Classification:
Public

Document Type:
Scientific Report

Document Reference:
PRJ-NICTA-PM-008

Status:
Final version

Revision:
1.1

Date:
December 12, 2006

A joint venture between:
The University of Western Australia &
Curtin University of Technology

E-mail: ericl@watri.org.au
Tel: +61 (0)8 6488 4642

Title:

Particle Filter with Integrated Voice Activity Detection for Acoustic Source Tracking

Author(s):

Eric A. Lehmann and Anders M. Johansson

Document History:

Revision	Date	Comments
1.0	31-10-06	Initial revision
1.1	12-12-06	Formatted to NICTA/WATRI technical report guidelines

Contents

Abstract **2**

1 Introduction **3**

2 Bayesian Filtering for Target Tracking **3**
2.1 ASLT Problem Definition . . . 3
2.2 State-Space Filtering . . . 4
2.3 Sequential Monte Carlo Approach . . . 4

3 PF for Acoustic Source Tracking **5**
3.1 Target Dynamics . . . 6
3.2 Likelihood Function . . . 6
3.3 PF Algorithm Outputs . . . 8

4 Voice Activity Detection **9**
4.1 SNR Estimation . . . 9
4.2 Statistical Detection . . . 10

5 Fusion of VAD Measurements **10**

6 Implementation **12**
6.1 Hardware configuration. . . . 12
6.2 Software. . . . 12

7 Experimental Results **13**
7.1 Assessment Parameters . . . 15
7.2 Image Method Simulations . . . 15
7.3 Real Audio Tracking . . . 17
7.4 Real-Time Audio Tracking . . . 19

8 Conclusion and Future Work **20**

Acknowledgement **21**

Bibliography **22**

Abstract

In noisy and reverberant environments, the problem of acoustic source localisation and tracking (ASLT) using an array of microphones presents a number of challenging difficulties. One of the main issues when considering real-world situations involving human speakers is the temporally discontinuous nature of speech signals: the presence of silence gaps in the speech can easily misguide the tracking algorithm, even in practical environments with low to moderate noise and reverberation levels. A natural extension of currently available sound source tracking algorithms is the integration of a voice activity detection (VAD) scheme. In this report, we describe a new ASLT algorithm based on a particle filtering (PF) approach where VAD measurements are fused within the statistical framework of the PF implementation. Tracking accuracy results for the proposed method are presented on the basis of synthetic audio samples generated with the image method, furthermore performance results obtained with a real-time implementation of the algorithm, and using real audio data recorded in a reverberant room, are presented. Compared to a previously proposed PF algorithm, the experimental results demonstrate the improved robustness of the method described in this work when tracking sources emitting real-world speech signals, which typically involve significant silence gaps between utterances.

1 Introduction

The concept of speaker localisation and tracking using an array of acoustic sensors has become an increasingly important field of research over the last few years [10, 20, 21]. Typical applications such as teleconferencing, automated multi-media capture, smart meeting rooms and lecture theatres, etc., are fast becoming an engineering reality. This in turns requires the development of increasingly sophisticated algorithms to deal efficiently with problems related to background noise and acoustic reverberation during the speech acquisition process.

A major part of the literature on the specific topic of acoustic source localisation and tracking (ASLT) typically focuses on implementations involving human speakers [4, 5, 8–10, 13, 17, 20]. One of the major difficulties in a practical implementation of ASLT for speech-based applications lies in the nonstationary character of typical speech signals, with potentially significant silence periods existing between separate utterances. During such silence gaps, currently available ASLT methods will usually keep updating the source location estimates as if the speaker was still active. The algorithm is therefore likely to momentarily lose track of the true source position since the updates are then based solely on disturbance sources such as reverberation and background noise, whose influence might be quite significant in practical situations. Whether the algorithm recovers from this momentary tracking error or not (and how fast the recovery process occurs) is mainly determined by how long the silence gap lasts. Consequently, existing works on acoustic source tracking either implicitly rely on the fact that silence periods in the considered speech signal remain relatively short [9, 17, 20, 21], or alternatively assume a stationary source signal (such as in vehicle tracking applications [6, 18]).

In the present work, we address this specific problem by presenting a new algorithm for ASLT that includes the data obtained from a voice activity detector (VAD) as an integral part of the target-tracking process. To the best of our knowledge, this fusion problem is yet to be considered in the acoustic source tracking literature, despite the fact that this approach can be regarded as a natural extension of currently existing ASLT algorithms developed for speech-based applications. In this report, we use an approach based on a particle filtering (PF) concept similar to that used previously in [21], and show how the VAD measurement modality can be efficiently fused within the statistical framework of sequential Monte Carlo (SMC) methods. Rather than simply using this additional measurement in the derivation of a mixed-mode likelihood, we consider the VAD data as a prior probability that the source localisation observations originate from the true source. As a result, the proposed particle filter, denoted PF-VAD, integrates the VAD data at a low level in the PF algorithm development. It hence benefits from the various advantages inherent to SMC methods (nonlinear and non-Gaussian processing) and is able to deal efficiently with significant gaps in the speech signal.

This report is organised as follows. The next chapter briefly reviews the basic principles of Bayesian filtering (state-space approach), and in Chapter 3, we derive the theoretical concepts required by the PF methodology on the basis of the specific ASLT problem definition. The derivation of this statistical framework then allows the integration of VAD measurements within the PF algorithm. Chapter 4 contains a review of the VAD scheme used in this work (based on [7]), and we then update this basic scheme for the specific speaker tracking purpose considered in this work. We further derive three different types of VAD outputs (considering both hard and soft decisions) to be used within the PF algorithm, and the proposed PF-VAD method is finally presented in Chapter 5. A real-time implementation of the algorithm is described in Chapter 6 followed by a performance assessment in Chapter 7, which also includes the results obtained with a PF method previously developed in [21] for comparison purposes.

2 Bayesian Filtering for Target Tracking

2.1 ASLT Problem Definition

Consider an array of M acoustic sensors distributed at known locations in a reverberant environment with known acoustic wave propagation speed c. For a typical application of speaker tracking, the microphones are usually scattered around the considered enclosure in such a way that the acoustic source always remains within the interior of the sensor array. This type of setup allows for a better localisation accuracy compared to, for instance, a concentrated linear or circular array. Assuming a single sound source, the problem consists in estimating the location of this "target" in the current coordinate system based on the

signals $f_m(t)$, $m \in \{1, \ldots, M\}$, provided by the microphones. It is further assumed that the sensor signals are sampled in time and decomposed into a series of successive frames $k = 1, 2, \ldots$, of equal length L before being processed. The problem is then considered on the basis of the discrete-time variable k.

Note that the derivations presented in this work focus on a two-dimensional problem setting where the height of the source is considered known, or of no particular importance. The acoustic sensors are therefore placed at a constant height in the enclosure, and the aim is to ultimately provide a two-dimensional estimate of the source location on this horizontal plane only. The following developments can however be easily generalised to include the third dimension if necessary.

2.2 State-Space Filtering

Assuming that a Cartesian coordinate system with known origin has been defined for the considered tracking problem, let $\mathbf{X}_k$ represent the state variable for time frame k, corresponding to the position $[x_k \;\; y_k]^{\mathrm{T}}$ and velocity $[\dot{x}_k \;\; \dot{y}_k]^{\mathrm{T}}$ of the target in the state space:

$$\mathbf{X}_k = [x_k \;\; y_k \;\; \dot{x}_k \;\; \dot{y}_k]^{\mathrm{T}}.$$

At any time step k, each microphone in the array delivers a frame of audio signal which can be processed using some localisation technique such as, for instance, steered beamforming (SBF) or time-delay estimation (TDE). Let $\mathbf{Y}_k$ denote the observation variable (measurement) which, in the case of ASLT, typically corresponds to the localisation information resulting from this preprocessing of the audio signals.

Using a Bayesian filtering approach and assuming Markovian dynamics, this system can be globally represented by means of the following two equations [2]:

$$\mathbf{X}_k = g(\mathbf{X}_{k-1}, \mathbf{u}_k)\,, \tag{1a}$$

$$\mathbf{Y}_k = h(\mathbf{X}_k, \mathbf{v}_k)\,, \tag{1b}$$

where $g(\cdot)$ and $h(\cdot)$ are possibly nonlinear functions, and $\mathbf{u}_k$ and $\mathbf{v}_k$ are possibly non-Gaussian noise variables. Ultimately, one would like to compute the so-called posterior probability density function (PDF) $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$, where $\mathbf{Y}_{1:k} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_k\}$ represents the concatenation of all measurements up to time k. The density $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$ contains all the statistical information available regarding the current condition of the state variable $\mathbf{X}_k$, and an estimate $\widehat{\mathbf{X}}_k$ of the state then follows, for instance, as the mean or the mode of this PDF.

The solution to this Bayesian filtering problem consists of the following two steps of prediction and update [3]. Assuming that the posterior density $p(\mathbf{X}_{k-1}|\mathbf{Y}_{1:k-1})$ is known at time $k-1$, the posterior PDF $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$ for the current time step k can be computed using the following equations:

$$p(\mathbf{X}_k|\mathbf{Y}_{1:k-1}) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})\, p(\mathbf{X}_{k-1}|\mathbf{Y}_{1:k-1})\, \mathrm{d}\mathbf{X}_{k-1}\,,$$

$$p(\mathbf{X}_k|\mathbf{Y}_{1:k}) \propto p(\mathbf{Y}_k|\mathbf{X}_k)\, p(\mathbf{X}_k|\mathbf{Y}_{1:k-1})\,,$$

where $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ is the transition density, and $p(\mathbf{Y}_k|\mathbf{X}_k)$ is the so-called likelihood function.

2.3 Sequential Monte Carlo Approach

Particle filtering is an approximation technique that solves the Bayesian filtering problem by representing the posterior density as a set of N samples of the state space $\mathbf{X}_k^{(n)}$ (particles) with associated weights $w_k^{(n)}$, $n \in \{1, \ldots, N\}$ (see e.g. [3]). Originally proposed by Gordon *et al.* in [11], the so-called bootstrap algorithm is an attractive PF variant due to its simplicity of implementation and low computational demands, despite being a sub-optimal PF representative. Assuming that the set of particles and weights $\{(\mathbf{X}_{k-1}^{(n)}, w_{k-1}^{(n)})\}_{n=1}^N$ is a discrete representation of the posterior density at time $k-1$, $p(\mathbf{X}_{k-1}|\mathbf{Y}_{1:k-1})$, the generic iteration update for the bootstrap PF algorithm is given in Algorithm 1. Following this iteration, the new set of particles and weights $\{(\mathbf{X}_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is approximately distributed as the current posterior density $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$. The sample set approximation of the posterior PDF can then be obtained using:

$$p(\mathbf{X}_k|\mathbf{Y}_{1:k}) \approx \sum_{n=1}^{N} w_k^{(n)}\, \delta(\mathbf{X}_k - \mathbf{X}_k^{(n)})\,,$$

Assumption: at time $k-1$, the set of particles $\mathbf{X}_{k-1}^{(n)}$ and weights $w_{k-1}^{(n)}$, $n \in \{1,\ldots,N\}$, is a discrete representa-tion of the posterior $p(\mathbf{X}_{k-1}|\mathbf{Y}_{1:k-1})$.

Iteration: given the observation $\mathbf{Y}_k$ obtained at the current time k, update the particle set as follows:

1. *Prediction:* propagate the particles through the transition equation, $\widetilde{\mathbf{X}}_k^{(n)} = g(\mathbf{X}_{k-1}^{(n)}, \mathbf{u}_k)$.
2. *Update:* assign each particle a likelihood weight, $\tilde{w}_k^{(n)} = w_{k-1}^{(n)} \cdot p(\mathbf{Y}_k|\widetilde{\mathbf{X}}_k^{(n)})$, then normalize the weights:
$$w_k^{(n)} = \tilde{w}_k^{(n)} \cdot \Big(\sum_{i=1}^{N} \tilde{w}_k^{(i)}\Big)^{-1}.$$
3. *Resampling:* compute the effective sample size,
$$N_{\text{eff}} = \Big(\sum_{n=1}^{N} \big(w_k^{(n)}\big)^2\Big)^{-1}.$$
If N_{eff} is above some pre-defined threshold N_{thr}, simply define $\mathbf{X}_k^{(n)} = \widetilde{\mathbf{X}}_k^{(n)}$, $\forall n$. Otherwise, draw N new samples $\mathbf{X}_k^{(n)}$, $n \in \{1,\ldots,N\}$, from the existing set of particles $\{\widetilde{\mathbf{X}}_k^{(i)}\}_{i=1}^N$ according to their weights $w_k^{(i)}$, then reset the weights to uniform values: $w_k^{(n)} = 1/N$, $\forall n$.

Result: the set $\{(\mathbf{X}_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is approximately distributed as the posterior density $p(\mathbf{X}_k|\mathbf{Y}_{1:k})$.

Alg. 1. Generic bootstrap PF algorithm.

where $\delta(\cdot)$ is the Dirac delta function, and an estimate $\widehat{\mathbf{X}}_k$ of the target state for the current time step k follows as:

$$\begin{aligned}\widehat{\mathbf{X}}_k &= \int \mathbf{X}_k \cdot p(\mathbf{X}_k|\mathbf{Y}_{1:k})\,\mathrm{d}\mathbf{X}_k \\ &\approx \sum_{n=1}^{N} w_k^{(n)} \mathbf{X}_k^{(n)}. \end{aligned} \tag{2}$$

It can be shown that the variance of the weights $w_k^{(n)}$ can only increase over time, which decreases the overall accuracy of the algorithm. This constitutes the so-called degeneracy problem, known to affect any PF implementation. The conditional resampling step in Algorithm 1 is introduced as way to mitigate these effects. This resampling process can be easily implemented using a scheme based on a cumulative weight function, see e.g. [11]. Alternatively, several other resampling methods are also available from the particle filtering literature, see e.g. [3].

The main disadvantage of the bootstrap algorithm is that during the prediction step, the particles are relocated in the state space without knowledge of the current measurement $\mathbf{Y}_k$. Some regions of the state space with potentially high posterior likelihood might hence be omitted during the iteration. Despite this drawback, this algorithm constitutes a good basis for the evaluation of particle filtering methods in the context of the current application, keeping in mind that the use of a more elaborate PF method would also increase the accuracy of the resulting tracking algorithm.

3 PF for Acoustic Source Tracking

The particle filtering concepts presented in this chapter are based upon those derived previously in [21], where a sequential estimation framework was developed for the specific problem of acoustic source

localisation and tracking. More information on this topic can be found in this publication and the references cited therein if necessary.

From Algorithm 1, it can be seen that the particle filtering method involves the definition of two important concepts: the source dynamics (through the transition function $g(\cdot)$) and the likelihood function $p(\mathbf{Y}_k|\mathbf{X}_k)$, which are derived in the sequel.

3.1 Target Dynamics

In order to remain consistent with previous literature [20, 21], a Langevin process is used to model the target dynamics in (1a). This model is typically used to characterise various types of stochastic motion, and it has proved to be a good choice for acoustic speaker tracking. The source motion in each of the Cartesian coordinates is assumed to be an independent first-order process, which can be described by the following equation:

$$\mathbf{X}_k = \begin{bmatrix} 1 & 0 & aT_{\mathrm{U}} & 0 \\ 0 & 1 & 0 & aT_{\mathrm{U}} \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix} \cdot \mathbf{X}_{k-1} + \begin{bmatrix} bT_{\mathrm{U}} & 0 \\ 0 & bT_{\mathrm{U}} \\ b & 0 \\ 0 & b \end{bmatrix} \cdot \mathbf{u}_k \,, \tag{3a}$$

with the noise variable

$$\mathbf{u}_k \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right), \tag{3b}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the density of a multi-dimensional Gaussian random variable with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The parameter T_{U} corresponds to the time interval separating two consecutive updates of the particle filter, and the other model parameters in (3) are defined as

$$a = \exp(-\beta\, T_{\mathrm{U}}) \,,$$
$$b = \bar{v}\, \sqrt{1 - a^2} \,,$$

with $\bar{v}$ the steady-state velocity parameter and β the rate constant.

3.2 Likelihood Function[1]

Experimental results from previous research carried out on particle filtering for ASLT have shown that steered beamforming (SBF) delivers an improved tracking performance compared to TDE-based methods [16, 21]. Hence, the SBF principle is here also used as a basis for the derivation of the likelihood function. With $F_m(\omega) = \mathcal{F}\{f_m(t)\}$ the Fourier transform of the signal data from the m-th sensor, and with $\|\cdot\|$ denoting the Euclidean norm, the output $\mathcal{P}(\boldsymbol{\ell})$ of a delay-and-sum beamformer steered to the location $\boldsymbol{\ell} = [x \;\; y]^{\mathrm{T}}$ is given as

$$\mathcal{P}(\boldsymbol{\ell}) = \int_{\Omega} \Big| \sum_{m=1}^{M} W_m(\omega)\, F_m(\omega)\, \mathrm{e}^{j\omega \|\boldsymbol{\ell} - \boldsymbol{\ell}_m\|/c} \Big|^2 \, \mathrm{d}\omega \,, \tag{4}$$

where $\boldsymbol{\ell}_m = [x_m \;\; y_m]^{\mathrm{T}}$ is the known position of the m-th microphone, $W_m(\cdot)$ is a frequency weighting term, and Ω corresponds to the frequency range of interest, which is typically defined as $\Omega = \{\omega \,|\, 2\pi \cdot 300\,\mathrm{Hz} \leqslant \omega \leqslant 2\pi \cdot 3000\,\mathrm{Hz}\}$ for speech processing applications. In the following, the term $W_m(\cdot)$ is computed according to the PHAT (phase transform) weighting [14], for $m \in \{1, \ldots, M\}$:

$$W_m(\omega) = \big|F_m(\omega)\big|^{-1}. \tag{5}$$

For a given state $\mathbf{X}$, the likelihood function $p(\mathbf{Y}|\mathbf{X})$ measures the probability of receiving the data $\mathbf{Y}$. The SBF formula given in (4) effectively measures the level of acoustic energy that originates from a given focus location. The likelihood function should hence be chosen to reflect the fact that peaks in the SBF output $\mathcal{P}(\cdot)$ correspond to likely source locations, as well as the fact that, occasionally, there may be no peak in the SBF output corresponding to the true source due, for instance, to the effects of disturbances

[1] For clarity, the frame subindex k is omitted in this chapter, implicitly assuming that all the variables of interest refer to the current frame of data k.

such as reverberation. The position of the peaks may also have slight errors due to noise or inaccurate sensor calibration. Based on these considerations, one approach to defining the likelihood function is to first select the positions $\hat{\boldsymbol{\ell}}_\theta$, $\theta \in \{1, \ldots, \Theta\}$, of the Θ largest local maxima in the current SBF output. The generic observation variable $\mathbf{Y}$ is then typically defined as the set containing the selected SBF peak locations:

$$\mathbf{Y} \triangleq \{\hat{\boldsymbol{\ell}}_1, \ldots, \hat{\boldsymbol{\ell}}_\Theta\}\,, \tag{6}$$

and the following $\Theta + 1$ hypotheses can be considered:

$$\begin{aligned} &\mathcal{H}_\theta : \text{SBF peak at location } \hat{\boldsymbol{\ell}}_\theta \text{ is due to true source,} \\ &\mathcal{H}_0 : \text{no peak in the SBF output is due to true source,} \end{aligned}$$

with $\theta \in \{1, \ldots, \Theta\}$. The likelihood function is then given as follows:

$$p(\mathbf{Y}|\mathbf{X}) = \sum_{i=0}^{\Theta} q_i \cdot p(\mathbf{Y}|\mathbf{X}, \mathcal{H}_i)\,, \tag{7}$$

with $q_i = p(\mathcal{H}_i|\mathbf{X})$, $i \in \{0, \ldots, \Theta\}$, the prior probabilities of the hypotheses. Without prior knowledge regarding the occurrence of each hypothesis, these probabilities are usually assumed equal and independent of the source location:

$$q_\theta = \frac{1 - q_0}{\Theta}\,, \quad \theta \in \{1, \ldots, \Theta\}\,.$$

Assuming statistical independence between the different peak locations in the SBF measurement, the conditional terms on the right-hand side of (7) are given as follows:

$$p(\mathbf{Y}|\mathbf{X}, \mathcal{H}_i) = \prod_{\theta=1}^{\Theta} p\big(\hat{\boldsymbol{\ell}}_\theta|\mathbf{X}, \mathcal{H}_i\big)\,, \quad i \in \{0, \ldots, \Theta\}\,. \tag{8}$$

In a diffuse sound field comprising many different frequency components, such as the sound field resulting from reverberation, the energy density can be assumed uniform throughout the considered enclosure [22]. This means that given hypothesis $\mathcal{H}_0$, maximising the SBF output will result in a random location distributed uniformly across the state space. Given $\mathcal{H}_\theta$, $\theta \neq 0$, the likelihood of a measurement originating from the source is typically modeled as a Gaussian PDF with variance $\sigma^2_{\mathbf{Y}}$, to account for measurement and calibration errors. Thus, with $\mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denoting a Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\boldsymbol{\xi}$, the likelihood for each SBF peak can be defined as follows:

$$p\big(\hat{\boldsymbol{\ell}}_\theta|\mathbf{X}, \mathcal{H}_i\big) = \begin{cases} \mathcal{N}\big(\boldsymbol{\ell}_\mathbf{X}; \hat{\boldsymbol{\ell}}_\theta, \sigma^2_{\mathbf{Y}}\mathbf{I}\big) & \text{if } \theta = i, \\ \mathcal{U}_\mathcal{D}(\boldsymbol{\ell}_\mathbf{X}) & \text{otherwise,} \end{cases} \tag{9}$$

where $\boldsymbol{\ell}_\mathbf{X} = [x \;\; y]^\mathrm{T}$ corresponds to the top half of the state vector $\mathbf{X}$, $\mathbf{I}$ is the 2×2 identity matrix, and with $\mathcal{U}_\mathcal{D}(\cdot)$ the uniform PDF over the considered enclosure domain $\mathcal{D} = \{(x, y)\,|\,x_{\min} \leqslant x \leqslant x_{\max},\, y_{\min} \leqslant y \leqslant y_{\max}\}$.

The derivations presented so far suffer from a major drawback: the SBF output has to be computed across the entire domain $\mathcal{D}$ in order to find Θ local maxima $\hat{\boldsymbol{\ell}}_\theta$, which leads to a considerable computational load in practical implementations. One approach that circumvents this drawback is based on the concept of a "pseudo-likelihood", as introduced previously in [21]. This concept relies on the idea that the SBF output $\mathcal{P}(\cdot)$ itself can be used as a measure of likelihood. Adopting this approach implicitly reduces the number of hypotheses to the following two events:

$$\begin{aligned} &\mathcal{H}_0 : \text{SBF measurement originates from clutter,} \\ &\mathcal{H}_1 : \text{SBF measurement originates from true source,} \end{aligned} \tag{10}$$

with respective prior probabilities $q_0 = p(\mathcal{H}_0|\mathbf{X})$ and $q_1 = p(\mathcal{H}_1|\mathbf{X}) = 1 - q_0$. Note also that the pseudo-likelihood approach implicitly redefines the observation variable $\mathbf{Y}$ as the SBF output function $\mathcal{P}(\cdot)$ itself; $\mathbf{Y}$ hence does not correspond to a set of SBF peaks as given in (6) anymore. On the basis of (7), (8) and (9), the new likelihood function can be derived as

$$p(\mathbf{Y}|\mathbf{X}) = q_0 \cdot \mathcal{U}_\mathcal{D}(\boldsymbol{\ell}_\mathbf{X}) + \gamma\,(1 - q_0) \cdot \big(\mathcal{P}(\boldsymbol{\ell}_\mathbf{X})\big)^r, \tag{11}$$

where the nonlinear exponent r is used to help shape the SBF output to make it more amenable to source tracking [21].[2] The parameter γ in (11) is a normalisation constant ensuring that $\mathcal{P}(\cdot)$ is suitable for a use as density function, and computed in theory such that

$$\gamma \cdot \iint_{\mathcal{D}} \big(\mathcal{P}(\boldsymbol{\ell})\big)^r \, \mathrm{d}\boldsymbol{\ell} = 1\,. \tag{12}$$

However, the computation of γ according to (12) here again involves the computation of $\mathcal{P}(\cdot)$ across the entire domain $\mathcal{D}$, which is not desirable. In [21], this issue was solved by defining $q_0 = 0$ and $\gamma = 1$, arguing that the SBF measurements are always positive and that the update step of the PF algorithm would ensure that the particle weights are suitably normalised. In the present work however, a proper normalisation parameter γ in the pseudo-likelihood defined by (11) is necessary, since $q_0 \neq 0$ will be assumed in the following developments. Consequently, we propose a normalisation coefficient based on a different principle. As derived previously, a Gaussian likelihood model would typically first determine the global maximum $\hat{\boldsymbol{\ell}}$ of $\mathcal{P}(\cdot)$, and subsequently define $p(\mathbf{Y}|\mathbf{X})$ as a Gaussian density centered on $\hat{\boldsymbol{\ell}}$ and with a certain variance $\sigma^2_{\mathbf{Y}}$, see (9). For the pseudo-likelihood approach, we hence propose to normalise $\mathcal{P}(\cdot)$ so that its maximum value is equal to the peak value of this Gaussian PDF:

$$\begin{aligned} \gamma \cdot \max_{\boldsymbol{\ell}\in\mathcal{D}} \big\{ \big(\mathcal{P}(\boldsymbol{\ell})\big)^r \big\} &= \max_{\boldsymbol{\ell}\in\mathcal{D}} \big\{ \mathcal{N}(\boldsymbol{\ell}; \hat{\boldsymbol{\ell}}, \sigma^2_{\mathbf{Y}}\mathbf{I}) \big\} \\ &= (2\pi\,\sigma^2_{\mathbf{Y}})^{-1}. \end{aligned} \tag{13}$$

The value of the parameter γ can be derived from (13) as follows. Due to the PHAT weighting in (5), and using the representation $F_m(\omega) = |F_m(\omega)| \cdot \mathrm{e}^{j\phi_m(\omega)}$, the SBF output computed according to (4) becomes

$$\mathcal{P}(\boldsymbol{\ell}) = \int_{\Omega} \Big| \sum_{m=1}^{M} \mathrm{e}^{j\Phi_m(\omega)} \Big|^2 \, \mathrm{d}\omega\,,$$

with $\Phi_m(\omega) = \phi_m(\omega) + \omega \|\boldsymbol{\ell} - \boldsymbol{\ell}_m\|\, c^{-1}$. According to the Cauchy–Schwartz inequality, the SBF output values are thus bounded as follows:

$$\mathcal{P}(\boldsymbol{\ell}) \leqslant \int_{\Omega} \Big(\sum_{m=1}^{M} \big|\mathrm{e}^{j\Phi_m(\omega)}\big| \Big)^2 \mathrm{d}\omega = M^2(\omega_{\max} - \omega_{\min})\,, \tag{14}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the upper and lower limits of the frequency range Ω, respectively. Using the result of (14), the normalisation constant in (13) finally becomes

$$\gamma = \frac{1}{2\pi\,\sigma^2_{\mathbf{Y}}\, M^{2r} (\omega_{\max} - \omega_{\min})^r}.$$

The normalisation process described here ensures that the two PDFs in the mixture likelihood definition of (11) are properly scaled with respect to each other.

3.3 PF Algorithm Outputs

For each frame k of input data, the particle filter delivers the following two outputs. First, an estimate $\hat{\boldsymbol{\ell}}_{\mathbf{X},k}$ of the source position is computed according to (2):

$$\hat{\boldsymbol{\ell}}_{\mathbf{X},k} = \sum_{n=1}^{N} w_k^{(n)} \boldsymbol{\ell}_{\mathbf{X},k}^{(n)}\,,$$

where $\boldsymbol{\ell}_{\mathbf{X},k}^{(n)} = \big[x_k^{(n)} \;\; y_k^{(n)}\big]^{\mathrm{T}}$ corresponds to the location information in the n-th particle vector. The second output is a measure of the confidence level in the PF estimates, which can be obtained by computing the standard deviation of the particle set:

$$\varsigma_k = \sqrt{\sum_{n=1}^{N} w_k^{(n)} \big\| \boldsymbol{\ell}_{\mathbf{X},k}^{(n)} - \hat{\boldsymbol{\ell}}_{\mathbf{X},k} \big\|^2}\,. \tag{15}$$

[2] Using $r > 1$ typically increases the sharpness of the peaks while reducing the background noise variance in the SBF measurements.

The parameter ς_k provides a direct assessment of how reliable the PF considers its current source position estimate to be.

4 Voice Activity Detection

The voice activity detector (VAD) employed here relies on an estimate of the instantaneous signal-to-noise ratio (SNR) in the current block of data [7]. It assumes that the data recorded at the microphones is a combination of the speech signal and noise:

$$f_m(t) \triangleq s_m(t) + v_m(t)\,, \quad m \in \{1, \ldots, M\}\,,$$

where the signal $s_m(\cdot)$ and noise $v_m(\cdot)$ are uncorrelated. It is further assumed that the microphone signals are band-limited and sampled in time.

The scheme works on the basis of the expected noise power spectral density, which is estimated during nonspeech periods. The estimated noise level is then used during periods of speech activity to estimate the SNR from the observed signal. The assumption is that the speaker is active when the signal level is sufficiently higher than the noise level: the speech vs. nonspeech decision is made by comparing the mean SNR to a threshold, where the SNR average is taken over the considered frequency domain. The spectral resolution is defined to be lower than the frame length in order to decrease the variance of the signal power estimates. The specific application considered in this work makes it possible to reduce the variance further by averaging over multiple microphones. The frame length L is chosen such that the propagation delay to the different microphones does not impact significantly on the power estimate.

4.1 SNR Estimation

The instantaneous, reduced-resolution estimate $P_{f,d}(k)$ of the power spectral density for the d-th frequency band and the k-th frame of data from the microphones is obtained according to

$$P_{f,d}(k) = \frac{1}{M} \sum_{m=1}^{M} \int_{\Omega_d} \varphi(\omega) \left| \frac{1}{L} \sum_{l=kL-L+1}^{kL} f_m(l)\, \mathrm{e}^{jl\omega} \right|^2 \mathrm{d}\omega\,,$$

where the window function $\varphi(\omega)$ is here chosen to de-emphasise the lower frequency range, in order to suppress frequencies with high noise content. The integration regions Ω_d, $d \in \{1, \ldots, D\}$, divide the frequency space into a small number (typically eight) of non-overlapping bands of equal width. The background noise power $P_{v,d}$ is assumed to vary slowly in relation to the speech power. In practice, a time-varying estimate $\widehat{P}_{v,d}(k)$ of $P_{v,d}$ is obtained by averaging $P_{f,d}(\cdot)$ over time during the nonspeech periods detected by the algorithm. An initial estimate of $P_{v,d}$ is typically obtained during a short algorithm initialisation phase, carried out during a period of background noise only.

The instantaneous SNR for frequency band d is calculated according to

$$\psi_d(k) = \frac{P_{f,d}(k)}{P_{v,d}} - 1\,.$$

During nonspeech periods, we have $P_{f,d}(k) \approx P_{v,d}$, and the variance of the instantaneous SNR becomes

$$\begin{aligned} \sigma^2_{v,d} &= \mathbb{E}\big\{ \big(\psi_d(k) - \mathbb{E}\{\psi_d(k)\}\big)^2 \big\} \\ &= \mathbb{E}\big\{\psi^2_d(k)\big\}, \end{aligned}$$

where $\mathbb{E}\{\cdot\}$ represents the statistical expectation. Thus, an estimate $\hat{\sigma}^2_{v,d}(k)$ of the background noise variance can be found by averaging the square of the instantaneous SNR during nonspeech periods.

4.2 Statistical Detection

The speaker is assumed to be active during the k-th frame when the instantaneous SNR $\psi_d(k)$ is higher than a threshold η_d. The threshold can be derived by considering the problem as a hypothesis test:

$$\begin{aligned}\mathcal{H}_0 : \psi_d(k) &= \frac{P_{v,d}(k)}{P_{v,d}} - 1\,,\\ \mathcal{H}_1 : \psi_d(k) &= \frac{P_{v,d}(k) + P_{s,d}(k)}{P_{v,d}} - 1\\ &= \frac{P_{f,d}(k)}{P_{v,d}} - 1\,,\end{aligned}$$

where $P_{s,d}(k)$ and $P_{v,d}(k)$ are the instantaneous speech signal and noise power, respectively, the null hypothesis $\mathcal{H}_0$ denotes nonspeech, and $\mathcal{H}_1$ the alternative.

The PDF for the instantaneous SNR estimates during nonspeech can be defined as

$$p\big(\psi_d(k)|\mathcal{H}_0\big) = \frac{1}{\sqrt{2\pi\sigma_{v,d}^2}} \exp\left(\frac{-\psi_d^2(k)}{2\sigma_{v,d}^2}\right), \tag{16}$$

assuming that the estimates are Gaussian distributed. This assumption is not always correct, but works well as an approximation under real conditions [7]. From (16), the probability of false alarm P_{FA}, i.e., speech reported during nonspeech period, can then be formulated as

$$\begin{aligned}P_{\mathrm{FA}} &= \Pr\big\{\eta_d < \psi_d(k)|\mathcal{H}_0\big\}\\ &= \int_{\eta_d}^{\infty} \frac{1}{\sqrt{2\pi\sigma_{v,d}^2}} \exp\left(\frac{-\psi_d^2(k)}{2\sigma_{v,d}^2}\right) \mathrm{d}\psi_d(k)\,.\end{aligned} \tag{17}$$

By rearranging (17) and solving for η_d we obtain

$$\eta_d = \sqrt{2\sigma_{v,d}^2} \cdot \mathrm{erfc}^{-1}(2P_{\mathrm{FA}})\,,$$

where $\mathrm{erfc}(\cdot)$ is the complementary error function [12]. In a practical implementation, a time-varying estimate $\hat{\eta}_d(k)$ of the threshold is obtained by using the estimated background noise variance $\hat{\sigma}_{v,d}^2(k)$. Finally, the binary VAD decision $\rho(k)$ for speech is made by comparing the mean instantaneous SNR to the mean threshold, where the average is taken over all frequency bands:

$$\rho(k) = \begin{cases} 1 & \text{if } \sum_{d=1}^{D} \psi_d(k) > \sum_{d=1}^{D} \eta_d(k)\,,\\ 0 & \text{otherwise,}\end{cases}$$

where 1 denotes speech and 0 nonspeech.

Note that the operation of the algorithm depends on the state of its own output for determining when to start estimating the background noise power. During the SNR estimation process, a hangover scheme based on a state machine is therefore used in order to reduce the probability of speech entering the background noise estimate [7]. However, if the background noise power changes rapidly, the algorithm may enter a state where it will provide erroneous decisions, which is a limitation inherent to the considered VAD method. Experimental tests have however shown that this happens very rarely in practice, and that the algorithm is able to recover by itself in such cases after a short transitional period.

5 Fusion of VAD Measurements

A straightforward approach to merging different measurement modalities within the PF framework is via the definition of a combined likelihood function. This representation however would fuse both the VAD and SBF measurements at the same algorithmic level, implicitly assuming statistical independence between these two types of observation. In the context of the specific ASLT problem considered in this work, this is not completely justified: intuitively, if the VAD classifies the current frame of data as

nonspeech, the corresponding SBF measurement is likely to be unreliable in terms of source localisation accuracy. We hence adopt a different approach to the fusion problem, as described in the following.

The output of the VAD can be linked to the probability of the hypotheses in (10) in an obvious manner. For instance, considered as an indication of the likelihood that the current SBF observation originates from clutter only, the variable q_0 explicitly measures the probability of the acoustic source being inactive. Likewise, $q_1 = 1 - q_0$ corresponds to the likelihood of the source being active, an estimate of which is delivered by the VAD. Therefore, instead of setting the variable q_0 to a constant value in the design of the algorithm as done in [20, 21], we propose to use a time-varying q_0 parameter based on the output of the VAD as follows:

$$q_0(k) = 1 - \alpha(k)\,, \tag{18}$$

where $\alpha(k) \in [0, 1]$ is derived from the state of the VAD algorithm. The generic algorithm resulting from (18) and from the developments in Chapter 3 will be denoted PF-VAD from here on.

Three different methods for deriving the parameter $\alpha(k)$ form the VAD algorithm are suggested. These are defined as follows:

$$\begin{aligned}
\alpha_{\text{SNR}}(k) &= \frac{2}{\pi} \arctan\left(\overline{\psi}(k)\right), \\
\alpha_{\text{SP}}(k) &= \frac{\overline{P}_v(k) \cdot \overline{\psi}(k)}{\max\limits_{i<k}\left(\alpha_{\text{SP}}(i)\right)}, \\
\alpha_{\text{BIN}}(k) &= \rho(k)\,,
\end{aligned}$$

with the following definitions:

$$\begin{aligned}
\overline{\psi}(k) &= \sqrt{\frac{1}{D}\sum_{d=1}^{D} \psi_d(k)}\,, \\
\overline{P}_v(k) &= \sqrt{\frac{1}{D}\sum_{d=1}^{D} \widehat{P}_{v,d}(k)}\,.
\end{aligned}$$

The first method, i.e., the VAD output $\alpha_{\text{SNR}}(\cdot)$, maps the mean instantaneous SNR gain level (a number between 0 and ∞) to $\alpha(\cdot)$ through bilinear transformation. The reasoning behind this approach is that a high SNR should indicate that the signal received at the microphones contains information useful to the tracking algorithm. The second method, $\alpha_{\text{SP}}(\cdot)$, calculates an estimate of the speech signal level. The normalisation with respect to all previous maximum signal levels is carried out in order to remove the influence of the absolute signal level at the microphones. This approach effectively discards the noise level information and assumes that only the speech signal level information is useful to the tracking algorithm. The last method, $\alpha_{\text{BIN}}(\cdot)$, simply uses the binary output $\rho(\cdot)$ from the VAD as $\alpha(\cdot)$. The "all-or-nothing" approach used by this method potentially discards a substantial amount of useful information. It however still represents an alternative of potential interest, and is included here for the purpose of providing a performance comparison baseline.

Figure 1 shows an example of the different VAD outputs defined above. The curves obtained with these VAD methods will typically differ from each other as a function of the specific noise and reverberation level contained in the input signals. Compared to the binary output $\alpha_{\text{BIN}}(\cdot)$, the use of soft VAD information with $\alpha_{\text{SNR}}(\cdot)$ and $\alpha_{\text{SP}}(\cdot)$ allows the PF to track the source in a more subtle manner. For instance, a VAD output value $0 < \alpha(\cdot) < 1$ effectively indicates that the input signals may be partly corrupted by disturbance sources, and that the current SBF observation might not be fully accurate. The PF can then take account of this fact and use more caution when updating the particle set, and hence, when determining the source location estimate. With the binary VAD output $\alpha_{\text{BIN}}(\cdot)$, the source tracking process is basically turned fully on or off based on $\rho(\cdot)$ (hard decisions), which may not be advantageous when a high level of noise and/or reverberation is present. In the next chapter, results from experimental simulations of the PF-VAD method will determine which one of these three approaches delivers the best tracking performance.

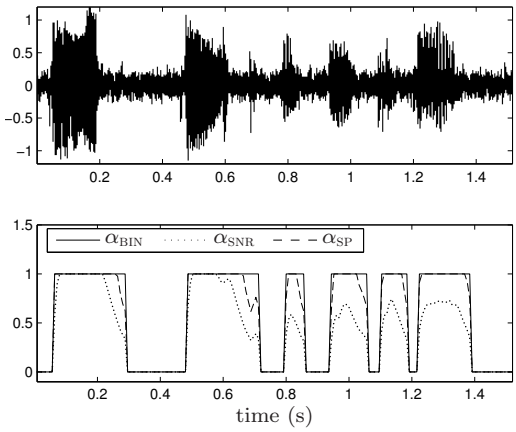

Figure 1. Practical example of three considered VAD methods. *Top plot*: input signal data. *Bottom plot*: resulting VAD outputs.

6 Implementation

The algorithm is implemented in software executed on a standard PC. The implementation uses single precision floating point arithmetic and is written in the C programming language. The microphone array is connected to the PC using a multi-channel analog input/output (I/O) card.

6.1 Hardware configuration.

The PC is a standard IBM-PC, equipped with a 1.8 GHz AMD Athlon processor and 512 MB of memory. The operating system on the PC is Debian GNU/Linux version 3.0. The kernel version is 2.6.8, and is compiled with preemptive multitasking.

The microphone array consists of eight elements which are mounted on a metal fixture in an octagonal pattern (see Figure 9). The microphones are connected to a preamplifier which in turn is connected to the I/O card. The microphone elements, model 2541/PRM902, and the preamplifier, model 2210, are from Larson Davis. The I/O card has 24-bit analog-to-digital converters with built-in anti-aliasing filters and is operated at a sample frequency of 16 kHz. The card is an M-Audio Delta-1010LT.

6.2 Software.

Both the beamformer in the PF and the power spectrum estimation in the VAD can be implemented using an FFT. This makes it possible to further integrate the two sub-algorithms and thereby reduce the computational complexity, see Figure 2. The delay-and-sum beamformer in (4) is the most critical part in terms of computational complexity, since it has to be executed N times per frame for the computation of the likelihood function in (11). Here, it is implemented according to

$$\mathcal{P}(\boldsymbol{\ell}) = \sum_{l=L_0}^{L_1} \left| \sum_{m=1}^{M} G_m(\omega_l)\, \mathrm{e}^{j\omega_l \|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c} \right|^2, \tag{19}$$

where $\omega_l = 2\pi l/L$, L_0 and L_1 define the frequency range of interest, and $G_m(\omega) = W_m(\omega) \cdot F_m(\omega)$. The computational complexity in the inner loop is reduced by rewriting the complex exponential in (19) according to

$$\begin{aligned} \mathrm{e}^{j\omega_{l+1}\|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c} &= \mathrm{e}^{j(\omega_l+\omega_1)\|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c} \\ &= \mathrm{e}^{j\omega_l\|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c} \cdot \mathrm{e}^{j\omega_1\|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c}. \end{aligned} \tag{20}$$

Thus, the exponential term for frequency band l in (19) can be computed recursively using the corresponding term for band $l-1$ multiplied by the constant $\mathrm{e}^{j\omega_1\|\boldsymbol{\ell}-\boldsymbol{\ell}_m\|/c}$, which is calculated only once for each particle. Experiments showed that this code optimization reduced the computational complexity by a factor 10.

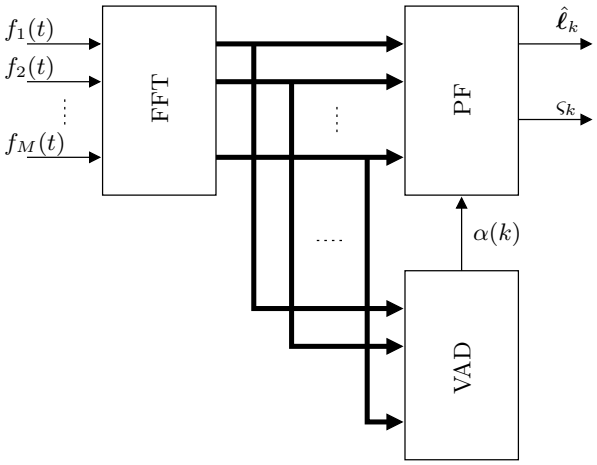

Figure 2. Block diagram showing the integration of the PF and the VAD.

	FFT	VAD	PF	Total	FLOPS
Theoretical	256	21.2	3.34k	3.62k	57.9M
Measured	225	53.7	4.19k	4.47k	71.5M

Table 1. Clock cycles per input sample for each sub-algorithm. The left most column lists the number of Floating Point Operations Per Second (FLOPS) for 16kHz sample frequency.

The computational load for the different parts of the implementation is found in Table 2, where $L_r = L_1 - L_0 + 1$, and R denotes the considered number of room dimensions (typically 2 or 3, for either a two or three-dimensional problem definition). The table is calculated considering a worst case scenario (e.g., when considering the execution of conditional algorithm sections). The computations required for the data fusion is included in the computational complexity for the VAD. The table shows that the bulk of the computations is in the particle filter, signifying that the new algorithm is not much more computationally complex compared to a traditional PF.

The actual number of clock cycles spent in each sub-algorithm has been measured by reading the time stamp counter (TSC) in the CPU. The results are presented in Table 1 along with theoretical number of floating point operations. The theoretical values have been obtained by inserting actual numerical values in the equations given in Table 2. The discrepancy between the measured and real values lies in the function calls overhead, integer operations, pipeline utilization, cache faults and a large number of branch instructions for the VAD. The parameters used for the implementation are $N = 100$, $D = 8$, $L = 512$, $R = 2$, $L_r = 86$ and $M = 8$. The total CPU usage for the algorithm process during execution was around 5%, which is consistent with the 71.5 MFLOPS computational load plus overhead for data acquisition. The results shows that the considered algorithm has a rather low overall computational complexity and can run comfortably on a system made up of widely available and low-cost hardware.

7 Experimental Results

This chapter presents some examples of the tracking results obtained with the proposed PF-VAD algorithm. The various parameters of the PF-VAD implementation were optimised empirically and set to the following values: the number of particles was set to $N = 50$, the effective sample size threshold $N_{\text{thr}} = 37.5$, the standard deviation of the observation density was defined as $\sigma_{\mathbf{Y}} = 0.15\,\text{m}$, and the nonlinear exponent was set to $r = 2$. Following standard definitions (see e.g. [20, 21]), the PF-VAD implementation made use of the propagation model parameters $\bar{v} = 0.8\,\text{m/s}$ and $\beta = 10\,\text{Hz}$. The VAD parameters were defined as $P_{\text{FA}} = 0.03$ and $D = 8$. The audio signals were sampled with a frequency of 16 kHz and processed in non-overlapping frames of 256 samples each.

For comparison purposes, the performance assessment given in this chapter also includes results

Operation	FFT	VAD	PF
Real divisions		$4D + 10$	ML_r
Real additions		$25D + L(M + 1)/2$	$ML_r + N(2L_r + 2MR + 4R + 2)$
Real multiplications		$9D + 3L(M + 1)/2$	$2ML_r + N(2L_r + MR + 5M + 4R + 8)$
Complex to real multiplications			ML_r
Complex additions	$ML/2 \log_2(L/2)$		ML_rN
Complex multiplications	$ML/2 \log_2(L/2)$		$2ML_rN$
Real square root		$D + 4$	$M(L_r + 2N)$
Complex exponential			$N(2M + 1)$
Pseudo random number			$N(R + 2)$

Table 2. Floating point operations per data frame.

from the SBF-PL algorithm, a sound source tracking scheme previously proposed in [21]. The SBF-PL method relies on a particle filtering approach similar to that presented in this work, but does not include any VAD measurements. The reader is referred to [21] for a more detailed description of the SBF-PL implementation.

7.1 Assessment Parameters

The experimental results make use of the following parameters to assess the tracking accuracy of the considered methods. The PF estimation error for the current frame is

$$\varepsilon_k = \left\| \boldsymbol{\ell}_{\mathrm{S},k} - \hat{\boldsymbol{\ell}}_{\mathbf{x},k} \right\| ,$$

where $\boldsymbol{\ell}_{\mathrm{S},k}$ is the ground-truth source position at time k. In order to assess the overall performance of the developed algorithm over a given sample of audio data, the average error is simply computed as

$$\bar{\varepsilon} = \frac{1}{K} \sum_{k=1}^{K} \varepsilon_k ,$$

with K representing the total number of frames in the considered audio sample. The standard deviation parameter ς_k, see (15), is also used here as an overall indication of the PF tracking performance in the following results presentation.

Due to the partially random nature of PF implementations, statistical averaging over a large number D of algorithm runs is used in the results presentation. A parameter of particular interest to ASLT is the percentage of these runs for which the tracking algorithm completely loses track of the target during the simulation, typically due to significant silence gaps in the speech. For each simulation run $d \in \{1, \ldots, D\}$, a track loss parameter is thus defined as:

$$\xi_d = \begin{cases} 1 & \text{if } \left(\sum_{k=K-k^*}^{K} \varepsilon_{k,d} \right) / (k^* - 1) > \delta \, , \\ 0 & \text{otherwise} \, , \end{cases}$$

where $k^* = \lceil 0.5 F_{\mathrm{s}} / L \rceil$. The parameter ξ_d effectively checks if the average estimation error over the last 0.5 s of audio data is smaller than some threshold, set here to $\delta = 0.1$ m, i.e., whether the algorithm is still correctly tracking the target or not at the end of the simulation run. The global track loss percentage (TLP) $\bar{\xi}$ for a given audio sample is then defined as

$$\bar{\xi} = \frac{1}{D} \sum_{d=1}^{D} \xi_d \, .$$

7.2 Image Method Simulations

The proposed PF algorithm was put to the test using synthetic reverberant audio data generated using the image source method [1]. The results presented in this section were obtained using audio data generated with the source trajectory, source signal and microphone setup depicted in Figure 3. The dimension of the enclosure was set to 3 m × 3 m × 2.5 m, and the height of the microphones, as well as that of the source, was defined as 1.5 m.

Figure 4 presents some typical results obtained with the two considered ASLT methods (where PF-VAD uses the speech-based VAD output α_{SP}), using the setup of Figure 3 with a reverberation time $T_{60} \approx 0.1$ s and input SNR of approximately 15 dB. This figure clearly illustrates the most significant outcome of the PF-VAD implementation. Fusing the VAD measurements within the PF framework effectively allows the tracking algorithm to put more emphasis on the considered dynamics model in (3) when spreading the particles during nonspeech periods, while at the same time reducing the importance of the SBF observations due to the fact that no useful information can be derived from them when the speaker is inactive. This consequently allows the PF to keep track of the silent target, and to resume tracking successfully when the speaker becomes active again. This can be distinctly noticed with the consistent increase of the ς_k values for PF-VAD (middle plot in Figure 4) during significant gaps in the speech signal. This specific effect originates from the influence of the VAD measurements on the effective

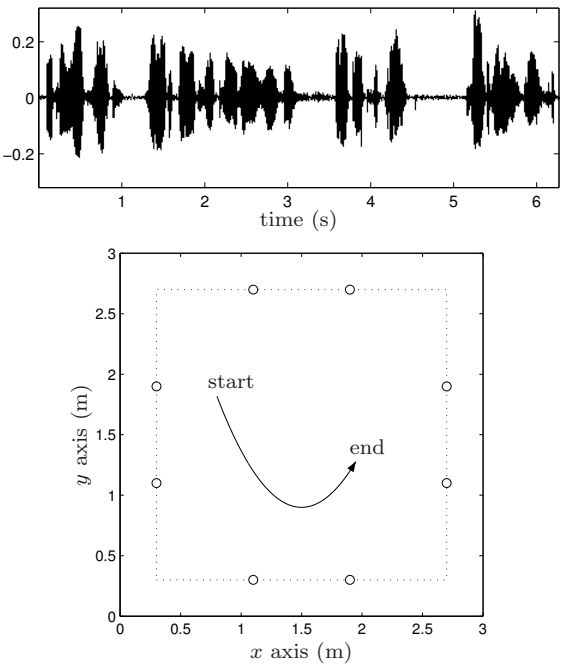

Figure 3. Setup for image method simulations. *Top plot*: source signal. *Bottom plot*: microphone positions (○) and parabolic source trajectory.

sample size parameter N_{eff}. The bottom graph in Figure 5 shows an example of the N_{eff} values computed during one run of PF-VAD, versus time. As described in Step 3 of Algorithm 1, the parameter N_{eff} is reset to N after the resampling stage is carried out, and the result in Figure 5 thus provides an overall view of the resampling frequency. This plot demonstrates how the VAD output "freezes" the N_{eff} value during nonspeech periods, effectively decreasing the occurrence of the particle resampling step, which in turn leads to a spatial evolution of the particles according to the dynamics model only.

As an important consequence of this fact, the standard deviation ς_k delivered by PF-VAD effectively reflects a "true" confidence level, i.e., in keeping with the estimation accuracy, and can be hence directly used as an indication of the reliability of the PF estimates. For instance, an obvious add-on to the PF-VAD method would be to simply discard the PF location estimates whenever ς_k is above a pre-defined threshold.

On the other hand, the more or less constant resampling frequency implemented as part of the SBF-PL method precludes this desired behaviour, meaning that the particles always remain very concentrated spatially. This essentially implies that during nonspeech periods, the SBF-PL particle filter continues its tracking as if the speaker was still active, and is hence much more likely to be driven off-track by the effects of reverberation and additive noise. An example of such a scenario is shown in the bottom plot of Figure 4, where SBF-PL loses track of the speaker at the end of the simulation due to a significant gap in the speech signal.

Figures 6 and 7 present the average tracking results obtained for the proposed PF-VAD algorithm, as well as a comparison with the previously developed SBF-PL method. These plots show the average error $\bar{\varepsilon}$ computed over a range of input SNR values (Figure 6) and reverberation times (Figure 7). Different T_{60} values were achieved by appropriately setting the walls' reflection coefficients in the image method implementation. Statistical averaging was performed due to the random nature of the PF implementation, and the results depicted in these figures represent the average over 100 simulation runs of the considered algorithms, using the above mentioned image method setup.

These results clearly demonstrate the superiority of the proposed PF-VAD algorithm. The SBF-PL method consistently exhibits a larger average error due to track losses occurring as a result of significant gaps in the considered speech signal (see the source signal plotted in Figure 3), which the PF-VAD implementation manages to avoid. Also, it must be kept in mind that the PF-VAD results shown in Figures 6 and 7 correspond to the mean error $\bar{\varepsilon}$ computed over the entire length of the considered audio sample. This typically also includes periods where the PF has a low confidence level in its estimates. As

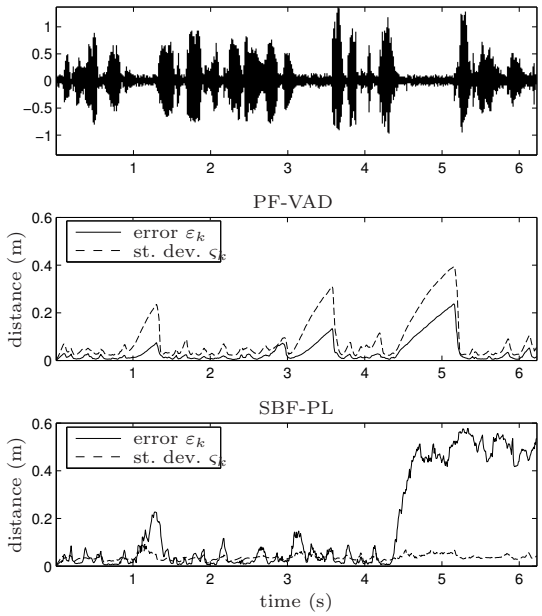

Figure 4. Tracking result examples for two ASLT methods, for $T_{60} \approx 0.1\,\mathrm{s}$ and SNR $\approx 15\,\mathrm{dB}$. *Top plot*: example of microphone signal. *Bottom plots*: estimation error and standard deviation for PF-VAD and SBF-PL (results averaged over 100 simulation runs).

mentioned earlier, the average performance of PF-VAD would improve even further if tracking estimates were discarded when ς_k is above a pre-defined threshold.

In regards to a comparison of the three tested VAD schemes with each other, it can be seen from Figures 6 and 7 that the speech-based VAD scheme α_{SP} generally tends to yield the best overall tracking performance, given the specific test setup considered in this chapter. This result suggests that the most useful information from a tracking point of view relies more on the amount of speech present during a given time frame, rather than the speech-to-noise ratio, which, for instance, may become large despite a small speech signal level in some circumstances.

7.3 Real Audio Tracking

A tracking assessment of algorithm PF-VAD was also conducted using samples of real audio data, recorded in a reverberant environment. The microphone array was set up in a room with dimensions $3.5\,\mathrm{m} \times 3.1\,\mathrm{m} \times 2.2\,\mathrm{m}$, with walls partially covered with sound absorbing foam, leading to a practical reverberation time $T_{60} \approx 0.3\,\mathrm{s}$ (frequency-averaged up to 24 kHz). An array of $M = 8$ omnidirectional microphones were positioned at a height of 1.51 m, in a fashion similar to that depicted in Figure 3. The area spanned by the array was approximately $2.52\,\mathrm{m} \times 2.52\,\mathrm{m}$.

In order to achieve an accurate assessment of the source tracking accuracy, it is necessary to obtain ground-truth measurements of the speaker trajectory during the recordings. A few methods have been proposed and used in previous literature works to this purpose. Some of these methods work on the basis of some sort of mechanical system allowing a loudspeaker to move along a predefined trajectory [19], or by using the location measurements obtained from a different tracking scheme, based e.g. on visual information [15]. These approaches however generate considerable difficulties related, for instance, to synchronisation of different data streams (such as audio and video), the inability to use real human speakers in some cases, and most of all, substantial hardware and software setup costs.

In the present work, a different approach is used for the purpose of acquiring audio data with exact knowledge of the moving sound source position. The ground-truth location data is extracted directly from the recorded audio data by means of a high-accuracy beamformer scanning the considered enclosure. The microphone signals are split into successive frames of 32 ms of data, with a 50% overlapping factor. Each frame is processed using the SBF formula in (4), at first computed on a relatively coarse grid across the entire search space. A coarse estimate of the current source position is obtained as the location maximising

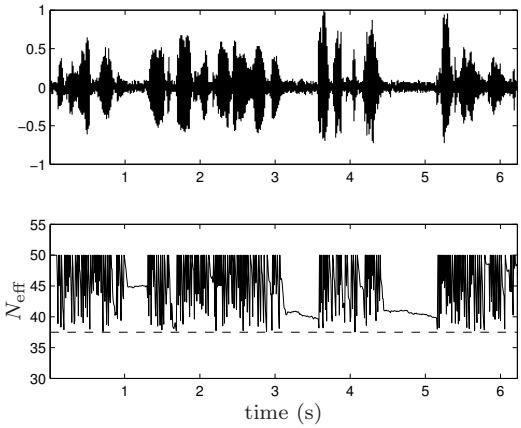

Figure 5. Overview of the resampling frequency during one run of PF-VAD. *Bottom plot*: effective sample size parameter N_{eff} vs. time (dashed line: threshold N_{thr}). *Top plot*: example of input signal used for this simulation.

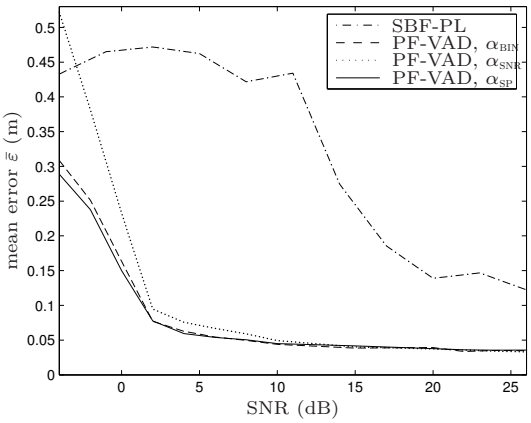

Figure 6. Average tracking error vs. input signal SNR, for $T_{60} \approx 0.1\,\text{s}$ (results averaged over 100 simulation runs).

the SBF output, and this estimate is then refined by considering a high-accuracy grid (uniform 1 mm spacing between grid points) centered around the region of interest.

An approximate knowledge of the overall source path across the room, combined with the use of some voice activity detection scheme, allows the easy discrimination of outliers and yields a series of two-dimensional location data points versus time, as shown in Figure 8. This graph presents the ground-truth measurements results for a sample of audio data recorded in the above mentioned room setup with a male speaker moving freely within the enclosure. Finally, a polynomial approximation can be fitted to the obtained SBF localisation data in order to obtain an estimate of the true source trajectory over the entire audio sample length.

The approach described here has the main advantage of producing source position data which is automatically synchronised to the audio signals, and leads to minimal hardware setup costs. It also allows any kind of sound source to be used (including human speakers) along any arbitrary trajectories, and it has proved to work particularly well for the controlled environment considered in this work (i.e. single-target tracking and low-level additive noise). It is estimated that the ground-truth location data generated with this method are accurate to within a few centimeters of the true source trajectory.

Samples of audio data were recorded for the following array setup. An array of $M = 8$ omnidirectional microphones was set up at a constant height of 1.51 m in a room with dimensions $3.5\,\text{m} \times 3.1\,\text{m} \times 2.2\,\text{m}$, in a square fashion with one sensor pair centered on each side of the square (distance of 0.8 m between the sensors in each pair). The area spanned by the array was $2.52\,\text{m} \times 2.52\,\text{m}$.

Two different types of environment were considered. In the first one, the walls of the enclosure were fully padded with sound-absorbing panels (no padding on the floor and ceiling), leading to a practical

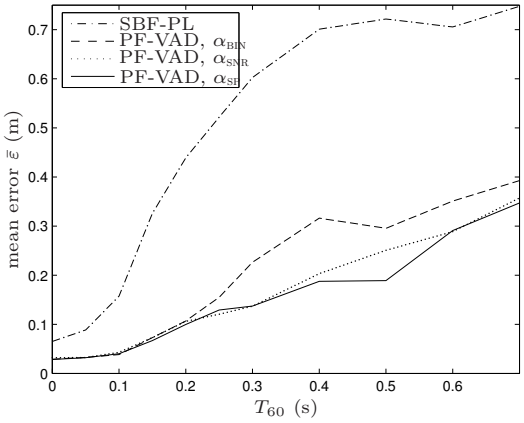

Figure 7. Average tracking error vs. reverberation time T_{60}, with input SNR of about 20 dB (results averaged over 100 simulation runs).

reverberation time $T_{60} \approx 0.27$ s (frequency-averaged up to 24 kHz). In the second environment, parts of the padding were removed, increasing the reverberation to $T_{60} \approx 0.34$ s. In each environment, two different signal sources were used. The first one was a male speaker, moving randomly across the room while uttering a series of sentences separated by silence gaps. The second source was a loudspeaker emitting a female speech signal (also containing silence gaps). The loudspeaker was carried randomly across the room during the recordings in order to simulate a mobile source. For each speaker–environment combination, a total of five recordings were taken, each corresponding to a different trajectory and source signal.

In both environments, background noise was recorded separately (i.e., with no speech source present) by means of a series of loudspeakers emitting either average white Gaussian noise (AWGN) or babble noise. The recordings of these noise signals were then combined additively to the speech signals with varying SNR levels in order to generate the input data to the tracking algorithm. This way of splitting the noise and speech recordings specifically allowed the measurement of ground-truth source location data according to the method described above.

The results presented in Tables 3 and 4 show the average tracking performance obtained with these real-audio recordings for PF-VAD and SBF-PL, respectively. Each result in these tables corresponds to the average computed over 250 values, corresponding to 50 algorithm runs carried out for each of the five audio samples recorded in a given speaker–environment configuration (statistical averaging was performed due to the random nature of PF methods).

A comparison between these tables shows that PF-VAD results are consistently better than those obtained for SBF-PL. Some of the considered tracking scenarios in the above experiments still prove too difficult to deal with for both tested algorithms, yielding large error and TLP factors. Apart from the reverberation and noise levels, the degree of tracking difficulty is typically determined by the specific target trajectory (sharp turns), the frequency content of the source signal (male vs. female), and the type of acoustic source utilized (human vs. loudspeaker). The results presented here however demonstrate the superiority of PF-VAD compared to a similar PF implementation that does not integrate VAD data. Thus, the VAD-based algorithm presented in this work is more efficient at avoiding track losses during significant speech inactivity periods, and is therefore better suited for practical ASLT implementations.

7.4 Real-Time Audio Tracking

The real-time implementation was evaluated using the same scenario as described in the above chapter, with $T_{60} = 0.27$ s and SNR of approximately 20 dB. The software program saves the audio data along with the estimated source position to disk during execution. A typical tracking result obtained with the above setup is depicted in Figure 9, along with the sound picked up by one of the microphones. The plot shows that a successful and accurate tracking is achieved during periods of speech activity. A typical example of temporary track loss resulting from a nonspeech period can be observed in Figure 9 towards the end of the simulation, where the estimates slightly deviate from the true source trajectory. The

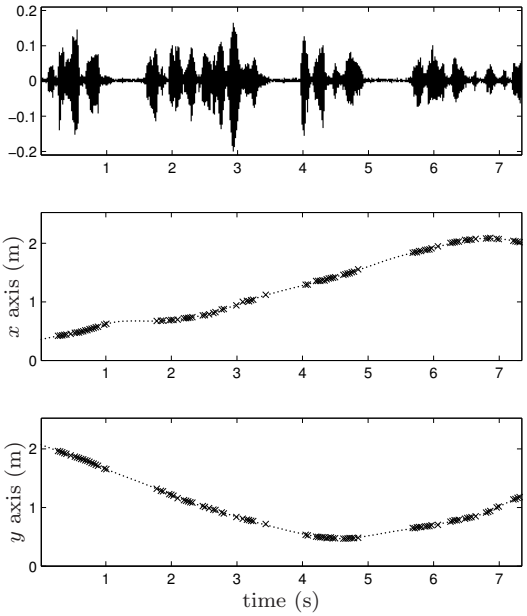

Figure 8. Ground-truth source position from microphone array data. *Top plot*: example of recorded sensor signal. *Bottom plots*: high-accuracy SBF localisation output (×) and polynomial trajectory interpolation (dotted line) in x and y dimensions.

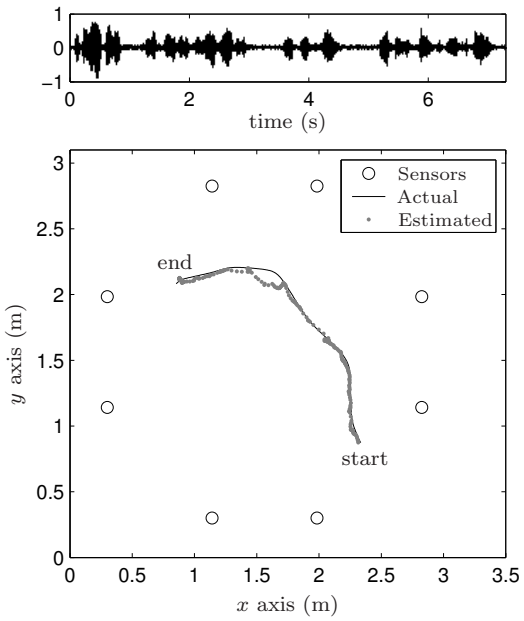

Figure 9. Real-time tracking of human speaker walking in noisy room. The top plot shows an example of signal recorded by one of the microphones.

considered algorithm is however able to successfully resume tracking when the speaker becomes active again.

8 Conclusion and Future Work

This work is concerned with the problem of tracking a human speaker in reverberant and noisy environments by means of an array of acoustic sensors. We derived a PF-based method that integrates VAD measurements at a low level in the statistical algorithm framework. Provided the dynamics of the consid-

		$T_{60} \approx 0.27\,\mathrm{s}$		$T_{60} \approx 0.34\,\mathrm{s}$	
		male	female	male	female
AWGN 5 dB	error $\bar{\varepsilon}$	0.488	0.558	0.860	0.869
	TLP $\bar{\xi}$	73.8	93.5	100.0	100.0
AWGN 10 dB	error $\bar{\varepsilon}$	0.071	0.114	0.280	0.724
	TLP $\bar{\xi}$	1.9	23.0	66.3	100.0
AWGN 15 dB	error $\bar{\varepsilon}$	0.047	0.065	0.143	0.358
	TLP $\bar{\xi}$	1.9	14.5	35.6	91.5
babble 5 dB	error $\bar{\varepsilon}$	0.049	0.099	0.178	0.374
	TLP $\bar{\xi}$	3.8	24.5	39.4	93.5
babble 10 dB	error $\bar{\varepsilon}$	0.044	0.072	0.106	0.332
	TLP $\bar{\xi}$	1.9	17.5	20.6	92.0
babble 15 dB	error $\bar{\varepsilon}$	0.047	0.062	0.092	0.329
	TLP $\bar{\xi}$	4.4	12.0	17.5	92.0

Table 3. Tracking performance results for PF-VAD algorithm: average estimation error $\bar{\varepsilon}$ (m) and track loss percentage $\bar{\xi}$ (%).

		$T_{60} \approx 0.27\,\mathrm{s}$		$T_{60} \approx 0.34\,\mathrm{s}$	
		male	female	male	female
AWGN 5 dB	error $\bar{\varepsilon}$	0.646	0.466	0.765	0.831
	TLP $\bar{\xi}$	85.6	90.0	97.5	95.5
AWGN 10 dB	error $\bar{\varepsilon}$	0.560	0.441	0.726	0.796
	TLP $\bar{\xi}$	78.1	83.0	92.5	95.0
AWGN 15 dB	error $\bar{\varepsilon}$	0.380	0.349	0.662	0.785
	TLP $\bar{\xi}$	66.9	71.5	86.9	93.0
babble 5 dB	error $\bar{\varepsilon}$	0.618	0.553	0.902	0.662
	TLP $\bar{\xi}$	83.1	90.5	98.1	96.0
babble 10 dB	error $\bar{\varepsilon}$	0.455	0.450	0.787	0.650
	TLP $\bar{\xi}$	75.0	87.0	92.5	96.5
babble 15 dB	error $\bar{\varepsilon}$	0.300	0.424	0.764	0.633
	TLP $\bar{\xi}$	62.5	81.0	93.1	96.5

Table 4. Tracking performance results for SBF-PL algorithm: average estimation error $\bar{\varepsilon}$ (m) and track loss percentage $\bar{\xi}$ (%).

ered acoustic source are properly modeled, the proposed PF-VAD method greatly reduces the likelihood of a complete track loss during long silence gaps in the speech signal. The proposed algorithm hence provides an improved tracking performance for real-world implementations compared to previously derived PF methods, this result is further . As a further result of the proposed implementation, the standard deviation of the particle set can now be used as a reliable indication of the filter's own estimation accuracy. The obvious limitation inherent to the current developments is that only one single speaker can be tracked at a time. This work will however serve as a basis for further research on the problem of multiple speaker tracking using the principle of microphone array beamforming.

The results from the real-time implementation show that the algorithm is suitable for real-world implementations. The computational complexity is low enough for a real-time processing on low-power embedded systems using currently existing hardware, and the performance is good enough to successfully track a moving speaker in an acoustically adverse environment.

Acknowledgement

The authors would like to thank Alan Davis for the help provided in regards to the VAD scheme used in this report. This work was supported by National ICT Australia (NICTA) and the Australian Research Council (ARC) under grant no. DP0451111. NICTA is funded by the Australian Government's Department of Communications, Information Technology and the Arts, the Australian Research Council through Backing Australia's Ability, and the ICT Centre of Excellence programs.

References

[1] J. Allen and D. Berkeley. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.

[2] B. Anderson and J. Moore. *Optimal filtering*. Dover Publications, New York, 2005.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[4] D. Bechler, M. Grimm, and K. Kroschel. Speaker tracking with a microphone array using Kalman filtering. *Advances in Radio Science*, 1:113–117, 2003.

[5] J. Chen, L. Shue, and W. Ser. A new approach for speaker tracking in reverberant environment. *Signal Processing*, 82(7):1023–1028, July 2002.

[6] J. Chen, K. Yao, and R. Hudson. Acoustic source localization and beamforming: theory and practice. *EURASIP Journal on Applied Signal Processing*, (4):359–370, May 2003.

[7] A. Davis, S. Nordholm, and R. Togneri. Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold. *IEEE Transactions on Speech and Audio Processing*, 14(2):412–424, March 2006.

[8] S. Doclo and M. Moonen. Robust adaptive time delay estimation for speaker localization in noisy and reverberant acoustic environments. *EURASIP Journal on Applied Signal Processing*, 11:1110–1124, 2003.

[9] T. Dvorkind and S. Gannot. Speaker localization exploiting spatial-temporal information. In *Proceedings of the International Workshop on Acoustic Echo and Noise Control*, pages 295–298, Kyoto, Japan, September 2003.

[10] S. Gannot and T. Dvorkind. Microphone array speaker localizers using spatial-temporal information. *EURASIP Journal on Applied Signal Processing*, 2006, 2006. Article ID 59625, 17 pages.

[11] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2):107–113, April 1993.

[12] S. Haykin. *Communication Systems*. Wiley, New York, 3rd edition, 1994.

[13] Y. Huang, J. Benesty, and G. Elko. Passive acoustic source localization for video camera steering. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 909–912, Istanbul, Turkey, June 2000.

[14] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, August 1976.

[15] M. Krindis, G. Stamou, H. Teutsch, S. Spors, N. Nikolaidis, R. Rabenstein, and I. Pitas. An audio-visual database for evaluating person tracking algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 237–240, Philadelphia, USA, March 2005.

[16] E. Lehmann, D. Ward, and R. Williamson. Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 177–180, Hong Kong, China, April 2003.

[17] I. Potamitis, H. Chen, and G. Tremoulis. Tracking of multiple moving speakers with multiple microphone arrays. *IEEE Transactions on Speech and Audio Processing*, 12(5):520–529, 2004.

[18] X. Sheng and Y. Hu. Sequential acoustic energy based source localization using particle filter in a distributed sensor network. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 972–975, Montreal, Canada, May 2004.

[19] S. Spors, R. Rabenstein, and N. Strobel. A multi-sensor object localization system. In *Workshop on Vision, Modeling and Visualization*, volume 5, pages 19–26, Stuttgart, Germany, November 2001.

[20] J. Vermaak and A. Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3021–3024, Salt Lake City, USA, May 2001.

[21] D. Ward, E. Lehmann, and R. 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.

[22] R. Waterhouse. Statistical properties of reverberant sound fields. *Journal of the Acoustical Society of America*, 43(6):1436–1444, 1968.