster will typically be more more diffuse.

The mean location error for the highest grid density - approximately 0.05m - gives an indication of the upper bound on the performance of this localisation algorithm.

Parameters of Activity Indicator and Stability during Speaker Silence

Illustrative Example: Speaker with Silent Pauses

As mentioned previously the activity indicator allows us to determine directly the activity of a speaker from the particle filter behaviour. Knowledge of the speaker activity is useful both as an algorithm output but also in improving computational stability during speaker silence. In this section the main parameters of this system — the birth and death probabilities, P_b and P_d , are experimented with.

Figure 7.6 illustrates a speaker moving in a room. The speaker is silent on two occasions — between 5–9 seconds and between 18–23 seconds (with its location during silence indicated by dotted lines). The particle filter tracking the speaker estimates the source location correctly during speaker activity, as would be expected. However, note how the uncertainty of the X and Y position estimates grow during sections of speaker inactivity — symbolising increasing speaker location uncertainty — before renewing accurate tracking when the speaker continues.

Previous filtering algorithms required a sufficiently large particle set to adequately sample the entire surrounding region during this silent pause. Instead, using the TBD approach, if a particle is proposed to be inactive, using the Markovian birth/death process detailed in Section 7.4.3 then the associated target likelihood ratio is *defined to be unity* (Equation 7.12). This means that the set of currently inactive particles will have little effect on computation.

For example: if P_b had been set to 0.1, at each time frame only 10% of inactive particles will become active and the particle likelihood ratio determined via an SBF evaluation. The remaining inactive particles will not require one. This means that prolonged speaker inactivity need not lead to hugely increased computation — if properly recognised and quantified. The ability of the system to recognise speaker activity quickly is guaranteed by the typical frame-rate of this system (ΔT) — approximately 30 frames per second. This

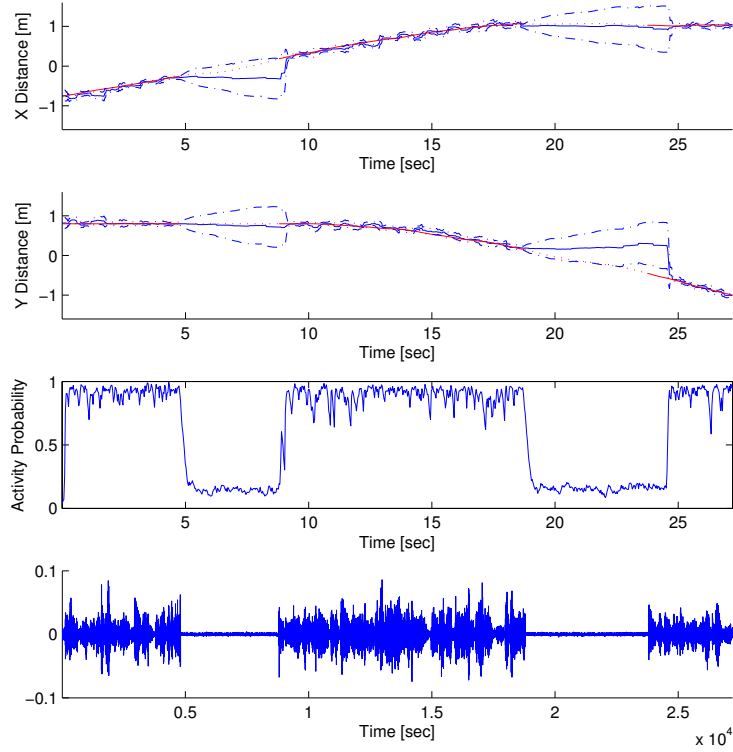


Fig. 7.6. Example of single target tracking with speech pauses. Tracking performance in the X and Y directions is shown in upper and centre-top figures respectively. The correct path is shown a red line (solid when active, dotted when silent), the particle filter mean location estimate with a solid blue line and estimate variance bars are shown either side of the estimate in dashed blue. Note how the variance of the position estimate increases during the two periods of extended silence. The lower centre plot shows the evolution of the overall probability of activity, $p(\lambda_{\text{overall}})$, while the bottom plot is of one of the recorded speech signals.

means that if $P_b = P_d = 0.1$, for example, each particle will sample the SBF surface roughly 3 times per second during inactivity.

Optimisation of Activity Switching Parameters

So as to optimise these switching parameters, P_b and P_d , the speech sample illustrated in Figure 7.6 was tracked while varying the probabilities of birth and death. Figure 7.7 represents the results of 50 runs of the algorithm, each with 250 particles.

The time taken for the filter to recognise that the speaker has resumed speaking (after a pause) and to resume accurate tracking is illustrated in the upper plot (for the two different speech pauses, as illustrated in Figure 7.6). For $P_b = P_d = 0.025$, this shows that after approximately 0.6 seconds the filter has resumed typically accurate source tracking. The performance for this and larger settings of the parameters are deemed to be reasonable. Below this value however, the filter struggles to recognise speech resumption due the domination by the inactive particles of the few active particles (note that no measurements are presented for $P_b = P_d = 0.001$ as for each speech pause the particle filter completely failed to resume tracking after the silence).

However, the responsiveness of the system for large values of P_b and P_d must be traded against increased SBF computation - due to a more diffuse particle cluster. The centre plot illustrates that for larger activity parameter setting ($(P_b; P_d) > 0.05$) extra SBF computation is required when the speaker is silent (red and blue) than when it is active (black). For P_b and P_d in the region of 0.025–0.05 it can be seen that the filter is computationally stable regardless of speaker activity. Below this inactive tracking is actually less demanding than active tracking.

The lower plot illustrates, over the entire speech sample — including both active and inactive periods, the percentage of iterations in which the speaker activity was correctly labelled. Because of the poor responsiveness of filters with a lower setting of P_b and P_d , a significant proportion of the frames are mislabelled. For $P_b = P_d = 0.5$, the activity variable is seen to switch randomly between activity and inactivity - as would be expected. Again the best performance is seen in the mid-range — with the correct labelling of upwards of 95% of frames.

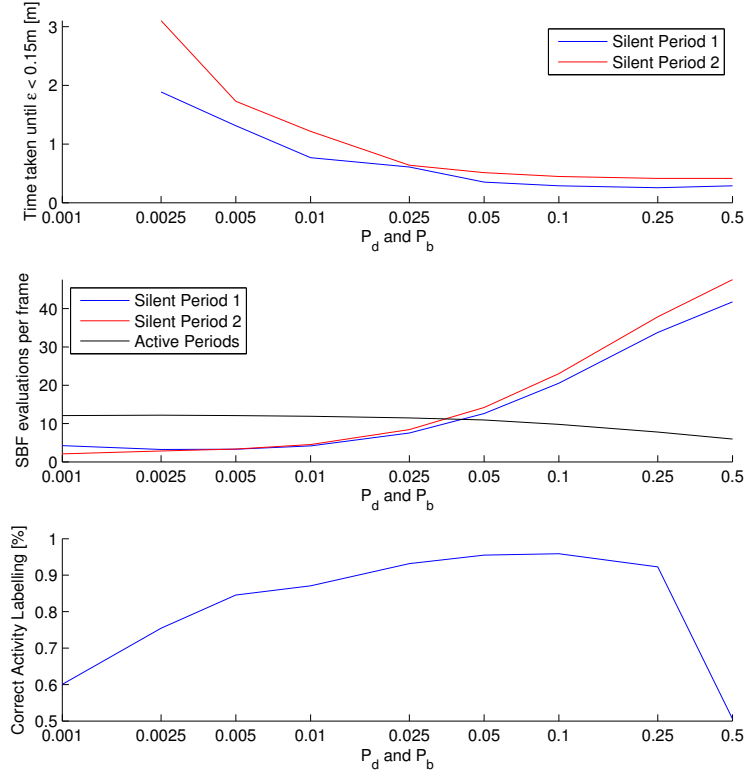


Fig. 7.7. The effect that varying P_b and P_d has on the ability for the TBD filter to correct itself after a period of speaker silence (middle) and the amount of computation required during such a silence when compared to what is typically required during typical active tracking (upper). Also shown is the percentage of correct activity labellings over the entire tracking segment. The x-axis of each plot has a logarithmic scale. See Section 7.5.2 for more details.

For the remainder of this thesis the switching parameters were set to be $P_b = P_d = 0.05$. Note that the optimality of this parameter choice is affected by the duration and frequency of speech pauses as well as the level of background noise and reverberation. These parameters have not been experimented with here. Furthermore no logical argument has been proposed to vary P_b versus P_d .

Note that in principle one could estimate the parameters P_b and P_d by ML or Bayesian methods, this has not been done here.

Finally referring to Figure 7.6 once more, the mean source activity variable (lower-middle plot) is seen to accurately determine activity (when compared to one of the recorded speech signals - bottom plot), with some reactionary delay.

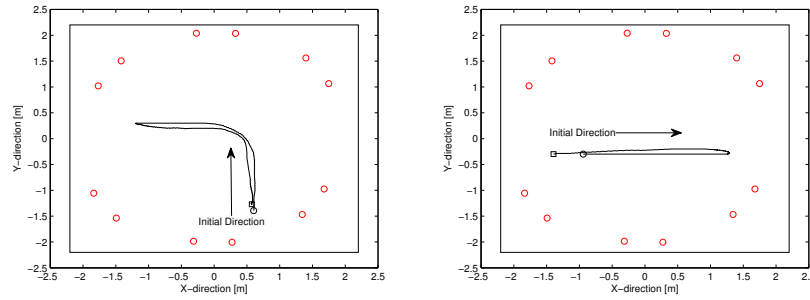


Fig. 7.8. Paths taken by the sources tested in Section 7.5.3. Example 1 is to the left, while example 2 is to the right. Note that in each case the targets double back upon the original path and return to the original location. Performance is not shown.

7.5.3 Monte Carlo Simulation Results

A final set of tests of this single source TBD algorithm was carried out against some common AST metrics. See Appendix B for more details of these metrics. The results presented in Table 7.1 provide a comparison between the performance of the proposed SBF Track Before Detect algorithm and the GCC-based particle filter, [90], as well the Pseudo-Likelihood and Gaussian Likelihood SBF-based particle filters, [93]. The paths of the sources are illustrated in Figure 7.8. Performance is measured both in terms of mean error, $\bar{\epsilon}$, mean standard deviation of the particle cluster (MSTD) and a measure of the percentage of tracks which fail completely (Track Loss Percentage, TLP). All of these parameters are explained in Appendix B.2. Parameters of the dynamical model and other common system settings were set equal in each algorithm, while parameters unique to a particular algorithm follow those

Method	$\bar{\epsilon}$ (m)	MSTD (m)	TLP	N_p	Time (sec)
Example 1 — 60.8sec					
GCC-GL	0.076	0.109	0	100	74.73
SBF-PL	0.073	0.310	0	100	126.64
SBF-GL	0.105	0.359	0	100	560.67
SBF-TBD	0.083	0.061	0	1000	73.09
Example 2 — 33.6sec					
GCC-GL	0.088	0.128	0	100	35.27
SBF-PL	0.108	0.330	18	100	71.74
SBF-GL	0.125	0.353	22	100	311.14
SBF-TBD	0.108	0.053	0	1000	48.65

Table 7.1. *Comparative Results for GCC based bootstrap, SBF based bootstrap and SBF TBD particle filters tracking a single source. Each figure has been averaged for 50 algorithm runs*

quoted in their respective papers. Note that the particle numbers used for each of the algorithms vary in this experiment, but an effort have been made to equalise the algorithm runtime instead. 1000 particles have been used for the TBD algorithm but only 100 for the SBF-PL algorithm.

The average tracking error of the proposed algorithm is shown to be similar to that of each of the other algorithms. However, the purpose of illustrating this experimental result is not to identify the superior performance of the TBD filter but rather to illustrate that it gives similar performance to previous methods while increasing stability. The MSTD for the TBD filter, indicative of tracking stability, is substantially lower than for the other algorithms because of an increase (by an order of magnitude) in the number of particles — this without the computation time increasing dramatically.

The computation time of the TBD algorithm remains reasonable because of the vast reduction in the proportion of likelihoods that need be calculated using the SBF-TBD. It is anticipated that the SBF-TBD with thousands of particles will comfortably run in real-time on a typical modern computer.

7.6 Multi Target Acoustic Source Tracking

The modification of single target AST algorithms to track more than one simultaneously active target would seem at first glance to be a simple and logical extension. However the acoustic field within a typical room is affected by each sound source's activity - something that is not the case for a radar scanning system, for example. This means that signal-to-signal correlation — required to produce effective GCC or SBF function — is severely compromised. In turn this reduces the proportion of frames providing useful measurements — this before even considering the problem of source-to-measurement data association. This effect has been discussed more thoroughly in Section 5.2.5.

Multiple target acoustic source tracking using the GCC as a measurement function has been attempted by Ma et al., [64, 92]. The experiments carried out to test this algorithm's performance used signals simulated using the image method and assumed the speakers to be ideal point sources. Both of these simplifications are unrealistic and the resulting method is unlikely to successfully operate when using real recordings.

Furthermore the number of microphone pairs (4) used in the presented simulations is insufficient to provide adequate coverage of a typical room when using the GCC measurement function. As mentioned previously, at least two GCC angle estimates are necessary at all times to provide a 2-D location estimate. Given the effect of speaker orientation, approaching 10 pairs would be necessary to track two real sources using the GCC. But could increasing the number of microphones used provide a solution? This is unlikely.

The complexity caused by an unknown number of speakers regularly criss-crossing each other's path in each of the GCC functions while simultaneously fading in and out of activity during silence makes for a very difficult data association problem. Figure 7.10 illustrates this problem. For these reasons the SBF will instead be used as the measurement function.

7.6.1 Multi Target Track Before Detect

Multi-target TBD is a relatively new extension of the TBD methodology, [7]. According to the TBD methodology we have assumed that the source may

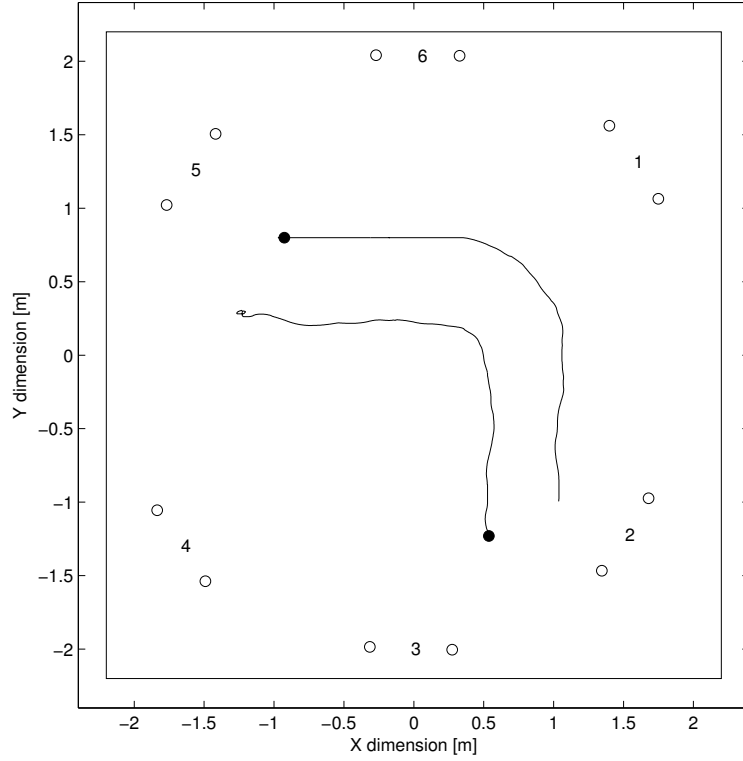


Fig. 7.9. Paths taken by two sources moving in a room. Illustrated also is the room boundary and the location of the microphone pairs. See Figure 7.10 for more details.

influence only the pixel value corresponding to the cell in which it is located (or the region surrounding the source location if smearing has occurred due to the sensor) hence, as Kreucher et al [54] suggest, we will consider the sources to behave independently when widely separated. Tracking in this scenario will be identical to the single source case in Section 7.4. Alternatively, when sources are closely spaced a joint likelihood will be considered. The transition between these two states is explained in Section 7.6.4.

In the general MTT literature the targets are assumed to be of a military nature — particularly airborne targets. As such there is a wealth of research in the field of extended target models which attempts to model targets travelling

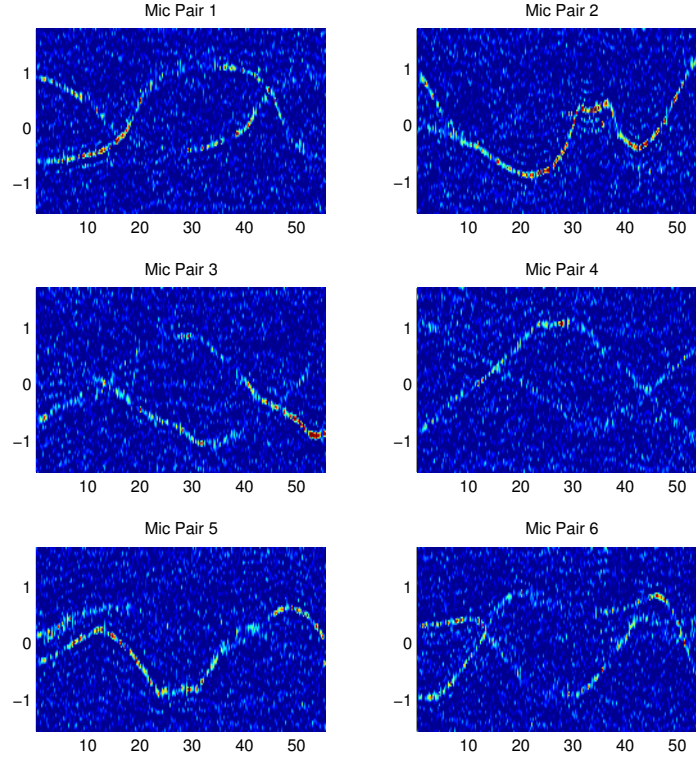


Fig. 7.10. *Illustration of the GCC delay paths for two moving sources. This figure shows the GCC delays that would be expected for each of the microphone pairs in the scenario seen in Figure 7.9. Each source begins at the circular marker and over 60 seconds moves around the room and back to the markers (following the same path). One complication is illustrated at 15 seconds when the target traces cross in 5 of the 6 microphone pairings - causing considerable target identification complication. A second complication is illustrated in the trace corresponding to microphone pair 2 (for the sample duration and in other microphone traces to a lesser extent). Because source two is facing away from this microphone pair it cannot be observed for the duration.*

as a group. One target can also be modelled as spawning a second — which would correspond to a missile launch for example.

None of this research is particularly applicable to AST. For instance, two human speakers moving in a room⁵ will not separate or coalesce - except in the most intimate of circumstances! To preclude this behaviour (within our algorithm) we will introduce a source-to-source repulsive effect for closely spaced targets. A joint particle state will then represent the sources' combined behaviour and track the sources jointly. These two scenarios — joint and disjoint tracking will explained in the following sections.

7.6.2 Joint Tracking: Widely spaced sources

First consider two widely separated sources. The sources will be considered to be independent of one another and will be considered as single individual targets. A state vector for the source s at time frame k will be

$$\alpha_k^s = (x_k^s, \dot{x}_k^s, y_k^s, \dot{y}_k^s, \lambda_k^s) \quad (7.14)$$

with an associated weighting w_k^s specific to that target track. As in the case of single source tracking, the generic dynamical model (Section 7.3.1) will be used as the transition prior in the prediction step.

Because the sources are widely separated, it will be assumed that only SBF pixels in the vicinity of the true source position will be affected by the source's speech signal. As a result, the likelihood ratio for source s will be identical to the single source case and evaluated in a similar way to Equation 7.12. The individual particle weights will be set to be $w_k^s \propto q_k^s$ in the same way also.

7.6.3 Tracking more than one closely positioned source

Instead consider a joint state vector for two sources located close to one another at time k ,

⁵ This work will concern itself only with a two source scenario. The possibility of extension to three or more sources is discussed in Section 7.9 but would require a further coding effort without affecting the core algorithm.

$$\begin{aligned}
\alpha_k &= (\alpha_k^1, \alpha_k^2) \\
\alpha_k^1 &= (x_1, y_1, \dot{x}_1, \dot{y}_1, \lambda_k^1) \\
\alpha_k^2 &= (x_2, y_2, \dot{x}_2, \dot{y}_2, \lambda_k^2),
\end{aligned} \tag{7.15}$$

with a single associated weighting w_k for the entire particle target cluster. As in the case of joint source tracking, the individual sources will be propagated according to the Langevin model, however we will modify the model subtly to disallow the coalescence of two speech sources.

Source Repulsion Mechanism

The distance between the two target positions can be obtained by

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{7.16}$$

with a relative angle between them of

$$\theta_{12} = \angle((x_1, y_1), (x_2, y_2)). \tag{7.17}$$

We shall propose that beyond a certain particle separation, $d_{12} > d_{\text{rep}}$, the sources are neither attracted to one another nor repulsed — we simply model two people moving independently around a room with the usual Langevin motion model of Section 7.3.1. However when sources become closer than this, $d_{12} \leq d_{\text{rep}}$, a repulsive effect will force them apart. This force is modelled as an accelerating force applied in the opposite relative direction of θ_{12} — much like a pair of polar equal magnets. A simple squared function has proven to work satisfactorily,

$$F_{\text{rep}}(\alpha_k) = \begin{cases} a_{\text{rep}}(d_{12} - d_{\text{rep}})^2 & \text{if } d_{12} \leq d_{\text{rep}} \\ 0 & \text{otherwise} \end{cases} \tag{7.18}$$

where a_{rep} and d_{rep} are constants chosen empirically to give reasonable behaviour, as illustrated in Figure 7.11. This force is then decomposed into its separate $\mathcal{X}$ and $\mathcal{Y}$ components, which for the first source is

$$\begin{aligned}
F_{\text{rep},x}^1(\alpha_k) &= \cos(\theta_{12})F_{\text{rep}}(\alpha_k) \\
F_{\text{rep},y}^1(\alpha_k) &= \sin(\theta_{12})F_{\text{rep}}(\alpha_k)
\end{aligned} \tag{7.19}$$

while the force applied to the second source is the equal opposite force

$$\begin{aligned} F_{\text{rep},x}^2(\alpha_k) &= -F_{\text{rep},x}^1(\alpha_k) \\ F_{\text{rep},y}^2(\alpha_k) &= -F_{\text{rep},y}^1(\alpha_k). \end{aligned} \quad (7.20)$$

See Figure 7.11 for a graphical illustration of the decomposition. These resultant vectors are added to the original dynamical model (in this case for the $\mathcal{X}$ -coordinate of source s)

$$\dot{x}_k^s = a_x \dot{x}_{k-1}^s + b_x F_x + F_{\text{rep},x}^s(\alpha_{k-1}) \quad (7.21)$$

$$x_k^s = x_{k-1}^s + dT \dot{x}_k^s. \quad (7.22)$$

Finally the likelihoods for each source position are again based on the SBF image pixel

$$q^s(\mathbf{Z}\alpha) \propto \begin{cases} 1 & \text{for } \lambda^s = 1 \\ \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} & \text{and } |i\Delta - x_s| < \Delta/2 \\ & \text{and } |j\Delta - y_s| < \Delta/2, \\ 1 & \text{otherwise.} \end{cases} \quad (7.23)$$

Following from Equation 7.4 and assuming that the sources may not occupy the same pixel cell, $C(\alpha^1) \cap C(\alpha^2) = \emptyset$, the product of the two likelihood ratios is calculated to give an overall likelihood ratio for the joint particle cluster

$$q(\alpha) = \prod_{s=1}^{N_s} q^s(\alpha). \quad (7.24)$$

Finally, as with the disjoint source case the particle set is normalised for each iteration and resampled when necessary.

7.6.4 Transition between tracking mechanisms

Transitions between the joint particle filter and two individual particle filter systems will be decided by a simple decision based on the MMSE estimate of the source particles and their variances⁶. While this may not be as accurate

⁶ Note: the transition between the joint and disjoint particle filters uses the distance between the MMSE estimates of *the entire particle set*. The unrelated repulsion mechanism, in Section 7.6.3, uses the distance between *individual* particles.

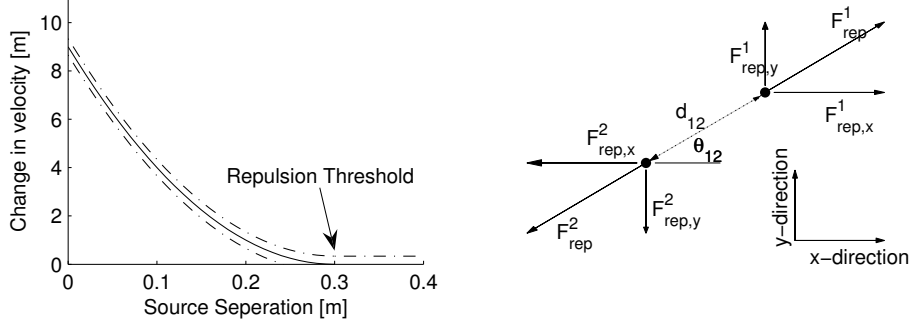


Fig. 7.11. *Left: Illustration of the repulsion effect: as source separation, d_{12} , falls below the threshold, d_{rep} , the force becomes increasingly significant. Right: The decomposition of the resultant force into dimensional components.*

an estimator as the Kullback-Leibler Divergence, for example, it has proven to be sufficient in practice.

The decision is taken according to

$$I_c = \begin{cases} 1 & \text{if } d_{12,\text{MMSE}} \leq d_{\text{thres}} \\ & \text{or } (d_{12,\text{MMSE}} - \sigma_1 - \sigma_2) \leq \sigma_{\text{thres}} \\ 0 & \text{otherwise.} \end{cases} \quad (7.25)$$

The integration of this state transition into the algorithm is detailed in Algorithm 2.

7.7 Multiple Target Tracking Results

An evaluation of the tracking performance of the algorithm is presented in this section. The test recordings were carried out using the same system described previously. Each source was generated individually by an audio signal played through a computer speaker and recorded by the microphone array. The signals were then linearly mixed before the MTT algorithm was run.

7.7.1 Illustrative Results

Figure 7.13 shows an illustration of the tracking performance for two different examples of two source tracking. The duration of the two samples, 32 seconds

Algorithm 2: *Switching Acoustic Source MTT Algorithm*

```

for  $p \in \{1 : N_p\}$  do
  Determine  $I_c$  using Eq 7.25
  if  $I_c = 1$  then
    Predict joint  $\alpha_k^s$  using  $\alpha_{k-1}^s$  and repulsion dynamical model
    Evaluate  $q_{s,k}^{(p)}$  using Eq 7.23
    Weight  $w_k^{(p)*}$  according to Eq 7.24
    For each  $p$  set  $w_k^{(p)} = w_k^{(p)*} / \sum_{p=1}^{p=N_p} \{w_k^{(p)*}\}$ 
    if  $D_{MAP} > D_{thres}$  then
      Set  $I_c \leftarrow 0$ 
      Divide State Vector into individual Source State Vectors
      Resample separately for each target  $w_{s,k}^{(p)}$ 
    else
      Resample if necessary
  else
    for  $s \in \{1 : N_s\}$  do
      Predict  $\alpha_k^s$  using previous sample set  $\alpha_{k-1}^s$  and dynamical model
      Weight  $w_{t,k}^{(p)*}$  according to Equation 7.12
      For each  $p$  set  $w_{s,k}^{*(p)} = w_{t,k}^{(p)*} / \sum_{p=1}^{p=N_p} \{w_{t,k}^{(p)*}\}$ 
    if  $D_{MAP} > D_{thres}$  then
      for  $s \in \{1 : N_s\}$  do
        Resample if necessary
    else
      Set  $I_c \leftarrow 1$ 
      for  $s \in \{1 : N_s\}$  do
        Resample
        Combine individual state spaces:  $\alpha_k^{(p)} \leftarrow [\alpha_{1,k}^{(p)}, \dots, \alpha_{N_s,k}^{(p)}]$ 

```

and 54 seconds respectively, which is long compared to what has been tested previously in the literature.

The particle filter is seen to track the two targets successfully. Note how the variance of the location estimate varies — particularly for Source 2 in Example 1 (upper plot). This coincides with a portion of audio in which Source 1 dominates the second source. Because Source 2 is unobservable the size of particle cluster (as represented by the uncertainty ellipses) will expand

to represent this uncertainty. This is similar to the algorithmic behaviour observed during a silent gap of a single source sample. When the target is observable once more, the particle filter returns to tracking accurately.

Example of sources passing closely

Figure 7.12 shows the track of two sources passing close to one another. Initially the two sources are disjoint and move freely — tracking their respective sources. In later frames the sources move close together and are considered jointly. The repulsion mechanism drives the sources apart and as a result the targets do not coalesce. After the sources pass each other and become disjoint the particle clouds return to tracking without any bias. Note also the increasing and decreasing size of the uncertainty ellipses which is, again, illustrative of the varying activity and inactivity of each source.

7.7.2 Performance Evaluation

Comparative results for the proposed tracking algorithm are presented in Table 7.2 using the two recordings in Figure 7.13. Source 1 in each case is a female speaker and Source 2 is a male speaker.

The results show that accurate tracking of two sources speaking simultaneously is successful and that the performance is only somewhat degraded when compared to the single source case — this despite the fact that dual source recordings will have a much lower proportion of useful peak measurements due to cross-signal interference. Finally the computation time is increased by about a factor of two, this is satisfactory. Using the TBD framework has allowed us to avoid the data association problem which is often computationally intensive.

7.8 Further Work

A number of issues are yet to be addressed for this algorithm.

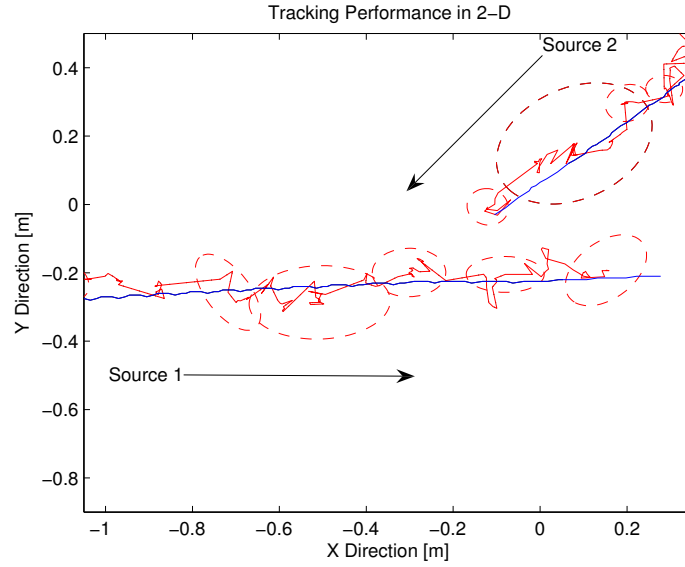


Fig. 7.12. Behaviour of two targets passing close to one another. The uncertainty ellipses are shown for every 25th frame. Note how the size of the particle clusters grow and shrink over time. This is because of the changing activity of the sources.

Source	$\bar{\epsilon}$ (m)	MSTD (m)	TLP (%)	N_p	Time (sec)
Example 1 — 32sec					
1	0.110	0.076	2	1000	92.14
2	0.114	0.118	10		
Example 2 — 54.4sec					
1	0.11937	0.11275	2	1000	184.66
2	0.108	0.330	4		

Table 7.2. Illustrative Results for the SBF TBD particle filter tracking two sources for the examples in Section 7.13. Each figure is the average for 50 algorithm runs

As mentioned there does exist the possibility of instability for extended silence. A high level algorithm component which halts tracking after extended silence would be required in such a scenario.

However more generally the assumption of source activity throughout the algorithm run needs to be removed. In the following chapter, an algorithm to handle initiation and removal of source tracks is introduced for the first time.

Secondly, as discussed by Salmond and Birch, [82], the performance of the TBD algorithm can be improved if the resolution of the measurement grid is improved so that the source can illuminate more than a single grid point (which would then be modelled as a scattered measurement). This could possibly increase tracking accuracy.

7.9 Conclusions

In this chapter a multi-target Track Before Detect algorithm has been proposed which can track multiple, simultaneously active speech sources.

The chapter began by discussing the characteristics of the SBF measurement function and the underlying measurement precision provided. This illustrated the computational inefficiency of tracking targets using a dense cloud of individual particles — each evaluating the measurement function at minutely different physical locations.

In subsequent sections an algorithm using the pixel-based TBD framework was introduced. This algorithm reduced the proportion of likelihoods which are typically calculated and allowed for a vast increase in the number of particles to be used without an increase in the computational effort. The tracking performance of the algorithm was shown to be similar to other single source AST algorithms.

Finally, a straightforward extension of the algorithm to track two sources was demonstrated. Performance for two source examples was seen to be only somewhat degraded when compared to single source scenarios and with only a factor of two increase in computation. Tracking stability for closely spaced targets was maintained using a novel repulsion mechanism.

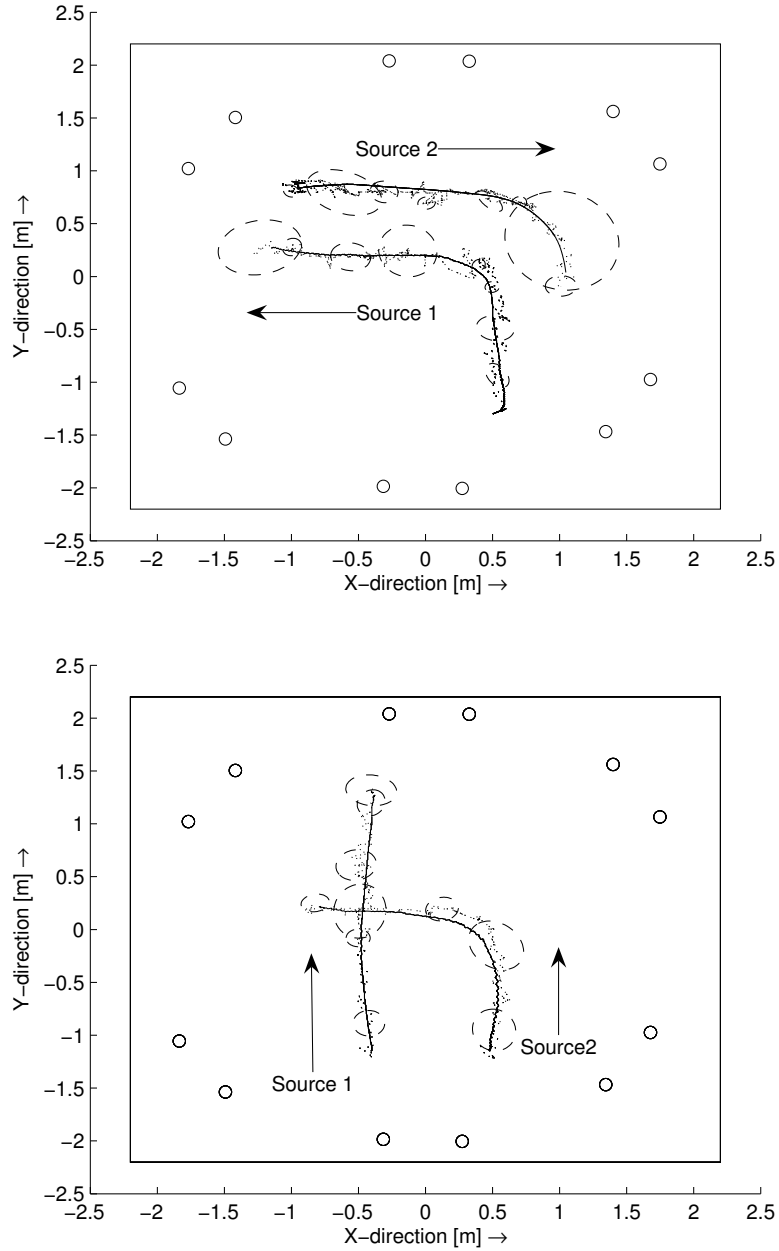


Fig. 7.13. Two sample recordings of two sources moving in a room, which were used to test the performance of the algorithm. The ground truth measurements are in black. An example of the tracking performance is overlaid on each plot (dotted black lines). Example 1 is the top plot. The microphone positions are indicated with circles. Uncertainty ellipses are shown every 100 frames.

Multiple Speaker Tracking for an unknown number of targets

As discussed in the preceding chapters, Acoustic Source Tracking has developed recently from tracking single-source recordings in synthetic environments [90], to tracking in real and challenging environments [93]. Furthermore these algorithms have typically assumed that the source(s) are active from the start of the algorithm and run to its end without any major silent pauses — which is obviously an over-idealisation. The technique proposed in Chapter 7 made similar assumptions also, although it extended the basic approaches by allowing for periods of speech inactivity and improved tracking stability.

In the following an entirely probabilistic strategy is proposed which identifies newly active sources, keeps track of them and removes them when they become inactive. Likely targets are proposed using the concept of an existence grid [67]. This process is carried out while also tracking the target locations using the Track Before Detect framework introduced in Chapter 7.

Before outlining this framework we will briefly discuss other approaches suggested to solve this problem and go on to comment on the performance the grid would ideally exhibit.

The development of this work was first presented in [36].

8.1 Tracking target activity

As discussed briefly in Sections 4.5.2, some ad-hoc approaches have attempted to deterministically identify active targets based around heuristic decisions and to then perform tracking. Sturim et al [86] first proposed a Kalman-filter

tracking solution of this form and a number of similar implementations using particle filter-based tracking have subsequently followed [51, 70, 71].

Alternatively Lehmann and Williamson [63] introduced an algorithm which allows for switching between conversational sources (i.e non-simultaneously active sources) when one of the speakers stops speaking. The system does not, however, attempt to determine if a source is actually active — with operation continuing regardless. When no target is active the particle cluster will simply spread out to cover the entire measurement space until a target becomes active. Furthermore it is not possible to easily adapt this system to function for more than one simultaneously active target.

Meanwhile within the general field of tracking a number of methodologies have been introduced to keep track of the number of active sources in a more principled manner. These include the Joint Probabilistic Data Association Filter (JPDAF, [4]) and Independent Partition Particle Filter (IPPF, [73]). These methodologies typically assume a military target returning position measurements with a constant signal to noise ratio. However acoustic targets are by their nature temporally discontinuous, giving rise to a dramatically varying SNR from one tracking iteration to the next for acoustic sources. It is obvious that a hybrid approach to the problem, which draws on elements from the generic tracking literature and allying them with observations from the measurement data is necessary. While the proposed method — based on the concept of an existence grid — deviates from the optimal MTT solution to some degree, it has however proven to be both stable and well-performing in practice.

An application of random finite set (RFS) theory to this field has been proposed [64, 92]. The measurement framework utilised a combination of GCC-based likelihood functions. However as discussed in Section 7.6, this measurement framework is unsuited to the problem at hand due to the complications caused by frequent target crossing in the GCC. This makes measurement-to-source assignments very difficult indeed.

Retaining the Steered Beamformer as the primary localisation function, in the following section we will propose a strategy to monitor regions of the

surveillance space for activity which are then used as the basis of a particle proposal mechanism.

8.2 Existence Grid

An important part of our particle filtering algorithm is an effective proposal mechanism for initiating new targets and deleting existing ones. An approach which does not include such a carefully designed data-dependent proposal mechanism is likely to suffer from poor exploration of the variable dimension target space. To achieve this goal we adopt an existence grid approach, based quite closely upon [67], but with likelihood functions carefully designed for our acoustic localisation framework. This proposal mechanism is incorporated into two tracking algorithms in the following sections, one of which is a fully Bayesian variable dimension tracking algorithm.

This existence grid is a low resolution grid overlayed on the surveillance region and updated at each iteration to reflect our belief in the existence of target(s) in each of the cells of the grid.

8.2.1 Design Choices

As mentioned previously, the Steered Beamformer function (SBF) provides an indirect measure of how likely it is that a speech sample originated at a region or location. The steered response power, evaluated at a particular location of this continuous function, is found by the integration of the cross-signal correlation over a defined range of frequencies. This allows for two free design parameters:

- The frequency range used for the integration — affecting the precision of the location estimates
- The set of locations evaluated — affecting the spatial extent of the evaluated surface.

Evaluating the entire surface using the full range of frequencies is of course impractical and a compromise is hence necessary. Evaluating the SBF function

using a low band of frequencies, in our case we have chosen $\Omega \in [100, 400]\text{Hz}$, reduces the peaked nature of the underlying surface, as discussed in [56, 63]. Figure 8.1 illustrates the SBF evaluated using two different ranges (on a full density grid). The low frequency version can be seen to give a broad estimate of the source location — limited by the wavelengths of the signals involved.

Because of this, the function can be implemented at a much lower density without aliasing occurring. As a result a low resolution grid, with in the region of 15-25 cells with cell dimensions in the order of 60-120cm across, can provide a coarse estimate of regional activity for the current frame of audio.

It is important to note that because of these two design choices the computational draw of this module is very small — especially when compared with the ensuing particle filter. Table 8.1 provides a comparison between the evaluation of this surface, compared to those required when evaluating the particle filter likelihoods and also those required for a full density SBF grid.

Using the Bayesian update framework discussed by Moreland et al [67], this grid of instantaneous values can be combined with previous data to give a posterior estimate of activity in each cell — the existence grid.

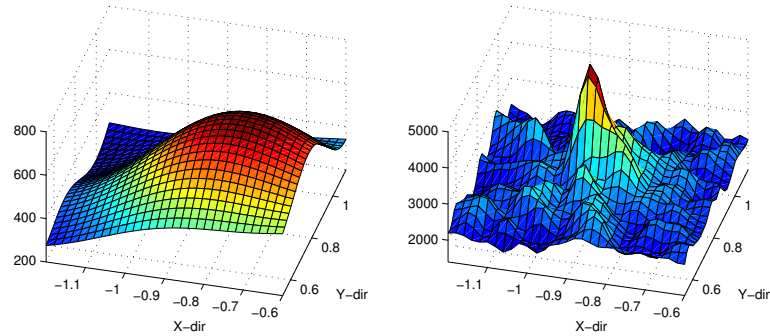


Fig. 8.1. Graphical comparison between SBF functions for the same audio frame. Left shows the SBF for the frequency range 100-400Hz, right shows the comparative surface integrated from 200-6000Hz. The true source position is at $[-0.92, 0.80]$ approximately. Note that the lower frequency surface need not have been evaluated at this high grid density (0.02m) to observe its main characteristics.

	Likelihood-SBF	Existence-SBF	Likelihood-SBF Grid
Band of frequencies used	200-6000 Hz	100-400 Hz	2000-6000 Hz
Typical 3dB peak width	5cm	50cm	5cm
Grid cell size	N-A	80cm	~5cm
Frequencies integrated	742	38	742
Total evaluations	~100	100	~6400
Relative Computation	~742×100	38×100	~742×6400
Evaluated in this algorithm	Yes	Yes	No

Table 8.1. *Computational Comparison between SBF Functions*

8.2.2 Desirable traits of the Existence Grid

Before implementing such a grid, we will first discuss the performance that would be expected and required from it. In determining source activity, we will choose to place higher weight on quickly finding newly active sources (ideally within a fraction of a second) than on quickly removing sources that have recently become inactive.

Furthermore, bearing in mind the measurements carried out in Section 5.2.5, we can typically expect in the region of 60% of frames to provide *active* and accurate location estimates (where we arbitrarily defined using a 6000 unit SRP threshold) when a single source is actually speaking (although that experiment was specific to the frequency range [200 – 400]Hz. Thus for a grid cell to gain considerable attention, we will require its existence value to become elevated after source activity within it for perhaps only a couple frames out of a possible 30 frames per second.

Conversely when the source becomes inactive or leaves a grid cell, the existence cell value will die away gradually over the course of a second or two — returning to the background level.

8.2.3 Evaluating the Grid

First consider a grid of J cells each of uniform size, $(\Delta x, \Delta y)$, spread across the surveillance region. The cells will be numbered $j = 1, \dots, J$. For each cell the SBF function, Equation 2.14, will be evaluated at the cell centre point.

The location of the centre of the grid cell will be $l_j = (x_j, y_j)$. The SBF will be integrated over the range of frequencies $\Omega \in [100, 400]\text{Hz}$.

Using the mapping presented in Section 7.4.2, these values are then transformed, using a normal CDF, onto the range $[0, 1]$

$$z_j = \Phi(\mathcal{S}(l_j); \bar{\mathcal{S}}, \sigma_{\mathcal{S}}^2) \quad (8.1)$$

where the mean and variance of the normal distribution will now to be set to be $\bar{\mathcal{S}} = 450$ and $\sigma_{\mathcal{S}} = 50$. These parameters are chosen after careful review of the data — such that typical measurements recorded in noise will be at the lower end of this range, while measurements for active sources will be at the upper end of the range. Using this measurement grid as an input, we will now update an existence grid across the surveillance region.

Moreland et al, [67], presented a two step Bayesian update rule for the estimation of the probability of a source existing in a particular cell (Equations 48 and 49 from their paper). First the previous estimate is updated using our prior information of how we expect the source activities to typically evolve:

$$g_j^{t|t-1} = g_j^{t-1}(1 - \zeta_j^t) + (1 - g_j^{t-1})\epsilon_j^t \quad (8.2)$$

For our application this could include information about locations in the room in which a source is more likely to be located, for example, or how long we expect someone to typically speak before pausing. Subsequently we update the existence values using information drawn from the current measurements

$$\begin{aligned} g_j^t &= \frac{g_j^{t|t-1} p(z^t | o_j^t = 1)}{p(z^t)} \\ &= \frac{g_j^{t|t-1} p(z^t | o_j^t = 1)}{g_j^{t|t-1} p(z^t | o_j^t = 1) + (1 - g_j^{t|t-1}) p(z^t | o_j^t = 0)}. \end{aligned} \quad (8.3)$$

which gives us the existence probabilities for the current iteration. Note that g_j^t is bounded within the range $[0, 1]$. A large value means a high chance of the existence of a target in that cell but it is not a probability, per say. To implement this update rule we need to formulate the likelihood functions $p(z_j | o_j = 1)$ and $p(z_j | o_j = 0)$.

8.2.4 Existence Grid Likelihood Functions

The likelihood functions we shall propose will simply be as follows, for cell j :

$$\begin{aligned} p(z_j | o_j^k = 1) &= c_1(\mathcal{N}(z_j; 1, \sigma_1) + q_1), \quad 0 < z_j < 1 \\ p(z_j | o_j^k = 0) &= c_0(\mathcal{N}(z_j; 0, \sigma_0) + q_0), \quad 0 < z_j < 1 \end{aligned} \quad (8.4)$$

where q_1 and q_0 allow some heavy-tailed behaviour in both active and inactive cases. c_0 and c_1 are the normalising constants necessary to normalise the probability density functions in the interval $[0, 1]$. z_j is the (CDF-transformed) low frequency steered response power evaluated at the centre of cell j . Variance and noise floor constants used herein are as follows, based on careful tuning to real datasets:

- Active Source: $\sigma_1 = 0.02$ and $q_1 = 7$
- Inactive Source: $\sigma_0 = 0.4$ and $q_0 = 40$.

Note the large difference between the variances used — which illustrates that an *active source measurement* is deemed to be much more informative than an *inactive source measurement*.

The whole procedure produces, at each time frame k and for each cell j , a probability g_j for activity of target(s) within that cell. These values, in association with the configuration of active targets within particles at the previous time frame, are used to propose target initiations and deletions within the particle filter, which is now described.

8.2.5 Target Proposal Mechanism

Having evaluated the existence grid values, next we find probabilities for adding a new target or removing an existing target which will be used to propose the particle set. In this work the removal and then addition processes will be carried out in sequence for simplicity, although it is possible to combine the steps.

Removal Probabilities:

Consider a particle at time $k-1$ made up of s^{k-1} targets, $\mathcal{A} = (\alpha_1, \dots, \alpha_{s^{k-1}})$, located in cells $(l_1, \dots, l_{s^{k-1}})$. Firstly the set of relative probabilities of a

particular target *not existing* in the cell l_s^k given this combination of targets, is found relative to the other existing targets using Equation 53 from [67]:

$$\tau_s^k = \beta_s^k \frac{1 - g_{l_s^k}^k}{1/s^{k-1} \sum_{r=1}^{s^{k-1}} (1 - g_{l_r^k}^k)}. \quad (8.5)$$

where β_s^k is prior target occupancy constant which can be used to reflect regions which are more or less likely to be occupied — but in what follows this was chosen to be a constant value. This set of values correspond to the existence grid values, g_j^k , and will now be used to draw removal proposals when there exists at least one target within a particle.

The probability of removing a specific target instance from the target combination is as follows

$$\kappa_s = \frac{\tau_s^k}{1 - \tau_s^k} \prod_{r=1}^{s^{t-1}} (1 - \tau_r^k) \quad (8.6)$$

with the probability of not removing any of these targets given by

$$\kappa_0 = \prod_{r=1}^{s^{t-1}} (1 - \tau_r^t). \quad (8.7)$$

The normalised set of removal probabilities, $\mathcal{K} = \{\bar{\kappa}_0, \dots, \bar{\kappa}_{s_{k-1}}\}$, are denoted where $\sum_{s=0:s_{k-1}} \bar{\kappa}_s = 1$. Thus $\bar{\kappa}_0$ is the probability of no target being removed while the sum of the remaining values represents the probability of any of the targets being removed.

Addition Probabilities:

The following addition step is carried out. The relative probability of adding a new target to a specific cell, j , will be the product of the probability of a target existing in that cell with each of the probabilities of a target *not* existing in each of the other (vacant) cells,

$$\nu_j = \frac{g_j^k}{1 - g_j^k} \prod_{i \in A^k} (1 - g_i^k) \quad (8.8)$$

and the probability of no target being added to any of the cells will be as follows

$$\nu_0 = \prod_{i \in A^t} (1 - g_i^t) \quad (8.9)$$

The normalised set of addition probabilities, $\mathcal{V} = \{\nu_0, \dots, \nu_J\}$, are denoted where $\sum_{j=0:J} \bar{\nu}_j = 1$. $\bar{\nu}_0$ is the probability of no target being added while the sum of the remaining values represents the probability of any of the target being added.

Please see Sections C.1 and 8.5.2 respectively in which the remaining details of the removal and addition processes for each of the proposed algorithms are presented.

These equations are again drawn from the afore mentioned publication [67] — but make the simplification that only a single target may become active or become inactive at a particular time frame. This simplifies the calculations involved but is not seen to be a major limitation of the algorithm given the update rate (expected to be approximately 30 frames per second) and the number of targets expected to be present (in the region of 2-3 speakers).

8.2.6 Existence Grid Optimisation

To optimise the parameter settings of the existence grid likelihood function a series of experiments were carried out. Each parameter was optimised individually while maintaining the other three parameters unchanged.

Each optimisation attempted to maximise a metric which aimed to quantify the difference between the existence grid value for the cell in which the source was located versus the existence grid values of the other cells in the surveillance region. The parameter was averaged over 45 seconds of speech for two different scenarios (i) a single source was speaking and (ii) two sources speaking simultaneously. An initial burn-in period was excluded.

Figure 8.3 illustrates the typical evolution of the existence grid values when a single source (upper plot) and two sources were speaking (lower plot). Presented in magenta and black is the evolution of the existence grid values of the sources. Presented in red is the maximum existence grid value of the cells in which no sources were located, while the dotted red line represents the minimum value of the existence grid.

The metric of interest was defined as the proportion of frames in which both of the following conditions are true:

$$\begin{aligned}
g_{t_s} &> 2 \max_i g_{t \neq t_s} \\
g_{t_s} &> 1.2 \text{ median}(\max_{t, t \neq t_s} g_t).
\end{aligned} \tag{8.10}$$

where t_s is the existence cell in which the source is located. While the choice of this particular metric is arbitrary, it has been chosen provide results which reflect the type of behaviour we would expect. Finally it should be recognised that the resultant parameter optimisations are, thus, optimal only with regard to this metric.

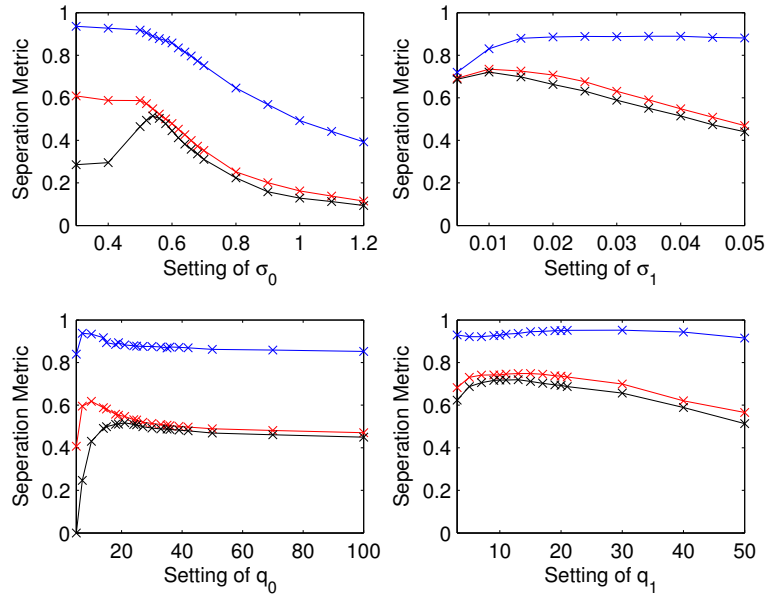


Fig. 8.2. Optimisation curves for the four parameters of the existence grid likelihood functions; σ_0 (upper left), σ_1 (upper right), q_0 (lower left) and q_1 (lower right). The parameter measured on the y-axis is the proportion of data frames in which the existence grid exceeded set of particular thresholds.

The experiments were carried out with the following settings

- $q_0 = 20$, $q_1 = 25$, $\sigma_1 = 0.04$ while varying σ_0
- $\sigma_0 = 0.54$, $q_0 = 20$, $q_1 = 25$ while varying σ_1
- $\sigma_1 = 0.0175$, $\sigma_0 = 0.54$, $q_1 = 25$ while varying q_0

- $q_0 = 25, \sigma_1 = 0.0175, \sigma_0 = 0.54$ while varying q_1

The results of the experiments are presented in Figure 8.2. The blue line represents the results for the single source sample while the red line represents the results for each of the two simultaneous speakers. It can be seen that each source exhibits similar performance (in that there is a single optimal peak setting) although the optimisation curve for σ_0 provides a less conclusive behaviour. The optimal choice of parameters was decided to be

- $\sigma_0 = 0.54$
- $\sigma_1 = 0.0175$
- $q_0 = 25$
- $q_1 = 15$

although it can be seen that performance is optimised for the two scenarios (one source and two sources) at slightly different settings. It is theorised that this is because there is lower frequency of measurement data for the two source sample when compared to the single source sample.

An illustration of the shape of the resultant optimal existence grid likelihood functions is presented in 8.3. As predicted in Section 8.2.2, the optimal functions highly reward measurements deemed to have originated from the source (modified SRP value ~ 1) while only mildly weights clutter measurements (modified SRP value ~ 0) as they are much less informative. Meanwhile Figure 8.4 illustrates the evolution of the existence function over time for one active source (upper) and two sources (lower) — illustrating that while the grid provides an indication of source activity it does so in a somewhat unreliable manner — this issue is returned to in the results section of this chapter.

8.3 Tracking Frameworks

This existence grid mechanism framework provides us with a logical mechanism for proposing particle sets to reflect the underlying activity of the different regions of the surveillance space. These particles sets can then be used to update the posterior distribution of the number of targets in the space and their positions.

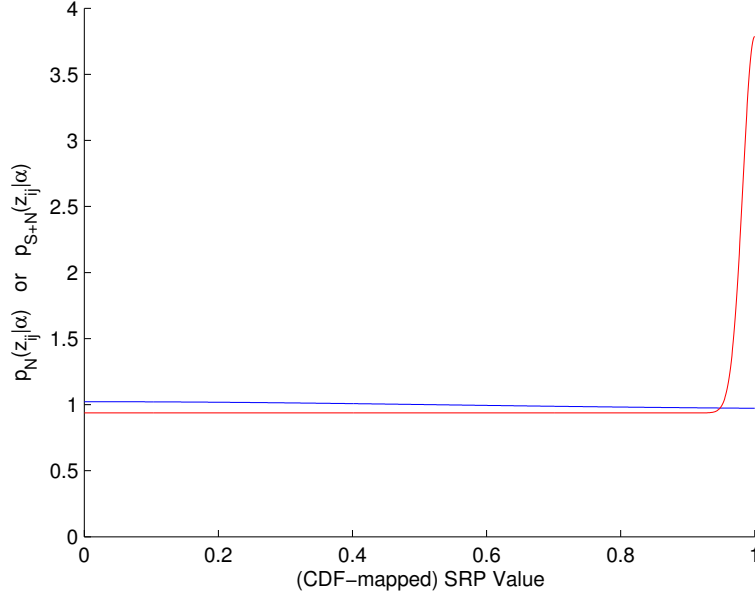


Fig. 8.3. *Optimal Existence Grid Likelihood Functions. The likelihood function for measurements of signal and noise are in red while those for noise only are in blue. Note the relative importance given to measurements deemed likely to indicate source activity.*

In the following sections two separate algorithms will be proposed to do this. The first algorithm will maintain distinct target-to-source associations and when the sources are widely separated will essentially operate distinct particle filters. However should sources move close to one another, complication will arise as to how a joint posterior can be formed. A compromise is proposed which makes the tracking framework suboptimal in this (unlikely) scenario.

The second algorithm proposed implements a variable dimension particle filter which attempts at all times to correctly determine the joint posterior distribution using a single state vector containing all targets. Because of the complicating factors of multi-target tracking this algorithm must be carefully implemented with foresight to recognise potential difficulties.

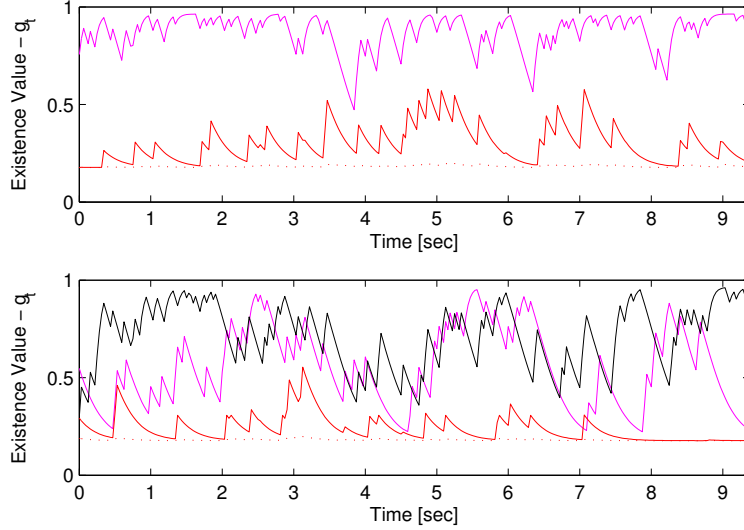


Fig. 8.4. *Illustration of the evolution of the existence grid values for a single source (upper plot) and for two sources speaking simultaneously (lower plot). See Section 8.2.6 for further explanation. The optimal parameters identified in that section were used here.*

Finally experimental results are later presented which demonstrate the strengths and weakness of the two algorithms.

8.4 Framework 1: Defined Target-Source Assignment

In this section a particle filter tracking algorithm which maintains a defined assignment for each target (state vector) incidence to a particular source track will be briefly outlined and discussed. The full details of this algorithm are presented in Appendix C and a summary of the algorithm is illustrated in Algorithm 3.

To maintain the same tracking framework presented in Chapter 7, the state vector maintains, at all times, a clear and distinct assignment of each target incidence to a particular hypothesised source. The target incidences are directly proposed from the existence grid and are assigned the target label of

Algorithm 3: *Defined Target-to-Source Tracks*

```

Update the existence grid
for  $p \in \{1 : N_p\}$  do
    Draw target removals according to (C.4)
    Evaluate if a new target should be added (C.5) and select a cell (C.6)
    if Particle  $p$  does not contain a target in the proposed cell then
        if Proposed cell contains an existing target track then
            Add target to currently existing track according to (C.9)
        else
            Add new target track to the tracking system
    Separate targets into clusters based on target separation
    Propagate retained targets using dynamical model
    for  $c \in \{1 : N_c\}$  do
        for  $p \in \{1 : N_p\}$  do
            Draw a new activity variable,  $q_{s,k}^{(p)}$ , for each target
            Evaluate the target likelihoods using (C.12)
        Update the overall weights  $w_k^{(p)*}$  of the cluster
        Resample the cluster set using the weights if necessary

```

the closest existing source. (If the proposed source is located far from all of the sources, a new target track will instead be initiated).

Having completed this step, there will exist a set of independent target tracks which can then be treated as independent particle filters, complete with weighting and resampling mechanisms. In this way the particle filter dimensionality can be reduced and the tracking system partitioned.

However complication arises when two sources move close to one another and can no longer be assumed independent. While a modification of the algorithm treats such sources jointly is presented in Section C.1.2, a correct reweighting mechanism was not developed.

Situations in which this approach provides successful tracking are presented in Section 8.6 followed by some more challenging unsuccessful approaches. As mentioned above, more details regarding this algorithm are presented in Appendix C.

Furthermore, as the target initiation and removal are directly proposed using the existence grid values the target tracks will be inherently unstable. This is illustrated and discussed in Section 8.6.

In the following section a full variable dimension particle filter is instead proposed in which each particle represents a single un-partitioned estimate of the underlying state space — an approach which develops a more right-minded solution to the problem. This filter will use the birth and death processes as a proposal mechanisms alone while also incorporating prior information to improve target stability. Furthermore, particles containing zero targets will have as legitimate a contribution to the framework as those containing one or more targets.

8.5 Framework 2: Undefined Target-Source Assignment

This tracking system will utilise a variable-dimension particle filter to keep track of the time-varying number of sources present in the room. The strategy is similar to the framework of [54, 72], combined with an existence grid-based target proposal method. The number of targets, S_k , within each individual particle may vary in the range $\{0, \dots, S_{\max}\}$, representing the number of speakers deemed to be active at any given time k . $S_{\max}$ is the maximum number of simultaneously active speakers and is chosen to be 3 in our experiments, although in principle the methods extend to more ‘crowded’ environments as well. An individual particle state-space, containing S_k targets at time k , is defined as follows

$$\mathcal{A}_k = (\alpha_k^1, \dots, \alpha_k^{S_k}, S_k) \quad (8.11)$$

with an associated particle weighting w_k . Each target, α_k^s , will contain position and velocity components in the $\mathcal{X}$ and $\mathcal{Y}$ -dimensions, as follows

$$\alpha_k^s = (x_k^s, y_k^s, \dot{x}_k^s, \dot{y}_k^s). \quad (8.12)$$

The aim of the particle filter is to update the posterior probability density for the entire vector given in Equation 8.11 using information drawn from the current measurement set, $\mathbf{Z}_k$. Note that the activity variable utilised previously has not been retained here.

8.5.1 Data Model

Once again the Langevin dynamical model will be retained for this algorithm which gives us a prior model for the state transition densities

$$p(\alpha_k^s | \alpha_{k-1}^s) = \mathcal{N}(\alpha_k^s; \mathbf{f}(\alpha_{k-1}^s), \sigma_{\mathbf{e}}^2) \quad (8.13)$$

where the system will once more be that presented in Section 4.2.2.

Target Number Transition

Within our framework we propose also to model the random appearance ('birth') and disappearance ('death') of speakers. Doing so explicitly is a departure from the method used in the previous algorithm in Section 8.4 where targets were initiated/deleted heuristically based on the existence grid. For simplicity we will assume that at most one target may appear or disappear at each time step. However, so as to simplify the implementation of the algorithm we will require that each of these processes will be applied in succession — removal followed by addition.

Firstly the removal process will provide a prior model of how the number of targets is likely to change given the possibility of removing a target

$$S_{k|k-1} = S_{k-1} + \epsilon_{k|k-1} \quad (8.14)$$

and it will do so with a prior removal probability distribution

$$p(S_{k|k-1} | S_{k-1}) = \begin{cases} \Pr(\epsilon_{k|k-1} = -1) = h_d \\ \Pr(\epsilon_{k|k-1} = 0) = 1 - h_d \end{cases} \quad (8.15)$$

where h_d is the probability of decrementing the number of targets.

Secondly the addition of a new target will be carried out in a similar way as follows

$$S_k = S_{k|k-1} + \epsilon_k \quad (8.16)$$

and will do so with a prior addition probability distribution

$$p(S_k | S_{k|k-1}) = \begin{cases} \Pr(\epsilon_k = 0) = 1 - h_b \\ \Pr(\epsilon_k = 1) = h_b \end{cases} \quad (8.17)$$

where h_b is the probability of incrementing the number of targets. In general h_b and h_d will be set equal, except when $S_{k-1} = 0$ where we will set $h_d = 0$, or when $S_{k-1} = S_{\max}$ where we will set $h_b = 0$. Note that in some cases it is possible for the target number to be decremented by the removal of a target, and then increment by the introduction of another — thus leaving the overall number of targets unchanged.

The overall prior probability distribution for the number of targets is then simply the product of the individual probabilities

$$p(S_k|S_{k-1}) = p(S_{k|k-1}|S_{k-1}) \times p(S_k|S_{k|k-1}). \quad (8.18)$$

Prior Distribution of new target positions

The prior state distribution of new target births, $p_0(\alpha_k^s)$, may be chosen to reflect areas of the room in which new speakers are more likely to appear — such as near the doorways of a room or perhaps near a white board in a lecture theatre. To maintain the generality of our approach no such information will be used at this stage and the prior distribution of the location parameters will be set to be uniform across the cell, $p_0(x_k^s, y_k^s) = \mathcal{U}_{\mathcal{T}}(x_k^s, y_k^s)$, where $\mathcal{T}$ will be the volume of the entire surveillance region. Secondly, the prior distribution of the velocity components $p_0(\dot{x}_k^s, \dot{y}_k^s)$ will be initiated normally around zero velocity to give

$$p_0(\alpha_k^s) = p_0(x_k^s, y_k^s) \times p_0(\dot{x}_k^s, \dot{y}_k^s) \quad (8.19)$$

Thus the overall prior distribution of the full state vector, $\mathcal{A}_k$, can be stated as follows

$$p(\mathcal{A}_k|\mathcal{A}_{k-1}) = p_{\alpha}(\alpha_k^{1:S_k}|\alpha_{k-1}^{1:S_k}, S_k, S_{k-1})p_S(S_k|S_{k-1}) \quad (8.20)$$

where the portion of the prior related to the target positions can be broken down as follows

$$p_{\alpha}(\alpha_k^{1:S_k} | \alpha_{k-1}^{1:S_k}, S_k, S_{k-1}) = \begin{cases} \prod_{s=1, s \neq s'}^{S_{k-1}} p(\alpha_k^s | \alpha_{k-1}^s) & \text{if } \epsilon_{k|k-1} = -1, \epsilon_k = 0 \\ \prod_{s=1}^{S_{k-1}} p(\alpha_k^s | \alpha_{k-1}^s) & \text{if } \epsilon_{k|k-1} = 0, \epsilon_k = 0 \\ p_0(\alpha_k^{S_k}) \times \prod_{s=1, s \neq s'}^{S_{k-1}} p(\alpha_k^s | \alpha_{k-1}^s) & \text{if } \epsilon_{k|k-1} = -1, \epsilon_k = 1 \\ p_0(\alpha_k^{S_k}) \times \prod_{s=1}^{S_{k-1}} p(\alpha_k^s | \alpha_{k-1}^s) & \text{if } \epsilon_{k|k-1} = 0, \epsilon_k = 1 \end{cases} \quad (8.21)$$

where s' is the target removed at time k .

8.5.2 Proposal Mechanism

As discussed in the introductory chapters, our goal is to estimate the joint posterior distribution of the target states recursively, using the two step Bayesian update rule presented in Section 3.3.

The problem at hand has many state variables and moreover has a time-varying number of speakers encoded into it. Hence, instead of sampling from the dynamical model alone, as would be done in the standard bootstrap versions of particle filtering, [44], we will instead sample the p th particle for the new state vector from an appropriately selected proposal function

$$\begin{aligned}\mathcal{A}_k^{(p)} &\sim q(\mathcal{A}_k | \mathcal{A}_{k-1}^{(p)}, \mathbf{Z}_{1:k}) \\ &\sim q_\alpha(\alpha_k^{(\cdot)} | \alpha_{k-1}^{(\cdot)(p)}, S_k^{(p)}, S_{k-1}^{(p)}, \mathbf{Z}_{1:k}) q_S(S_k | S_{k-1}^{(p)}, \mathbf{Z}_{1:k})\end{aligned}\quad (8.22)$$

where $q_\alpha(\cdot)$ and $q_S(\cdot)$ are importance sampling functions for the position/velocity and target number states respectively, and an appropriate correction is then made for the bias introduced in the importance weighting step (see [15] for details).

According to Equation 8.22, we first propose the new target number in time-frame k by first removing unsupported targets and then adding targets to newly active regions of the existence grid as follows.

1. Removal of targets: Using the existence cell probabilities evaluated in Section 8.2, $g_{1:J}$, the set of (normalised) relative probabilities for the removal of a target were evaluated using Equation 8.6, $\bar{\kappa}_0, \dots, \bar{\kappa}_{S_{k-1}}$. This set of values is then used to draw a decision of whether a target is removed or not

$$q(S_{k|k-1} | S_{k-1}^{(p)}) = \begin{cases} \Pr(\epsilon_{k|k-1} = -1) = 1 - \bar{\kappa}_0 \\ \Pr(\epsilon_{k|k-1} = 0) = \bar{\kappa}_0 \end{cases} \quad (8.23)$$

Should the removal of a target be decided upon, a random draw from the set of target removal probabilities is made, the associated target is removed and the intermediary target number is decremented as follows

$$S_{k|k-1}^{(p)} = S_{k-1}^{(p)} - 1. \quad (8.24)$$

Otherwise no action is taken.

2. Initiation of new targets: In a similar manner¹ to the above a set of (normalised) relative probabilities for the addition of a new target were evaluated in Section 8.8, $\bar{\nu}_1, \dots, \bar{\nu}_J$. An addition decision is then made as follows

$$q(S_k^{(p)} | S_{k|k-1}^{(p)}) = \begin{cases} \Pr(\epsilon_k = 0) = \bar{\nu}_0 \\ \Pr(\epsilon_k = 1) = 1 - \bar{\nu}_0 \end{cases} \quad (8.25)$$

Should a new target addition be decided upon, a random draw is made to choose a cell in which it should be initiated, drawn according to this set of probabilities, which are again normalised appropriately.

Having selected the cell, the target position is initialised using a weighted combination of a uniform distribution within the physical region of cell, $\mathcal{T}_j$, and a normal distribution centred on the weighted mean of any particle states currently existing in that cell, $\bar{\alpha}_k^{(j)}$. The idea being that some particles may have detected the correct object position in an earlier time frame,

$$\begin{aligned} \alpha_k^{s(p)} &\sim q_0(\alpha_k^s | \mathbf{Z}_{1:s}) \\ &\sim \beta \mathcal{N}(\alpha_k^s; \bar{\alpha}_k^{(j)}, \bar{\sigma}_k^{2(j)}) + (1 - \beta) \mathcal{U}_{\mathcal{T}_j}(\alpha_k^s) \end{aligned} \quad (8.26)$$

The parameter β is set to be 0.7 in what follows.

3. Updating of persistent target positions: Finally the states of targets persisting from time-step $k - 1$ to k are propagated using the usual Langevin dynamical model, $\alpha_k^{s(p)} \sim q(\alpha_k^s | \alpha_{k-1}^{s(p)}, \mathbf{Z}_k)$.

In this way four distinct events can occur: one target may be birthed to a particle, one may be removed from a particle, a target may be birthed and another removed and finally no change in the target set may occur from the previous time-step.

8.5.3 Importance Weights

Having determined the particle set for the current iteration, the importance weights will be updated using

¹ Currently there exists a supplementary limitation which forbids the addition of more than one target within a particle to the same cell.

Algorithm 4: *Variable Dimension AST Particle Filter*

```

Update the existence grid
for  $p \in \{1 : N_p\}$  do
    Draw target removals according to (8.23)
    Draw target additions according to (8.26)
Propagate maintained targets using dynamical model
for  $p \in \{1 : N_p\}$  do
    for  $s \in \{1 : N_s\}$  do
        Evaluate the target likelihood
        Evaluate the particle likelihood (8.28)
        Evaluate the weight correction (8.27)
    Update the weights  $w_k^{(p)*}$  of the particle set
Resample the particle set using the weights if necessary

```

$$w_k^{(p)} \propto w_{k-1}^{(p)} \frac{l(\mathbf{Z}_k | \mathcal{A}_k^{(p)}) p(\mathcal{A}_k^{(p)} | \mathcal{A}_{k-1}^{(p)})}{q(\mathcal{A}_k^{(p)} | \mathcal{A}_{k-1}^{(p)}, \mathbf{Z}_{1:k})}. \quad (8.27)$$

where the likelihood term is determined up to a constant of proportionality by using a likelihood ratio calculation, as in [35, 82] and Chapter 7. The SBF grid with a 10cm density integrated over the frequency range of 200-6000Hz was adopted. The formulation as a likelihood ratio implies that we only need evaluate this function at the grid cells that contain targets, and the computation need only be made once and stored for each grid cell, and not for each particle containing a target within that cell. This SBF surface is computed separately from the low resolution SBF required for the activity grid detector in Section 8.2 and the likelihood ratio is calculated as

$$l(\mathbf{Z}_k | \mathcal{A}_k^{(p)}) = \prod_{s=1}^{s=S_k} l(z_{i_s} | \alpha_k^{s(p)}) \quad (8.28)$$

where $l(z_{i_s}, \alpha_k^{s(p)})$ is the individual target likelihood ratio for target s located in cell i_s . The measurement value, z_{i_s} , is derived from the steered response power of the SBF steered to the centre of that cell in the same way as (8.4).

In previous sections the likelihood ratio was formed by assuming that the distribution of measurement values maintained a (normal) distribution with identical variance statistics in both noise and a combination of signal and noise to give

$$l(z_{i_s} | \alpha_k^{s(p)}) = \exp\left(\frac{2z_{i_s} - 1}{2\sigma_N^2}\right). \quad (8.29)$$

However this can be seen to be an special case of the more general formula

$$\begin{aligned} l(z_{i_s} | \alpha_k^{s(p)}) &= \frac{p_{S+N}(z_{i_s} | \alpha_k^{s(p)})}{p_N(z_{i_s} | \alpha_k^{s(p)})} \\ p_{S+N}(z_{i_s} | \alpha_k^{s(p)}) &= c_1(\mathcal{N}(z_{i_s}; 1, \sigma_1) + q_1), \quad 0 < z_{i_s} < 1 \\ p_N(z_{i_s} | \alpha_k^{s(p)}) &= c_0(\mathcal{N}(z_{i_s}; 0, \sigma_0) + q_0), \quad 0 < z_{i_s} < 1 \end{aligned} \quad (8.30)$$

when $\sigma_0 = \sigma_1 = \sigma_N$ and $q_1 = q_0 = 0$. Some experimentation with this more general formula was carried out so as to choose parameters which give suitable behaviour.

8.5.4 Discussion

This algorithm provides a more principled and straightforward approach to multi-target acoustic source tracking with a time varying number of sources. A representation of the overall algorithm model can be seen in Algorithm 4.

It should be noted that because of the temporal discontinuity of speech, for multiple acoustic source tracking it is necessary to trade off the better tracking accuracy of a dominant source against improved tracking stability of weaker, less active sources. This trade-off involves careful choice of the likelihood parameters and judicious use of resampling strategy parameters.

The resampling time-frame is one of the system parameters that is in particular need of careful tuning. When two sources (for example) are active we can typically expect only 45% to give location estimates in the region of an individual source while the same proportion of frames will contain clutter measurements elsewhere in the surveillance space (See Figure 5.12 for more details of this experiment). This means that is possible for there to be extended periods in which only one of the sources is unobservable. Because of this it is important to implement the algorithm such that resampling of the filter occurs at a lower rate than this so as to allow the particle set to reflect the behaviour of the sources and to avoid degeneracy.

Finally in the conclusion section of this chapter, Section 8.7, more general discussion about the limitation of these algorithms is given as well as further comparison with the previous algorithm.

8.6 Experiments

To test the algorithm, a set of recordings were made in a typical office room with twelve microphones spaced around a roughly 5m x 5m space and illustrated in Figure 8.8. The setup and other details were identical to that used in Chapter 7. 500 particles were used in iterations which allowed for realtime operation in MATLAB on a typical PC (1.20GHz, 2GB RAM).

First we will illustrate the performance limitations of the algorithm presented in Section 8.4. Section 8.6.1 illustrates the algorithm tracking a single intermittent target correctly, however in Section 8.6.2 tracking of two simultaneous sources was attempted. The behaviour of the existence grid is such that, when the number of particles is stable tracking is quite accurate, however when it is unstable the proposal mechanism is unable to correctly determine activity and decided to halt tracking of the source.

Second we will present example simulations in which the algorithm presented in Section 8.5 is more successful. First we illustrate successful tracking of two non-overlapping conversational speech sources (Section 8.6.2). In the following section a much more challenging circumstance of two simultaneously active sources, each alternating between activity and inactivity, is examined and tracking is seen to be accurate and stable.

Finally in Section 8.6.3 the results of a series of Monte Carlo simulations are presented to examine the performance of the preferred algorithm.

8.6.1 Intermittent Single Speakers Examples

The first experiment carried out will illustrate the tracking performance of the first algorithm presented in Section 8.4. The speech sample tested was that of Source 1 in Figure 8.8 (without the presence of the second indicated source). Figure 8.5 illustrates the results of an algorithm run.

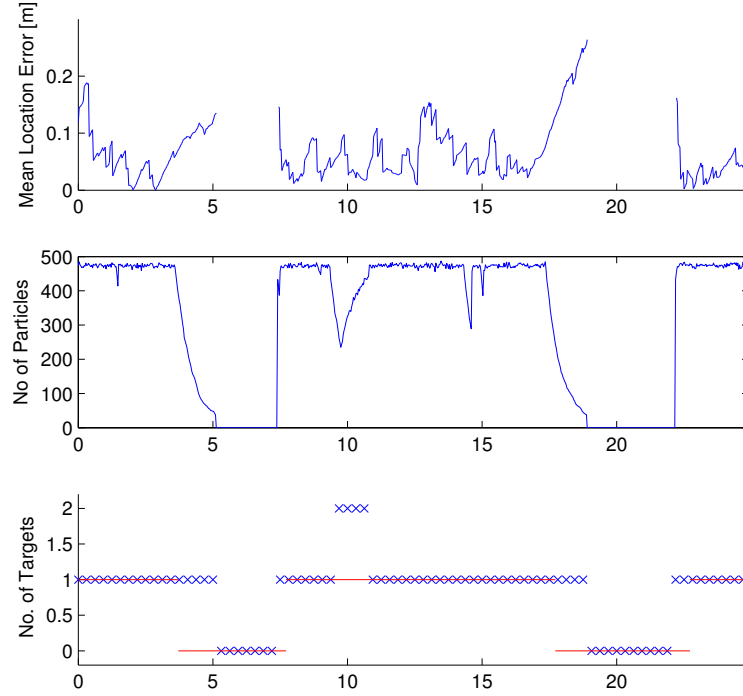


Fig. 8.5. *Successful tracking of a single intermittent speech source using the algorithm presented in Section 8.4. See Section 8.6.1 for more details.*

The red line in the lower plot illustrates the portions of time in which the source was active or inactive, overlaid on this plot is the number of particle filter tracks present. It can be seen that the algorithm is broadly successful in determining when the source is active or inactive although there is some lag in recognising when the speaker pauses. The upper plot illustrates the tracking accuracy of the filter (when tracking is underway). When the source is active tracking accuracy is comparable to other AST algorithms, but for the period after the speaker becomes silent (and therefore impossible to track in this way) it can be seen that the system continues to provide an erroneous tracking estimate.

The cause of the system's lack of response is indicated in the centre plot. When the source stops speaking, the number of particles in the system only

gradually decays, as it is directly connected to the magnitude of the associated existence grid value. While this slow response might be acceptable in this single source example, in the following section we illustrate that this particular algorithm will not be successful for two source tracking.

8.6.2 Two Speakers Examples

As discussed in Section 8.2.6, despite optimisation of the parameters of the existence grid there will exist occasions in which regions of the grid are incorrectly determined to be active or inactive. Figure 8.4 illustrated that when two sources are active there are likely to be occasions in which the number of source regions is underestimated and also occasions in which regions are falsely presented as active.

Figure 8.6 presents the tracking results for a sample of two source tracking. The tracking accuracy of each source is presented as well as the number of particles tracking that source.

The filter can be seen to broadly determine that there are two sources active, and when it does so tracking is broadly accurate. However stability of the tracking segments has been greatly compromised. Each source is tracked for only a few seconds at a time before the target track is erroneously removed. Because the proposal and removal of particles is directly connected to the associated existence grid values this algorithm has proven to be unsuccessful. Instead the variable dimension particle filter, proposed in Section 8.5, uses the existence grid only as an indirect proposal mechanism and as such the instability observed here can be avoided.

Two Intermittent Speakers

To illustrate the performance of the second algorithm from Section 8.5 we will present similar experiments to those presented in the previous sections. Figure 8.7 depicts tracking performance in the $\mathcal{X}$ and $\mathcal{Y}$ -dimensions for two alternating speakers taking part in a conversation. The duration of the sample is 20 seconds. The location of each source during **active** speech is indicated by a red line. Overlaid is the results of a typical run of the algorithm in

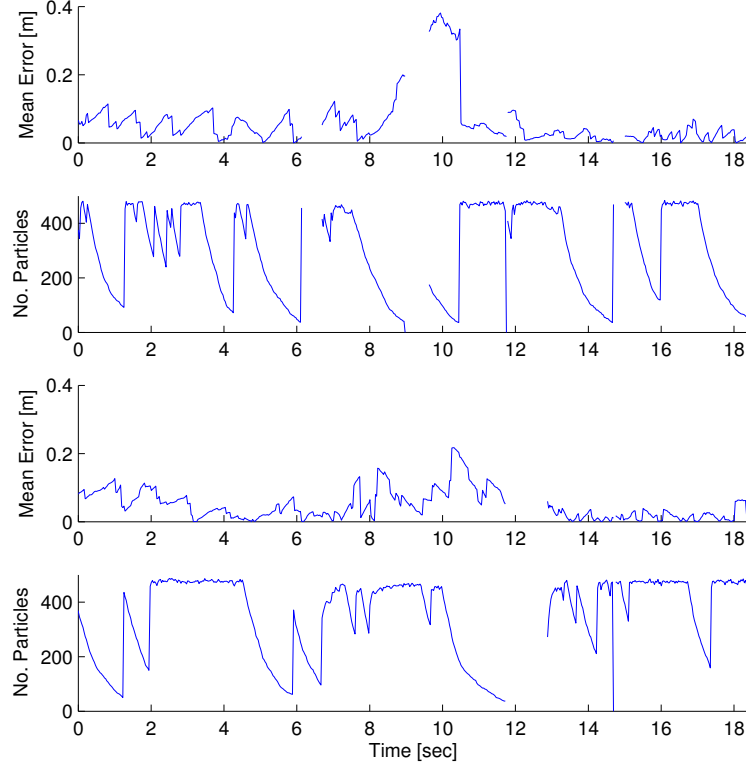


Fig. 8.6. *Unsuccessful tracking of two simultaneously active speech sources using the algorithm presented in Section 8.4 illustrating the limitations of the algorithm. See Section 8.6.2 for more details.*

blue. Every twentieth frame the variance of the location estimate is indicated by error bars. The algorithm can be seen to correctly identify and track the active source and to quickly switch between the two speakers.

Two Speakers in Conversation

Figure 8.8 illustrates the tracking of two sources alternating between activity and inactivity which includes segments in which both sources are simultaneously speaking. The upper plot illustrates tracking performance in both X and Y dimensions with source position estimates indicated by crosses. The lower

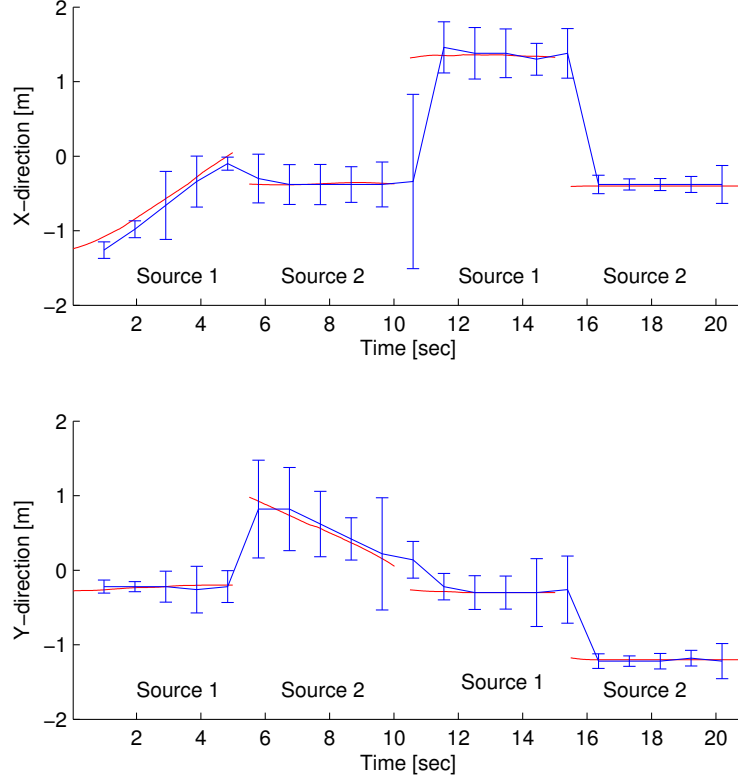


Fig. 8.7. Tracking two sources in conversation using the algorithm presented in Section 8.5. Note how at 10 seconds the error bars indicate high uncertainty in the silent gap between the speakers before continuing accurate tracking.

plot illustrates the number of sources estimated to be active (again indicated by crosses) compared to the number that actually were.

As mentioned previously the existence grid only broadly indicates regions in which the source is active, this method is successful because it uses this grid only as a preliminary indication of regional source activity. The algorithm is seen to preform tracking of both of the sources successfully — both when they were active and where they were active — despite the complexity caused by the repeated activity transition of the sources.

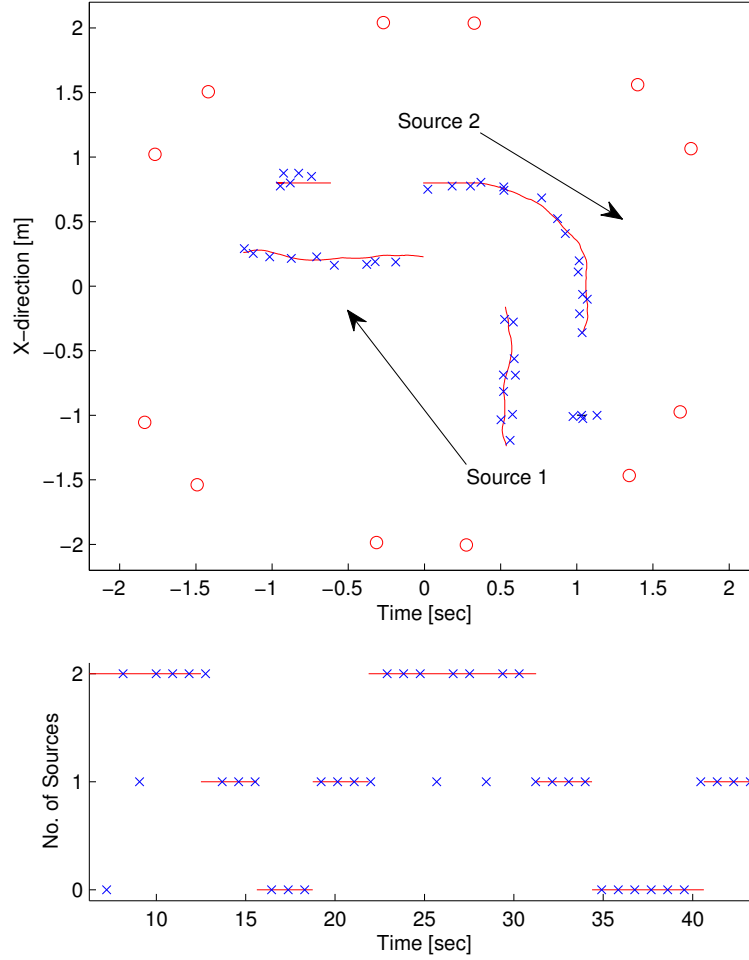


Fig. 8.8. Tracking of two intermittent and overlapping sources using the algorithm presented in Section 8.5. Also indicated is the mean estimate of the number of sources active compared to the number that actually were active.

8.6.3 Monte Carlo Simulations

Finally in this section we will present the results of a series of Monte Carlo simulations to examine the performance of the algorithm more closely. As mentioned previously, the proposed algorithm is unique in determining the

presence, activity and continuity of speech sources in an entirely probabilistic manner² and as such there exists no method which can be compared with the proposed algorithm. Hence the results presented here give only an illustrative example of the performance of the algorithm.

Because the algorithm utilises a variable dimension particle filter, the particle system will evolve to contain a mixed set of targets (containing between 0 and 3 targets) and as a result explicit estimation of source positions is not simple. We will choose to collapse the position dimensions of the targets onto one another so as to form a kernel surface with the combined set of particle positions. An example of this surface is illustrated in Figure 8.9 for two speech sources. The strong peaks in the kernel function are taken to be estimates of source positions.

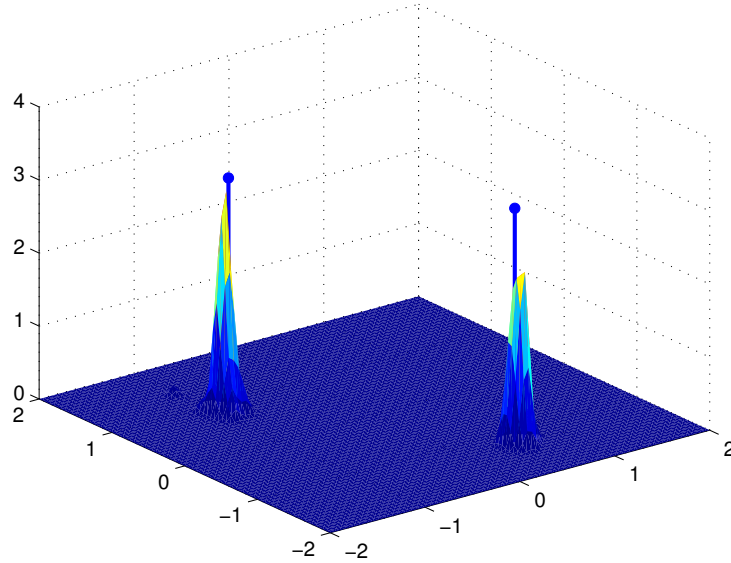


Fig. 8.9. *Kernel surface illustrating the distribution of particles tracking two sources (whose true positions indicated by blue dots).*

² Although the approach taken by Ma et al. [64] has the capacity, if re-implemented using the SBF measurement framework, to behave similarly.

Tracking of one intermittent source is examined here — in a similar way to the experiment in Section 8.6.1. The single source recording of Source 2 in Figure 8.8 (before the linear addition of Source 1) was used. This consisted of three periods of silence followed by three periods of speech activity and was 67 seconds in duration. The algorithm was run 50 times and the results were averaged.

The following metrics were then used to illustrate performance:

1. Mean Location Error (of frames when the source is active): 0.055m. This error is of a similar size to the mouth and the margin of error of the ground truth system, thus the system can be deemed to perform accurate tracking.
2. Percentage of frames (in which the source is active) that 70% of particle weight lies within 0.2m of the correct source position: 98.9%. This illustrates that tracking is stable — although it must be acknowledged that the test sample, while realistic, was not particularly challenging.
3. Mean error of the estimated number of targets: 0.394 targets **more** are estimated. Typically our implementation of the system overestimates the number of targets so as to avoid missing a target. Avoiding underestimation of sources is deemed to be more important in a number of applications.
4. Mean time taken for the proportion of particle weight lying within 0.2m of correct source position to rise to 70% (when becoming newly active): 0.28 seconds. As proposed in Section 8.2, the system quickly detects new sources.
5. Mean time taken for the proportion of particle weight lying within 0.2m of correct source position to fall to 30% (when becoming newly inactive): 0.87 seconds. As a result of a prior bias towards detecting new sources, old sources are removed more slowly.

8.7 Conclusions

A probabilistic algorithm for the detection and tracking of an unknown and time varying number of speaker has been proposed and demonstrated with real audio recordings. While there exists considerable scope for further optimisation of the algorithm, the results illustrate the promising behaviour of the system and an ability to track more than one source simultaneously and in real-time.

There still exists a need to improve the stability of the existence grid mechanism. Currently the existence grid is implemented using a grid of non-overlapping cells which can lead to instability during the passage of a target from one cell to the next. An alternative system utilising an two interleaved mesh grids could possibly remove this instability while requiring only a small increase in computing power.

However the main limitation of the algorithm is a maximum number of simultaneous active sources in the region of 2-3 sources. This limitation could perhaps be removed by notch filtering of dominant speakers and is further discussed in Chapter 9.

Part III

Conclusions

Conclusion

This thesis has presented a number of improvements to and extensions of the state of the art of Acoustic Source Tracking. Furthermore the results of experimental recordings carried out herein have illustrated fundamental behavioural characteristics so as to best determine the correct direction for future work in the field.

9.1 Contributions

Chapter 5 presented a series of experiments to quantify the performance limitations of the two commonly used AST localisation functions: the Generalised Cross-Correlation and the Steered Beamformer.

The experiments carried out with the GCC illustrated that correlation between recordings of human speech of a typical room is highly dependent on a series of environmental parameters. These include: the distance between the microphones, the distance from the source to the microphones, the ambient noise level and the presence of more than one speech source. The concept of speaker directivity was also discussed and quantified so as to provide motivation for an algorithm which estimates speaker orientation which was later presented in Chapter 6.

The second half of the chapter studied the phase transformed steered beamformer (PHAT-SBF). First, the characteristic distribution of the clutter and signal measurements was studied so as to motivate the choice of the location likelihood functions in subsequent chapters.

The shape of the SBF surface (also referred to as the Global Coherence Field) was shown to be directly related to the set of frequencies used to calculate the SBF function. It was seen that when implementing a discretised SBF surface, it is necessary to consider this frequency integration range when deciding upon a grid density so as to avoid peak aliasing.

Finally the distribution of accurate SBF location measurements was studied as a function of source movement, ambient noise level and in the presence of two sources. Deterioration in behaviour of the SBF function for a moving source was seen to be negligible. Meanwhile a predictable deterioration in the proportion of source measurements was seen with increasing noise level. However the presence of more than a single active source — be it another speech source or other environment effects — was seen to provide the most difficult challenge to AST algorithms as it was seen to substantially reduce the proportion of frames in which useful source measurements are to be expected.

Chapter 6 presented a novel algorithm to estimate the orientation of an active acoustic source. The proposed likelihood functions utilise the manner in which the distribution of GCC function peak magnitudes changes with source orientation.

After some experimentation to optimise the parameters of the orientation likelihood function, successful operation was illustrated for a moving speech source in which simultaneously position and orientation tracking was carried out.

Chapter 7 considers the experiments carried out in Chapter 5, before a strategy to discretise the SBF function is proposed. The proposed strategy utilises the Track-Before Detect Framework. This approach assumes that a source may contribute to only a single measurement cell of a discrete grid. This assumption was also discussed therein.

In taking this approach the proposed algorithm avoids an unnecessary thresholding process which would otherwise inoptimally identify the set of signal measurements. Because the proposed algorithm discretises the underlying likelihood function, computation may be better distributed as co-located particles will share the same likelihood value (which is evaluated once only).

This means that many more particles may be utilised by the system without increasing the overall computation — which in turn improves tracking stability.

The choice of the grid density is an important one (both from the point of view of tracking accuracy and computation). A series of experiments were carried out to ensure an optimal choice.

Because this approach avoids the need to assign a defined set of measurements, the surveillance space has effectively been partitioned and areas of the space not containing target particles are no longer considered. This also allows us to consider the regions containing different sources separately and in the later part of Chapter 7 a multi-target extension to this algorithm is proposed in which two simultaneously active sources are tracked.

However, as with other works in this field, there remains the fundamental assumption that the source positions are known at the outset of an algorithm run and that the source(s) remain active throughout. This assumption is obviously rather naïve. In practice an algorithm module which probabilistically determines source activity, proposes new sources and removes newly silent source would indeed be useful.

Chapter 8 such a module is considered. The two algorithms are proposed which monitor regions of the surveillance space using an existence grid. Using this grid the proposal of new particles (and in turn newly active speech sources) and the removal of unsupported particles are made. The first algorithm proposed that proposal and removal functions be directly connected to the existence grid — an approach which has proven to be unstable. This instability is caused by the slight pauses in speech between sentences and the interference of simultaneous speakers with one another.

Instead, a second algorithm was proposed which utilises a variable dimension particle filter drawn from the general tracking literature. The algorithm uses the existence grid as an initial proposal function before an importance weighting mechanism incorporates hypothesised prior behavioural information, the previous particle positions, the current measurement data to create a corrected weighting of the particle likelihoods.

Experiments illustrated that this algorithm successfully proposes new targets, tracks their positions and removes them when they become inactive. Further experiments showed successful operation for two simultaneous and intermittent sources.

9.2 Future Direction

While specific proposals for future improvements to the proposed algorithms have been suggested in their respective chapters, here we will suggest some future directions for the field in general.

It has been recognised here and elsewhere that a typical source will alternate between activity and inactivity frequently. This information has not as yet been used to improve the dynamical model - which typically alternates between tracking an active source and clustering around the source position estimate when it becomes inactive. For example, smoothing of particle filter estimates has the ability to improve the final estimate of source position — particular if used for an off-line task such as meeting room diarisation, in which latency would not be an issue.

Secondly to allow an increase in the number of simultaneously active speech sources (currently limited to 2-3 sources), it is necessary to improve the measurement algorithms used to provide source location estimates. Consideration could be given to combination the localisation process with other speech processing tasks — such as the combination of steered beamforming with source separation and/or notch filtering of recognised sources. Other possibilities include the use of filter-banks to provide source position estimates at different frequency bands.

Finally, as discussed previously, there exists a necessity to establish a freely available corpus of useful meeting room data which can be used to test and benchmark the various algorithms which currently exist.

Part IV

Appendices

A

Practical Implementation Details

As mentioned in Section 2.1.2, the experiments detailed in this thesis were carried out using an array of 12 microphones placed around the outside of a typical office room.

The microphones were hung from a set of retort stands 1.2m from the ground. Behind each of the microphones was placed piece of acoustic insulation foam so as to reduce the effect of reverberation from directly behind the source. Although the microphones were arranged in a pairwise fashion for most of the experiments carried out herein, there was no specific reason for this other than practicality.

We used typical off-the-shelf tie-clip microphones which cost just a few pounds each. The specification of the microphones was as follows:

- Model No: Yoga EM-1
- Transducer Type: Back electret condenser
- Frequency Response: 100Hz–16KHz
- Pick-up Pattern: Omnidirectional (although no specification of this pick-up pattern was provided.)
- Sensitivity: -64dB
- Impedance: 1000 ohms

Each of the microphone signals were fed, via set of 6 stereo pre-amplifiers, into an A/D converter. B-Tech BT26 phono/mic pre-amplifiers were used:

- Input impedance: 50k Ohms @ 1kHz
- Input level (mic): 0.5mV, (phono): 3mV

- Gain (mic): 50dB, (phono): 34dB
- Signal-to-noise ratio (mic): ≥ 48 dB, (phono): ≥ 55 dB
- Frequency response (mic): 70 - 20,000Hz, (phono - with RIAA EQ): 30 - 20,000Hz
- Output level: 150mV RMS
- Load impedance: ≥ 10 k Ohms
- Overload margin: 23dB
- Crosstalk: 55dB

The microphone inputs (and not the phono inputs) were used. The A/D converter used was an RME Fireface 800 with the following specifications:

- 8 x analog line I/O (used for 8 of the 12 microphones)
- 1 x ADAT digital I/O (used for 4 of the 12 microphones)
- 192 kHz/24-bit
- 2 Microphone/Line Inputs with Preamps (unused)
- 2 Instrument/Line Inputs (unused)
- 1 x stereo headphone output (unused)
- 2 x MIDI I/O (unused)

Because the Fireface 800 has only 8 input channels, four of the channels were instead fed into a Alesis ADAT-LX20 before being passed to the Fireface 800 along the ADAT channel.

The digitised signals were then passed to laptop via a FireWire 400 (IEEE 1394) cable. Recording was mainly controlled using Adobe Audition (version 1 and later version 3), a digital audio recording, mixing and editing program. Some coding was implemented in C++ using the PortAudio API, see Section A.1 for more details.

The following recording and framework variables were used in all the algorithms presented in this text:

- Audio Sampling Rate: 16kHz
- Frame Length: $L = 512$ samples
- Update Rate: $\Delta T = 31.25$ Hz
- Frame Overlap percentage: 50%

Finally it should be noted that the *source* used in many of these experiments was typically a computer speaker mounted on a platform. While this is not expected to behave significantly different to using a person speaking, it is obviously much easier to reproduce results, localise (for ground truth) and of course cheaper! While computer speakers have also been used in much of the literature, it is nonetheless important to recognise this fact.

A.0.1 Ground Truth

To provide ground truth information of the source positions as well as accurate microphone positioning, the Impulse Motion Capture System from PhaseSpace [75] was used. This optical motion capture system utilises a series of high-end scanning cameras to position LEDs which were attached to the source (either the computer speaker or a person). Operating at a data capture rate of 480Hz and to sub-millimetre accuracy, this system was indeed over-the-top for this application but did provide the required ground truth information.

It should be noted that having positioned the microphones accurately, sufficiently accurate ground truth source position information can be inferred from the localisation measurements.

A.1 Real-time Implementation

PortAudio is a free, cross platform, open-source, audio I/O library. The platform was used to implement a real-time particle filter to track sources (using the track-before detect framework). However the amount of coding effort required precluded the completion of a stable, properly working system during this PhD.

Details regarding such a system implemented at the Australian National University can be seen in the appendix of [56], which discusses some of the important issues such as latency and the required computing power required.

B

Notation

The following is a list of the notation used throughout this thesis. For continuity with existing research an effort has been made to retain the notation used within the literature. Note that notation used only occasionally in the thesis as been omitted from this list:

- a - amplitude attenuation
- τ - (relative) time delay
- L - number of samples per audio frame
- k - time frame number
- m - microphone number
- N_m - number of microphones
- r - microphone pair number
- N_r - number of microphone pairs
- t - GCC peak number
- N_t - number of GCC peaks
- T - set of GCC peaks for a microphone
- p - Particle number
- N_p - Number of particles
- s - Source number
- N_s - Number of sources
- c - Target cluster number
- N_c - Number of target clusters
- $\mathbf{Z}$ - Overall measurement vector (usually a set of SBF values)

- $z_{i,j}$ - measurement value for cell (i, j)
- ΔT - time step (amount of time between iterations)
- $d_{1,2}$ - Euclidian distance between two particles 1 and 2
- $l \equiv (x, y)$ - Location of a microphone or a particle
- F_{rep} - target repulsion force

Existence Grid:

- (i, j) - existence grid cell co-ordinate. Often replaced by j alone.
- $g_{i,j}$ - existence grid value of cell (i, j)
- $\tau_{i,j}$ - probability of a target *not* existing in cell (i, j)
- $\kappa_{i,j}$ - probability of removing a target existing in cell (i, j)
- $v_{i,j}$ - probability of adding a target to cell (i, j)
- $[\Delta i, \Delta j]$ - physical size of existence grid cells

State Vector:

- $\mathcal{A}$ - Overall state vector
- α - A single target vector
- x - X co-ordinate position
- $\dot{x}$ - X co-ordinate velocity
- y - Y co-ordinate position
- $\dot{y}$ - Y co-ordinate velocity
- λ - (Binary) activity variable.
- θ - Orientation
- $\dot{\theta}$ - Orientation velocity
- w - particle weight

B.1 Terms and Abbreviations

- AEDA - Adaptive Eigenvalue Decomposition Algorithm
- AST/ASL — Acoustic Source Tracking/Localisation
- CCF - Cross Correlation Function
- FFT - Fast Fourier Transform
- GCF - Global Coherence Field

- GCC - Generalised Cross Correlation (Function)
- MAP - Maximum a posteriori
- MTT - Multi-Target Tracking
- PHAT - Phase Transform
- RIR - Room Impulse Response
- RT_{60} - 60dB Reverberation Time
- SBF - Steered Beamformer (Function)
- SIR - Sequential Importance Function
- SMC - Sequential Monte Carlo
- SRP - Steered Response Power (of the SBF)
- TBD - Track-Before Detect
- TDOA - Time Delay of Arrival

B.2 Measurement Metrics

In this section a number of the measurement metrics used throughout this thesis are explained and defined. These metrics are widely used in the literature and the conventions suggested by Ward et al. [93] are retained here.

Root Mean Square Error (RMSE)

The Room Mean Square Error is an the average measure of source tracking accuracy. First the square of the (Euclidian) distance between the source position estimate, $\hat{l}_k$, and the known ground truth position, l_k , is found

$$\epsilon_k^2 = |l_k - \hat{l}_k|^2 \quad (\text{B.1})$$

The RMSE is then simply the square root of the average of this parameter over time

$$\text{RMSE} = \bar{\epsilon}_k = \sqrt{\frac{1}{N_k} \sum_{k=1}^{N_k} \epsilon_k^2} \quad (\text{B.2})$$

where N_k is the number of frames in a particular sample. The metric was also used to examine the orientation estimation algorithm in Chapter 6, albeit in only a single dimension.

Note that RMSE cannot be simply determined when using the variable dimension particle filters introduced in Chapter 8. Instead a kernel surface was formed using the particle positions and weights and the peak of that surface was assumed to be the position error. This particular metric is, however, not well defined. As a result of this is, it can be difficult to compare estimates created by variable dimension particle filters with either fixed dimension particle filters or instantaneous estimates.

Mean Standard Deviation of the particle cluster (MSTD)

The standard deviation is a measure of the spread of the particle cluster. Although the particle filtering is intended to model multi-modal distributions, the magnitude of the spread of the cluster is indicative of tracking stability and the uncertainty of position estimates. First the standard deviation is evaluated for each frame, around the estimated source position

$$\sigma_k = \sqrt{\sum_{p=1}^{N_p} w_k^p |l_k^p - \hat{l}_k|^2} \quad (\text{B.3})$$

and then the average of this parameter over time gives the MSTD

$$\text{MSTD} = \frac{1}{N_k} \sum_{k=1}^{N_k} \sigma_k. \quad (\text{B.4})$$

Track Loss Percentage (TLP)

Finally the Track Loss Percentage is a measure of the proportion of algorithm runs in which tracking is deemed to have failed at the end of the tracking run. An algorithm run in which tracking is lost is defined as a run in which the estimated position error for the final frame of the algorithm run is greater than some threshold

$$\xi = \begin{cases} 1 & \text{if } \epsilon_k > \delta \\ 0 & \text{otherwise.} \end{cases} \quad (\text{B.5})$$

where δ will be chosen, arbitrarily, to be 0.1m. This parameter has been defined somewhat differently elsewhere.

In addition to these measures, Chapter 8 introduces some heuristic measures to illustrate the responsiveness of the tracking algorithm (proposed therein) to react to changes in source activity.

C

Details of Alternative Multi-target Tracking Algorithm

In this section the full details of an multi-target particle filtering algorithm discussed in Section 8.4 are presented. This approach as proven to be instable in certain tracking circumstances. It is presented here as a comparison to another, improved, algorithm presented in Section 8.5.

Section 8.6 presents some experiments which illustrate the algorithm's instability.

C.1 Framework 1: Defined Target-Source Assignment

Consider the following scenario at time k . We shall define there to be S_k target tracks in the surveillance region — one for each source thought to exist at that time. Each of the tracks will contain a variable set of p_s target incidences up to a maximum of N_p particles. The number of target incidences in a particular track will vary according to the addition and removal processes outlined in Section 8.2.5 and will broadly reflect the actual number of sources that are likely to exist.

The state vector for an individual target incidence will be (dropping the time index for simplicity)

$$\alpha^s = (x^s, y^s, \dot{x}^s, \dot{y}^s, \lambda^s) \quad (\text{C.1})$$

which is as previously used in Equation 7.15, while the combination of the individual state vectors makes up the overall state vector.

$$\mathcal{A} = (\alpha^1, \dots, \alpha^{S_k}) \quad (\text{C.2})$$

Note that an individual particle will thus evolve to contain a mixed set of targets.

Secondly it should be reemphasised that while *existence* of a target track within the overall state vector represents longer sentence-length speech activity — as distinct from syllable-length *activity* measured by the variable λ^s in Chapter 7.

Particle Proposal Scheme

The set of state vectors at a particular time step, k , will be the combination of these state vectors maintained from the previous time step, updated via a removal and addition process to reflect the existence grid.

Maintained Targets

The state of a maintained target will be propagated according to the usual Langevin dynamical model, as described in Section 4.2.2. For closely spaced targets this dynamical model will be appended with a *repulsion mechanism* which will be discussed in Section C.1.2.

Removing Targets

Should the number of targets within the surveillance region change, the values of the existence grid will change so as to represent this. A target no longer supported by the existence grid will be removed from the containing state vector as follows: first the decision is made as to whether no targets or one target is to be removed

$$p(S_{k|k-1}|S_{k-1}) = \begin{cases} \Pr(\epsilon_{k|k-1} = -1) = 1 - \bar{\kappa}_0 \\ \Pr(\epsilon_{k|k-1} = 0) = \bar{\kappa}_0 \end{cases} \quad (\text{C.3})$$

where $\epsilon_{k|k-1}$ represents the number of targets removed from the particle at that time step. If a removal move is drawn then a target is chosen and removed from the system

$$\begin{aligned}
s &\sim [\bar{\kappa}_1, \dots, \bar{\kappa}_{S_{k-1}}] \\
\alpha^s &\leftarrow \emptyset
\end{aligned} \tag{C.4}$$

where the set of cell addition probabilities, $[\bar{\kappa}_1, \dots, \bar{\kappa}_{S_{k-1}}]$, are properly re-normalised.

The associated weight for this target is set to zero and *implicitly* the target number is decremented, $S_{k|k-1} = S_{k-1} + \epsilon_{k|k-1}$.

Adding Targets

In a similar way a new target instance may be added to a particle if supported by the existence grid. Again the first step is to decide to either add no targets or one

$$p(S_k | S_{k|k-1}) \sim \begin{cases} \Pr(\epsilon_k = 0) = \bar{\nu}_0 \\ \Pr(\epsilon_k = 1) = 1 - \bar{\nu}_0 \end{cases} \tag{C.5}$$

and if the add move is chosen a cell is selected

$$j \sim [\bar{\nu}_1, \dots, \bar{\nu}_J] \tag{C.6}$$

where the set of cell removal probabilities, $[\bar{\nu}_1, \dots, \bar{\nu}_J]$, are properly re-normalised. Once more the target number is incremented, $S_k = S_{k|k-1} + \epsilon_k$.

Should this new target instance be proposed in a cell containing an already existing target, s , it will be initialised using a normal distribution surrounding the currently existing target incidences, as follows (for the x-direction components)

$$x_s^{(p)} \sim N(x_{s,\text{MAP}}, \sigma_{s,\text{MAP}}) \tag{C.7}$$

$$\dot{x}_s^{(p)} \sim N(0, \sigma_{\text{init}}) \tag{C.8}$$

where $x_{s,\text{MAP}}$ and $\sigma_{s,\text{MAP}}$ are the MAP position and variance of the target. This is repeated for the $\mathcal{Y}$ -direction component in a similar manner. In this way the particle set will evolve to contain a mixed set of targets — some having one, two or more targets.

However if a new target is proposed in a cell which is devoid of other already existing targets then a completely new track may be necessary. In the following section such a procedure is discussed.

Finally note that the total *number of targets* within a particle is implicitly determined and no prior belief in how this number would evolve is encoded here. Algorithm 2 takes a different approach in which the number of targets is explicitly modelled.

C.1.1 Initialising and terminating target tracks

When a new person begins to speak, the value of the existence cell in which the speaker is located will gradually become elevated and hence the number of new target incidences being proposed within this cell, via Equation C.6, will increase. Should there be no target currently in the region of this cell a new target track will be added and implicitly the total number of targets present will increase.

We will draw the state of the new target incidence (with target number s and particle number p) uniformly¹ across the existence cell in question as follows:

$$\begin{aligned} x_s^{(p)} &\sim \mathcal{U}[x_j - \Delta x/2, x_j + \Delta x/2] \\ \dot{x}_s^{(p)} &\sim N(0, \sigma_{\text{init}}) \end{aligned} \tag{C.9}$$

and similarly for the $\mathcal{Y}$ -direction components. Should the number of proposals consistently exceed the number of removals (as removed via Equation C.4) the number of target incidences for this source will rise organically.

However when a target is no longer supported by the existence grid the number of incidences of it will fall away organically according to Equation C.4 until it is eventually removed.

C.1.2 Particle Update Procedure

Having determined the target set, the final step is to complete the iteration by evaluating the target likelihoods and associated weights.

In the previous chapter we assumed that all particles contained an equal number of targets — each defined to track a particular speech source. Instead

¹ More precise strategies for determining the initial location could have been proposed but this strategy has been seen to work in practice

here particles will contain a mixed set of targets. Some particles may contain a particular target or targets, while others may not.

First we will cluster the target tracks according to proximity: gathering together tracks that lie within 50cm of one another and hence likely to interact. This is carried out using the MAP position of the **ensemble set** of target positions for target track — rather than considering individual target combinations.

The overall state vector of the tracking system will thus be made up of N_c clusters, which will be as follows for particle p

$$\mathcal{A}^{(p)} = (\mathbf{C}_1^{(p)}, \dots, \mathbf{C}_{N_c}^{(p)}). \quad (\text{C.10})$$

which will have been randomly combined.

Individual clusters will essentially exist as separate particle filter systems made up of one or more targets. The cluster $\mathbf{C}_c^{(p)}$ will contain $N_{c,t}$ targets and be defined as follows

$$\mathbf{C}_c^{(p)} = (\alpha_{c,1}^p, \dots, \alpha_{c,N_{c,t}}^p). \quad (\text{C.11})$$

with an associated cluster weighting $w_c^{(p)}$. The individual target state vectors will, once more, contain the position and velocity of the targets in both dimensions as well as the activity variable introduced in the previous chapter.

This approach, common in the tracking literature [54], partitions the state vector and maintains manageable dimensionality. In contrast to this approach Section 8.5 proposes an un-partitioned variable dimension particle filter.

Individual Target Clusters

Consider a target s which is widely separated from all other targets in the surveillance space. This target alone form a cluster ($\mathbf{C}_c \equiv \alpha_s$). Its likelihood ratio will be defined in a similar manner to that proposed in Equation 7.12

$$q(\mathbf{Z}|\alpha_s) \propto \begin{cases} \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} & \text{for } \lambda_s = 1 \\ & \text{and } |i\Delta - x_s| < \Delta/2 \\ & \text{and } |j\Delta - y_s| < \Delta/2, \\ 1 & \text{otherwise} \end{cases} \quad (\text{C.12})$$

As mentioned above the weighting mechanism and resampling for this target will be carried out independent of any other target clusters.

Algorithm 5: *Defined Target-to-Source Tracks*

```

Update the existence grid
for  $p \in \{1 : N_p\}$  do
    Draw target removals according to (C.4)
    Evaluate if a new target should be added (C.5) and select a cell (C.6)
    if Particle  $p$  does not contain a target in the proposed cell then
        if Proposed cell contains an existing target track then
            Add target to currently existing track according to (C.9)
        else
            Add new target track to the tracking system
    Separate targets into clusters based on target separation
    Propagate retained targets using dynamical model
    for  $c \in \{1 : N_c\}$  do
        for  $p \in \{1 : N_p\}$  do
            Draw a new activity variable,  $q_{s,k}^{(p)}$ , for each target
            Evaluate the target likelihoods using (C.12)
        Update the overall weights  $w_k^{(p)*}$  of the cluster
        Resample the cluster set using the weights if necessary

```

Multiple Target Clusters

If instead two or more target tracks are located close to one another, it will no longer be possible to assume target independence. Instead these targets will be combined to form a cluster, $\mathbf{C}_c^{(p)}$, in Section C.1.2 and the cluster state vector will be as represented in Equation C.11.

Dynamical Model: As discussed in the previous chapter, it is behaviourally unlikely that two speakers can be located very close to one another (less than 20-30 cm). To disallow this from occurring we will retain the repulsion mechanism introduced in Section 7.6.3.

Likelihood Function: The overall cluster likelihood ratio will simply be the product of the individual target likelihood ratios as follows

$$q(\mathbf{Z}|\mathbf{C}_c^{(p)}) = \prod_{i=1}^{N_{c,t}} q(\mathbf{Z}|\alpha_{s_i}) \quad (\text{C.13})$$

where the ratios are as evaluated in Equation 7.12.

The soundness of this approach is discussed in the following section, while typical results for the algorithm, run on real recorded data, will be presented in Section 8.6.

C.1.3 Discussion

This algorithm provides a methodology for tracking a time varying number of sources and is in spirit a direct extension of the non-time varying algorithm proposed in the previous chapter. However to allow for stable performance when there is more than a single active source it was necessary to abandon a truly principled approach and to integrate a number of ad-hoc algorithm components.

For example, the number of targets incidences assigned to track a particular source is directly determined by the birth and death processes in Section C.1. Also a heuristic approach is taken to target weighting which does not correct for the bias in target number estimation.

Furthermore the concept of an empty state vector, corresponding to no active sources, was not explicitly modelled in this framework — biasing against this situation. This behaviour has proven to be unstable in certain circumstances.

Instead in the Section 8.5 a full variable dimension particle filter is proposed in which each particle represents a single un-partitioned estimate of the underlying state space — an approach which develops a more right-minded solution to the problem. This filter will use the birth and death processes as proposal mechanisms alone while also incorporating prior information to improve target stability. Furthermore, particles containing zero targets will have as legitimate a contribution to the framework as those containing one or more targets.

References

1. Samer A. Abdallah and Mark D. Plumbley. Unsupervised analysis of polyphonic music using sparse coding. *IEEE Transactions on Neural Networks*, 17:179–196, January 2006.
2. Jont B. Allen and David A. Berkley. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Am.*, 65(4):943–950, 1979.
3. M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, February 2002.
4. Yaakov Bar-Shalom and William Dale Blair. *Multitarget-Multisensor Tracking: Applications and Advances*, volume 3. Artech House, 2004.
5. Jacob Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *J. Acoust. Soc. Am.*, 107(1):384–391, 2000.
6. Terence Betlehem and Robert C. Williamson. Acoustic beamforming exploiting directionality of human speech sources. In *Proc. ICASSP’03*, april 2003.
7. Y. Boers and J.N. Driessen. A particle filter multi target track before detect application. *IEE Proceedings - Radar, Sonar and Navigation*, 151(6), 2004.
8. Michael Brandstein and Darren Ward. *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.
9. Michael S. Brandstein. *A Framework for Speech Source Localization Using Sensor Arrays*. PhD thesis, Brown University, May 1995.
10. Michael S. Brandstein. A pitch-based approach to time-delay estimation of reverberant speech. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA97)*, October 1997.

11. Michael S. Brandstein. Time-delay estimation of reverberated speech exploiting harmonic structure. *Journal of the Acoustical Society of America*, 105, May 1999.
12. Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays. In *Proceedings of Eurospeech 2005*, 2005.
13. Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. Speaker localization based on oriented global coherence field. In *Proceedings of Interspeech 2006*, September 2006.
14. Cristian Canton-Ferrer, Carlos Segura, Josep R. Casas, Montse Pardo, and Javier Hernando. Audiovisual head orientation estimation with particle filtering in multisensor scenarios. *EURASIP Journal on Advances in Signal Processing*, 2008.
15. O. Cappé, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential monte carlo. *IEEE Proceedings*, 95(5), 2007.
16. Jean Carletta. Announcing the ami meeting corpus. *The ELRA Newsletter*, 11:3–5, 2006.
17. Benoit Champagne, Stephane Bedard, and Alex Stephenne. Performance of time-delay estimation in the presence of room reverberation. *IEEE Trans. on Speech and Audio Processing*, 4(2):148–152, March 1996.
18. Neal Checka, Kevin Wilson, Michael Siracusa, and Trevor Darrell. Multiple person and speaker activity tracking with a particle filter. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.
19. Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang. Robust time delay estimation exploiting redundancy among multiple microphones. In *IEEE Transaction on Speech and Audio Processing*, volume 11, NOVEMBER 2003.
20. Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang. Time delay estimation using spatial correlation techniques. In *Proceedings of 8th IEEE International Workshop on Acoustic Echo and Noise Control*, 2003.
21. Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang. Performance of gcc- and amdf-based time-delay estimation in practical reverberant environments. In *EURASIP Journal on Applied Signal Processing*, pages 25–36, 2005.
22. Jingdong Chen, Jacob Benesty, and Yiteng Arden Huang. Time delay estimation in room acoustic environments: an overview. *A Special Issue of EURASIP JASP on Advances in Multimicrophone Speech Processing*, 2006.

23. Jingdong Chen, Yiteng (Arden) Huang, and Jacob Benesty. Time delay estimation via multichannel cross-correlation. *Proc. ICASSP 2005*, 2005.
24. Joe C. Chen, Ralph E. Hudson, and Kung Yao. Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field. *IEEE Trans. on Signal Processing*, 50(8):1843–1854, August 2002.
25. Joe C. Chen, Kung Yao, and Ralph E. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, March 2002.
26. Wei-Peng Chen, Chao-Ju Hou, and Lui Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 3(3):258–271, July-Aug. 2004.
27. D. Crisan and A. Doucet. A survey of convergence results on particle filtering for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, 2002.
28. Joseph H. DiBiase. *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. PhD thesis, Brown University, May 2000.
29. Jacek P. Dmochowski, Jacob Benesty, and Sofiene Affes. A generalized steered response power method for computationally viable source localization. *IEEE Transactions in Audio, Speech and Language Processing*, 15:2510–2526, November 2007.
30. Hoang Do and Harvey S. Silverman. A fast microphone array srp-phat source location implementation using coarse-to-fine region contraction (cfrc). In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2007.
31. Randal Douc, Olivier Cappe, and Eric Moulines. Comparison of resampling schemes for particle filtering. *Proceeding of the 4th International Symposium on Image and Signal Processing and Analysis (2005)*, 2005.
32. Arnaud Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Univ. of Cambridge, 1998.
33. Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2000.
34. H.K. Dunn and D.W. Farnsworth. Exploration of pressure field around the human head during speech. *J. Acoust. Soc. Am.*, 10:184–199, January 1939.
35. Maurice Fallon and Simon Godsill. Multi target acoustic source tracking using track before detect. In *Proceedings of WASPAA 2007*, October 2007.

36. Maurice Fallon and Simon Godsill. Multi target acoustic source tracking with an unknown and time varying number of targets. In *Proceedings of HSCMA 2008*, May 2008.
37. Maurice Fallon, Simon Godsill, and Andrew Blake. Joint acoustic source location and orientation estimation using sequential Monte Carlo. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 203–208, Montreal, Quebec, Canada, Sept. 18-20 2006.
38. Cedric Févotte and S.J. Godsill. A Bayesian approach for blind separation of sparse sources. *IEEE Trans. on Speech and Audio Processing*, 14(6), 2006.
39. Cedric Fevotte, Bruno Torrsani, Laurent Daudet, and Simon J. Godsill. Sparse linear regression with structured priors and application to denoising of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 2008.
40. W.J. Fitzgerald, R.L. Smith, A.T. Walden, and P.C. Young, editors. *Nonlinear and Nonstationary Signal Processing*. Cambridge University Press, Cambridge, Britain, 2001.
41. Daniel Gatica-Perez, Guillaume Lathoud, Iain McCowan, Jean-Marc Odobez, and Darren Moore. Audio-visual speaker tracking with importance particle filters. In *IEEE International Conference on Image Processing (ICIP)*, 2003.
42. Daniel Gatica-Perez, Guillaume Lathoud, Jean-Marc Odobez, and Iain McCowan. Multimodal multispeaker probabilistic tracking in meetings. In *International Conference on Multimodal Interfaces*, pages 183–190, 2005.
43. Daniel Gatica-Perez, Guillaume Lathoud, Jean-Marc Odobez, and Iain McCowan. Audiovisual probabilistic tracking of multiple speakers in meetings. *IEEE Transactions on Audio, Speech and Language Processing*, 15(2), february 2007.
44. Neil Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *Proceedings IEE-F*, 140:107–113, 1993.
45. Yusuke Hioka and Nozomu Hamada. Estimation of azimuth and elevation doa using microphones located at apices of regular tetrahedron. *IEEE Fundamentals - SE on Digital Signal Processing*, 87(8), August 2004.
46. Yiteng Huang, Jacob Benesty, and Gary W. Elko. Adaptive eigenvalue decomposition algorithm for real time acousticsource localization system. In *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP99)*, volume 2, pages 937–940, 1999.

47. Michael Isard and Andrew Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proceedings of the European Conference on Computer Vision*, pages 893–908, June 1998.
48. Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, April 1970.
49. Anders M. Johansson, Eric A. Lehmann, and Sven Nordholm. Real-time implementation of a particle filter with integrated voice activity detector for acoustic speaker tracking. In *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS'06)*, pages 1004–1007, December 2006.
50. Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.
51. Hyun-Don Kim, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Auditory and visual integration based localization and tracking of humans in daily-life environments. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2007)*, pages 2021–2027, October 2007.
52. Cyber Kitchen. <http://www.cksde.com>. Website, 2008.
53. Charles H. Knapp and G. Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 24:320–327, August 1976.
54. Chris Kreucher, Mark Morelande, Keith Kastella, and Alfred Hero. Particle filtering for multitarget detection and tracking. In *Aerospace, 2005 IEEE Conference*, March 2005.
55. Guillaume Lathoud, Jean-Marc Odobez, and Daniel Gatica-Perez. Av16.3: an audio-visual corpus for speaker localization and tracking. In *Proceedings of the 2004 MLMI Workshop*, 2004.
56. Eric Lehmann. *Particle Filtering Methods for Acoustic Source Localisation and Tracking*. PhD thesis, Australian National University, July 2004.
57. Eric Lehmann, Darren Ward, and Robert Williamson. Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room. *Proc. ICASSP 2003*, 2003.
58. Eric A. Lehmann and Anders M. Johansson. Experimental performance assessment of a particle filter with voice activity data fusion for acoustic speaker tracking. In *Proceedings of the IEEE Nordic Signal Processing Symposium (NORSIG'06)*, Reykjavik, Iceland, June 2006.

59. Eric A. Lehmann and Anders M. Johansson. Dynamics models for acoustic speaker tracking preliminary results. Technical report, Western Australian Telecommunications Research Institute, August 2007. Technical Report PRJ-NICTA-PM-023.
60. Eric A. Lehmann and Anders M. Johansson. Particle filter with integrated voice activity detection for acoustic source tracking. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007. Article ID 50870, 11 pages.
61. Eric A. Lehmann, Anders M. Johansson, and Sven Nordholm. Modeling of motion dynamics and its influence on the performance of a particle filter for acoustic speaker tracking. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '07)*, pages 98–101, October 2007.
62. Eric A. Lehmann and Robert C. Williamson. Importance sampling particle filter for robust acoustic source localization and tracking in reverberant environments. In *Proceedings of the Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*, pages C7–C8, Rutgers University, Piscataway, New Jersey, USA, March 2005.
63. Eric A. Lehmann and Robert C. Williamson. Particle filter design using importance sampling for acoustic source localisation and tracking in reverberant environments. *EURASIP Journal on Applied Signal Processing*, 2006.
64. Wing-Kin Ma, Ba-Ngu Vo, Sumeetpal Singh, and Adrian Baddeley. Tracking an unknown time-varying number of speakers using tdoa measurements: A random finite set approach. *IEEE Transactions on Signal Processing*, 54, September 2006.
65. M. Khademul Islam Molla, Keikichi Hirose, and Nobuaki Minematsu. Localization based separation of mixed audio signals with binary masking of hilbert spectrum. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2006)*, volume 5, May 2006.
66. Brian C.J. Moore. *An Introduction to the Psychology of Hearing*. Academic, London, second edition, 1982.
67. Mark Moreland, Chris Kreucher, and Keith Kastella. A Bayesian approach to multiple target detection and tracking. *IEEE Transactions on Signal Processing*, 55:1589–1604, May 2007.
68. Masamitsu Murase, Shunichi Yamamoto, Jean-Marc Valin, Kazuhiro Nakadai, Kentaro Yamada, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno.

- Multiple moving speaker tracking by microphone array on mobile robot. In *Proceedings of Interspeech 2005*, pages 249–252, September 2005.
69. Kazuhiro Nakadai, Ken ichi Hidai, Hiroshi G. Okuno, and Hiroaki Kitano. Real-time multiple speaker tracking by multi-modal integration for mobile robots. In *Proceedings of the European Conference on Speech Communication and Technology 2001 (Eurospeech)*, September 2001.
 70. Kazuhiro Nakadai, Hirofumi Nakajima, Masamitsu Murase, Satoshi Kaijiri, Kentaro Yamada, Takahiro Nakamura, Yuji Hasegawa, Hiroshi G. Okuno, and Hiroshi Tsujino. Robust tracking of multiple sound sources by spatial integration of room and robot microphone arrays. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2006*, 2006.
 71. Kazuhiro Nakadai, Hirofumi Nakajima, Masamitsu Murase, Hiroshi G. Okuno, Yuji Hasegawa, and Hiroshi Tsujino. Real-time tracking of multiple sound sources by integration of in-room and robot-embedded microphone arrays. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, October 2006.
 72. W. Ng, J. Li, S. Godsill, and S.K. Pang. Multitarget initiation, tracking and termination using bayesian monte carlo methods. *The Computer Journal*, 50(6), 2007.
 73. Matt Orton and William Fitzgerald. A bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Trans. on Signal Processing*, 50(2):216–223, 2002.
 74. Nam Trung Pham, Weimin Huang, and S. H. Ong. Tracking multiple speakers using cphd filter. In *Proceedings of the 15th international conference on Multimedia ACMM*, 2007.
 75. PhaseSpace. <http://www.phasespace.com>. Website, 2008.
 76. Ilyas Potamitis, Huimin Chen, and George Tremoulis. Tracking of multiple moving speakers with multiple microphone arrays. *Speech and Audio Processing, IEEE Transactions on*, 12(5):520–529, September 2004.
 77. Vikas C. Raykar, B. Yegnanarayana, S.R.M. Prasanna, and Ramani Duraiswami. Speaker localization using excitation source information in speech. *Speech and Audio Processing, IEEE Transactions on*, 13(5):751–761, Sept. 2005.
 78. Scott Rickard and Ozgur Yilmaz. On the approximate w-disjoint orthogonality of speech. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2002*, volume 1, pages 529–532, 2002.

79. Justinian Rosca, Joseph Ruanaidh, Alexander Jourjine, and Scott Rickard. Broadband direction-of-arrival estimation based on second order statistics. In *Proceedings of NIPS99*, 1999.
80. Richard Roy and Thomas Kailath. Esprit - estimation of signal parameters via rotational invariance techniques. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(7), July 1989.
81. J. M. Sachar and H. F. Silverman. A baseline algorithm for estimating talker orientation using acoustical data from a large aperture microphone array. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 04)*, volume 4, page 6568, May 2004.
82. D.J. Salmond and H. Birch. A particle filter for track-before-detect. In *American Control Conference, 2001. Proceedings of the*, volume 5, 2001.
83. Harvey F. Silverman. personal communication, October 2007.
84. Harvey F. Silverman, William R. Patterson III, and James L. Flanagan. The huge microphone array. *IEEE Concurrency*, 6(4):36–46, 1998.
85. Harvey F. Silverman and Joshua M. Sachar. The time-delay graph and the delayogram - new visualizations for time delay. *IEEE Signal Processing Letters*, 12(4), April 2005.
86. Douglas E. Sturim, Michael S. Brandstein, and Harvey F. Silverman. Tracking multiple talkers using microphone-array measurements. In *Proceedings of ICASSP-1997*, April 1997.
87. Vincent Y. F. Tan and Cdric Fvotte. A study of the effect of source sparsity for various transforms on blind audio source separation performance. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'05)*, November 2005.
88. Heinz Teutsch and Walter Kellermann. Acoustic source detection and localization based on wavefield decomposition using circular microphone arrays. *Journal of the Acoustical Society of America*, 120, November 2005.
89. Jaco Vermaak. Acoustic source localisation. Technical report, Microsoft Research, 2002.
90. Jaco Vermaak and Andrew Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. *Proc. ICASSP 2001*, 5:3021–3024, 2001.
91. Jaco Vermaak, M. Gangnet, Andrew Blake, and P. Prez. Sequential Monte Carlo fusion of sound and vision for speaker tracking. In *Proceedings of International Conference on Computer Vision 2001 (ICCV)*, 2001.

92. Ba-Ngu Vo, Wing-Kin Ma, and Sumeetpal Singh. Localizing an unknown time-varying number of speakers: A bayesian random finite set approach. *Proc. ICASSP 2005*, March 2005.
93. Darren B. Ward, Eric A. Lehmann, and Robert C. Williamson. Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Transactions on Speech and Audio Processing*, 11(6):826–836, November 2003.
94. E. Weinstein, K. Steele, A. Agarwal, and J. Glass. Loud: A 1020-node microphone array and acoustic beamformer. In *International Congress on Sound and Vibration (ICSV)*, july 2007.
95. Ozgur Yilmaz and Scott Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004.
96. Ying Yu and Harvey F. Silverman. An improved tdoa-based location estimation algorithm for large aperture microphone arrays. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP04)*, May 2004.
97. Dmitry N. Zotkin and Ramani Duraiswami. Accelerated speech source localization via a hierarchical search of steered response power. *IEEE Trans. on Speech and Audio Processing*, 12(5):499–508, September 2004.