EXPERIMENTAL COMPARISON OF PARTICLE FILTERING ALGORITHMS FOR ACOUSTIC SOURCE LOCALIZATION IN A REVERBERANT ROOM

Eric A. Lehmann[1]*, Darren B. Ward*[2]*, and Robert C. Williamson*[1]

[1]Department of Telecommunications Engineering, RSISE
The Australian National University
Canberra ACT 0200, Australia

[2]Department of Electrical & Electronic Engineering
Imperial College London
United Kingdom

ABSTRACT

Traditional acoustic source localization techniques attempt to determine the current location of an acoustic source from data obtained at an array of sensors during the current time only. Recently, state-space methods have been proposed that use particle filters to perform recursive estimation of the current source location using all previous data. In this paper we present an overview of these particle filter algorithms, and formulate performance measures for determining their ability to track a moving source. We present results of experiments using reverberant data recorded in a real room, and show that steered beamforming methods have improved performance over GCC-based approaches.

1. INTRODUCTION

The problem of locating and tracking an acoustic source in a reverberant room occurs in several applications, including automatic camera steering for video-conferencing, discriminating between individual talkers in multisource environments, and providing steering information for microphone arrays [1].

Traditional approaches to this problem collect data from several microphones and use a frame of data obtained at the current time to estimate the current source location. These traditional approaches can be divided into two categories: (i) time-delay estimation (TDE) methods such as the well-known generalized cross-correlation (GCC) function [2], which estimate location based on the time delay of arrival of signals at the receivers; and (ii) direct methods such as steered beamforming. Each method transforms the received frame of data into a function that exhibits a peak in the location corresponding to the source. We will refer to this function as the *localization function*. The practical disadvantage of these traditional approaches is that reverberation causes spurious peaks to occur in the localization function. These spurious peaks may have greater amplitude than the peak due to the true source, so that simply choosing the maximum peak to estimate the source location may not give accurate results.

A promising approach that overcomes the drawback of traditional methods is to use a state-space approach based on particle filtering, as recently described in [3, 4]. The key to these new techniques is that the peak due to the true source follows a dynamical model from frame to frame, whereas there is no temporal consistency to the spurious peaks. Using a sequential Monte Carlo approach, particle filters are used to recursively estimate the probability density of the unknown source location conditioned on all received data up to and including the current frame. Simulation results based on the image method [5] were used in [3, 4] to demonstrate the performance of these approaches. While the image method is useful for initial testing of algorithms, it is only through experiments using data recorded in real rooms that the true performance of source localization algorithms can be appreciated.

With that motivation in mind, in this paper we compare the performance of the particle filter algorithms in [3, 4] using experiments performed in a reverberant room. In the following section we formulate the localization problem, and present a general framework to describe the particle filter approach to this problem. We then describe several performance measures that are used to analyze the tracking ability of these algorithms, and present the performance of these algorithms using experimental data.

2. SOURCE LOCALIZATION

2.1. Problem Formulation

Consider a collection of M sensors positioned in arbitrary positions and located in a multipath environment. Assuming a single source, the discrete-time signal received at the mth sensor (where $m = 1, \ldots, M$) is:

$$x_m(k) = h_m(k) \star s(k) + n_m(k), \tag{1}$$

where $h_m(k)$ is the impulse response from the source to the mth sensor, $s(k)$ is the source signal, $n_m(k)$ is additive noise (assumed to be uncorrelated with the source signal and from sensor to sensor), and $\star$ denotes convolution. The impulse response from the source to any sensor can be separated into direct path and multipath terms, giving

$$x_m(k) = \frac{1}{4\pi\|\ell_s - \ell_m\|}\, s(k-\tau_m) + s(k) \star g_m(k) + n_m(k) \tag{2}$$

where $\ell_s = [\mathcal{X}_s, \mathcal{Y}_s, \mathcal{Z}_s]^T$ is the source location in Cartesian coordinates, ℓ_m is the sensor location, $g_m(k)$ is the component of the impulse response between the source and the mth sensor due to reverberation only, and $\|\cdot\|$ denotes the vector 2-norm. The delay from the source to the mth sensor is $\tau_m = c^{-1}\|\ell_s - \ell_m\|$, where c is the speed of wave propagation.

Assume that the data at each sensor are collected over a frame of L samples, and denote the data at the mth sensor for frame t as $\mathbf{x}_m(t)$. Stack the sensor frames to form the $L \times M$ matrix $\mathbf{X}_t = [\mathbf{x}_1(t)^T, \cdots, \mathbf{x}_M(t)^T]$ which represents the data received at the array during time frame t. We will refer to $\mathbf{X}_t$ as the *raw data*. The problem is to estimate the current location of the source from the raw data.

2.2. State-Space Approach

The source localization problem can be formulated in state-space form as follows. Let the source state at time t be

$$\boldsymbol{\alpha}_t = \left[\mathcal{X}_s, \mathcal{Y}_s, \mathcal{Z}_s, \dot{\mathcal{X}}_s, \dot{\mathcal{Y}}_s, \dot{\mathcal{Z}}_s\right]^T, \tag{3}$$

where $[\mathcal{X}_s, \mathcal{Y}_s, \mathcal{Z}_s]^T$ is the true source location in Cartesian coordinates, and $[\dot{\mathcal{X}}_s, \dot{\mathcal{Y}}_s, \dot{\mathcal{Z}}_s]^T$ is the source velocity. For a given state $\boldsymbol{\alpha}$, we will denote the location vector of the state as $\ell_{\boldsymbol{\alpha}}$.

At time t, assume that a measurement $\mathbf{y}_t$ of the unobserved state becomes available. This measurement is described by the state-space equation

$$\mathbf{y}_t = S(\boldsymbol{\alpha}_t, \mathbf{n}_1(t)), \tag{4}$$

where $S(\cdot)$ is an unknown, not necessarily linear, function of the state $\boldsymbol{\alpha}_t$ and a noise term $\mathbf{n}_1(t)$. Assume also that the state is a Markov process, which can be modelled by the state transition relation

$$\boldsymbol{\alpha}_t = T(\boldsymbol{\alpha}_{t-1}, \mathbf{n}_2(t)), \tag{5}$$

where $T(\cdot)$ is a known, not necessarily linear, function of the previous state and a noise term $\mathbf{n}_2(t)$.

Physically, the measurement $\mathbf{y}_t$ is obtained through some transformation of the raw data:

$$\mathbf{y}_t(\theta) = \mathbf{f}(\theta, \mathbf{X}_t), \tag{6}$$

where we refer to θ as the *localization parameter* and $\mathbf{f}(\cdot)$ as the *localization function*. This model can describe measurements obtained either through a TDE localization method or a direct localization method. One should note that (4) is a state-space equation that describes the measurements as a function of the unobserved state, whereas (6) describes how the measurements are physically obtained from the raw data.

Let $\mathbf{y}_{1:t} = [\mathbf{y}_1, \ldots, \mathbf{y}_t]$ denote the concatenation of all measurements up to time t. The aim is then to recursively estimate the conditional probability density $p(\boldsymbol{\alpha}_t|\mathbf{y}_{1:t})$; the source location can be estimated as the mean or mode of this density function. Unfortunately, in practice this posterior filtering density is unavailable. However, assuming that the posterior density at time $t-1$ is available, then the posterior at time t can be found through prediction and updating as [6]

$$p(\boldsymbol{\alpha}_t|\mathbf{y}_{1:t-1}) = \int p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})\, p(\boldsymbol{\alpha}_{t-1}|\mathbf{y}_{1:t-1})\, \mathrm{d}\boldsymbol{\alpha}_{t-1} \tag{7a}$$

$$p(\boldsymbol{\alpha}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\boldsymbol{\alpha}_t)\, p(\boldsymbol{\alpha}_t|\mathbf{y}_{1:t-1}), \tag{7b}$$

where $p(\boldsymbol{\alpha}_t|\mathbf{y}_{1:t-1})$ is the prior, $p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})$ is the state transition density, and $p(\mathbf{y}_t|\boldsymbol{\alpha}_t)$ is the likelihood (or measurement density).

3. LOCALIZATION USING PARTICLE FILTERING

In general no closed-form solution exists for (7a) and (7b), although these recursions can be approximated through Monte Carlo simulation of a set of particles (representing samples of the source state) having associated discrete probability masses. The generic particle filtering algorithm is described in [7].

Two recently proposed source localization algorithms have been developed using particle filtering and the state-space approach. In [3] a TDE localization method based on the GCC was used, whereas in [4] a direct localization method based on steered beamforming was used. Both of these algorithms can be described by the general particle filtering algorithm shown in Fig. 1. This is a standard particle filtering algorithm, and only steps 2)–4) are specific to the source localization problem. There are three algorithmic choices to be made: (i) what model to use for the source dynamics in Step 2); (ii) what localization function to use in Step 3); and (iii) how to calculate the likelihood function in Step 4).

Form an initial set of particles $\{\boldsymbol{\alpha}_0^{(i)}, i = 1:N\}$ and give them uniform weights $w_0^{(i)} = 1/N, i = 1:N$. For each new data frame:

1. Resample the particles from the previous frame $\{\boldsymbol{\alpha}_{t-1}^{(i)}\}$ according to their weights $\{w_{t-1}^{(i)}\}$ to form the resampled set of particles $\{\tilde{\boldsymbol{\alpha}}_{t-1}^{(i)}, i = 1:N\}$
2. Predict the new set of particles $\{\boldsymbol{\alpha}_t^{(i)}\}$ by propagating the resampled set $\{\tilde{\boldsymbol{\alpha}}_{t-1}^{(i)}\}$ through the source dynamical model
3. Transform the raw data into localization measurements using the localization function: $\mathbf{y}_t(\theta) = \mathbf{f}(\theta, \mathbf{X}_t)$
4. Form the likelihood function: $p(\mathbf{y}_t|\boldsymbol{\alpha}) = F(\mathbf{y}_t, \boldsymbol{\alpha})$
5. Weight the new particles according to the likelihood function: $w_t^{(i)} = p(\mathbf{y}_t|\boldsymbol{\alpha}_t^{(i)})$ and normalize so that $\sum_i w_t^{(i)} = 1$
6. Compute the current source location estimate $\hat{\ell}_s$ as the weighted sum of the particle locations $E\{\ell_t\} = \sum_{i=1}^{N} w_t^{(i)} \ell_{\boldsymbol{\alpha}}^{(i)}$
7. Store the particles and their respective weights $\{\boldsymbol{\alpha}_t^{(i)}, w_t^{(i)}, i = 1:N\}$

Fig. 1. Particle filtering algorithm for source localization.

Source Dynamics: several dynamical models can be used to model the time-varying location of a person moving in a room, e.g., [8]. One that is reasonably simple but has been shown to work well in practice is the Langevin model used in [3]. In this model the source motion in each of the Cartesian coordinates is assumed to be an independent second-order process. This model was also used in [4], and we use it in the experiments reported in the sequel.

The Localization Function: there are two classes of possible localization function, corresponding to the two methods used for conventional source localization: TDE and direct methods. In [3], a GCC localization function (TDE method) was used, whereas [4] used a steered beamformer localization function (direct method).

The Likelihood Function (LF): for a given state $\boldsymbol{\alpha}$, the LF measures the likelihood of receiving the data $\mathbf{y}_t$. In [3], a *Gaussian LF* was used. If K potential locations have been obtained from the localization function, then the Gaussian LF is formed by assuming that either one of these potential locations is due to the true source location corrupted by additive Gaussian noise, or none of the potential locations is due to the true source location. A design parameter q_0 reflects the prior probability that none of the potential locations is due to the source location. In [4] a *pseudo-LF* was used, wherein the localization function was used directly as the basis of the likelihood. Analogous with the use of q_0 in the Gaussian LF, a lower bound can be included in the pseudo-LF to allow for the case where no peak in the localization function corresponds to the true source location.

4. PERFORMANCE MEASURES

To provide a reproducible and algorithm-independent assessment of the tracking ability of a particle filter applied to the problem of acoustic source localization, we now formulate three specific performance parameters.

Mean Square Error (MSE): for each frame of raw data $\mathbf{X}_t$ received from the sensors, the particle filter delivers an estimate of the current source location as $\hat{\ell}_{\mathrm{s}} = E\{\ell_t\}$. The square error ε_t for time frame t is computed as $\varepsilon_t = \|\ell_{\mathrm{s}} - \hat{\ell}_{\mathrm{s}}\|^2$. The MSE value then corresponds to the variable ε_t averaged over the total number of frames in the processing of the audio signal.

This parameter gives an indication of how much the source location estimate deviates from the true source position. A high MSE value hence always reflects an inaccurate tracking ability.

Mean Standard Deviation (MSTD): for each time frame t, the standard deviation ς_t of the particle set is defined as $\varsigma_t = [\sum_{i=1}^{N} w_t^{(i)} \|\ell_{\boldsymbol{\alpha}}^{(i)} - \hat{\ell}_{\mathrm{s}}\|^2]^{\frac{1}{2}}$. The MSTD value is the variable ς_t averaged over the total number of frames in the audio sample.

This parameter is an accuracy measure of the estimated source position delivered by the particle filter. A large ς_t value means that the position estimate $\hat{\ell}_{\mathrm{s}}$ results from a widely spread particle set, indicating a low level of estimation certainty.

Frame Convergence Ratio (FCR): we first define the term *convergence* as follows. For time frame t, the particle filter is said to be converging toward the true source position ℓ_{s} if this latter lies within one standard deviation ς_t from the estimated source location $\hat{\ell}_{\mathrm{s}}$. In other words, the particle filter is convergent if the following inequality holds: $\|\ell_{\mathrm{s}} - \hat{\ell}_{\mathrm{s}}\| \leqslant \varsigma_t + \delta$, where the parameter δ accounts for the inaccuracy of the source position measurements during the audio recordings. The parameter FCR is defined as the percentage of frames for which the particle filter has been found to converge, over the entire audio sample length.

5. EXPERIMENTS

A series of experiments using real audio data has been performed to determine the performance of two particle filtering methods, namely the algorithms based on: (i) GCC localization function used as Gaussian likelihood (GCC-GL, similar to the algorithm used in [3]); and (ii) steered beamformer localization function used as pseudo-likelihood (SBF-PL, essentially the algorithm of [4]). These tests allow for a comparative assessment of the tracking ability of each method when used in reverberant and noisy conditions.

5.1. Experimental setup

Hardware setup: the recording environment was a typical office room measuring roughly $2.9\mathrm{m} \times 3.8\mathrm{m} \times 2.7\mathrm{m}$, with various encased or protruding spaces (windows, door, furniture, etc.). The frequency-averaged reverberation time RT_{60} was experimentally measured to be 0.39s. The recording setup made use of a total of 8 microphones positioned at a constant height and organized as one pair on each wall of the room.

The moving sound source was simulated in the room as a loudspeaker in upright position and following a predefined path at a constant height of 1.464m (distance from the floor to the center of the speaker cone). For practical reasons, the source trajectory was always a straight line, showing a variety of lengths and orientations. A small source of error (estimated to be less than 10cm) may have been introduced when monitoring the position of the speaker for the duration of the recording. The measurement inaccuracy parameter δ was therefore set to 0.1m.

The audio samples used as source signals were speech utterances by male speakers taken from the TIMIT database, with a sample length varying from 3.6s to 7.5s. The sensor signals were all sampled at 8kHz and band-pass filtered between 300 and 3000Hz prior to particle filter processing.

Software setup: to ensure a fair comparison of the two methods, the parameters of each algorithm were independently tuned using a reference audio sample to achieve the best particle filter performance. This process was done empirically by simulating each algorithm a number of times with varying parameters until a satisfactory performance was achieved.

The particle set for each algorithm was initialized by placing each particle at the start location of the sound source in the room. This way, the unpredictable effects of a uniform initial particle distribution were reduced to a negligible level.[†] In both algorithms, the incoming sensor signals were split into frames of $L = 512$ samples (frame length of 64ms) and the processing was carried out using a frame overlapping factor of 0.5.

5.2. Analysis of experimental results

To illustrate some of the experimental results, we present some typical plots obtained from algorithm SBF-PL. The first plot in Fig. 2 shows an example of the function used as pseudo-likelihood plotted for one signal frame over the entire 2D state-space (note that for SBF-PL, this likelihood function is to be evaluated *only* at the particles' positions). This plot shows clearly the multi-hypothesis character of the observation: the true source peak is located at the $(\mathcal{X}, \mathcal{Y})$-coordinate position $(0.75, 2.3)$, other peaks are clutter measurements due to reverberation.

The other two plots in Fig. 2 present the tracking result in the $\mathcal{X}$ and $\mathcal{Y}$ coordinates for a 3.8s run of algorithm SBF-PL. It demonstrates the ability of this method to accurately track the sound source across the room despite the relatively high level of reverberation. This kind of result typically yields tracking quality values of $\mathrm{MSE} = 0.013\mathrm{m}^2$, $\mathrm{MSTD} = 0.094\mathrm{m}$ and $\mathrm{FCR} = 0.95$.

5.3. Comparative results

The results presented here have been obtained in the following manner. Each method under test was run 100 times with each one of 6 different real audio samples, implying a variety of source signals and trajectories. Since a different level of performance is usually achieved for different source signals and paths, the results obtained for each of the audio samples are given separately. Table 1 contains the values obtained for the performance assessment parameters averaged over the 100 real audio simulations.

5.4. Discussion

In Table 1, the differences in the overall performance results from one sample to the other reflect a variable degree of tracking difficulty for the algorithms, resulting typically from the quality of the audio signals and the specific trajectory of the sound source.

It can be seen that for a couple of samples, the performances of both algorithms are similar. However, as soon as the quality of the experimental conditions diminishes, the tracking performance of GCC-GL rapidly deteriorates whereas SBF-PL still manages to function with a reasonable level of tracking accuracy, as indicated

[†] We are only interested in the performance of the algorithms in *tracking* mode. Initialization considerations are of course important for a functional system, but we do not examine this in the present paper.

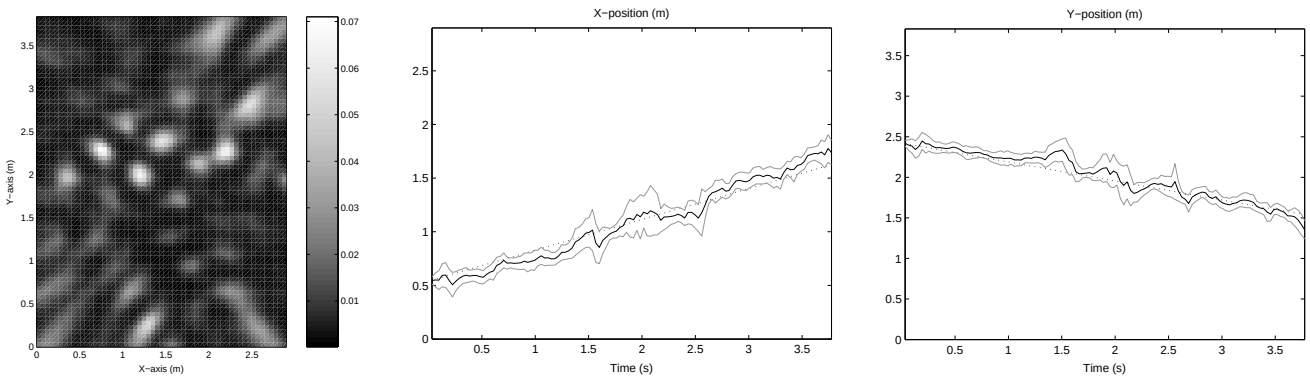

Fig. 2. Example results from algorithm SBF-PL using real audio data. *Left plot*: beamformer output function used as pseudo-likelihood, for one signal frame. *Middle and right plots*: tracking results with true source trajectory (dotted line), estimated source trajectory (solid line), and lines representing $\pm$ one standard deviation of the particle set from its weighted mean $\hat{\ell}_s$ (grey lines).

	SBFPL	GCCGL
MSE	0.082	0.026
MSTD	0.212	0.098
FCR	86.1	80.3
MSE	0.022	0.057
MSTD	0.182	0.109
FCR	97.3	67.5
MSE	0.021	0.403
MSTD	0.193	0.116
FCR	97.7	33.9

	SBFPL	GCCGL
MSE	0.170	0.168
MSTD	0.219	0.111
FCR	74.0	49.4
MSE	0.024	0.282
MSTD	0.174	0.116
FCR	97.8	34.1
MSE	0.171	0.848
MSTD	0.247	0.116
FCR	79.2	19.8

Table 1. Comparative results. Each of the 6 main rows shows the average performance measures (MSE in m^2, MSTD in m, FCR in %) for a different sample of real audio data.

by the MSE and FCR parameters.[‡] This behavior is confirmed when investigating the 2D likelihood function frame-after-frame for each algorithm. The GCC-based likelihood shows a distinctively lower level of robustness against spurious peaks. Also, the presence of the true source peak in this likelihood function is usually more emphasized with the SBF method.

The MSTD values shown in Table 1 are more or less constant for both algorithms, which reflects the fact that this value is mainly a result of the specific parameter setting chosen for each of them.

6. CONCLUSIONS

Carrying out source localization in the practical environment of a moderately reverberant office room is a complicated task. Even low levels of reverberation or background noise can rapidly become detrimental to classical TDE-based methods. Under such adverse conditions, incorporating these observations in the framework of a sequential Monte Carlo method proves to be a substantial advantage. Using audio data samples recorded in a real room, we have furthermore demonstrated that algorithms based on a steered beamforming principle show a higher degree of robustness against reverberation and background noise. An added attraction of the SBF-PL method is that, on the basis of some work not reported in this paper, it seems quite feasible to implement this algorithm in real-time on a standard personal computer.

[‡]The MSTD value has an indirect influence on the parameter FCR: an increased MSTD value may be partly responsible for a large FCR value.

7. ACKNOWLEDGEMENTS

We wish to thank: Kris Modrak for the measurements and recordings used in Sec. 5; Arnaud Doucet and Nando de Freitas for providing the resampling code used in the particle filter algorithm. This work was supported by the Australian Research Council.

8. REFERENCES

[1] M.S. Brandstein and D.B. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, Berlin, 2001.

[2] C.H. Knapp and G.C. Carter, "The generalized correlation method of estimation of time delay," *IEEE Trans. ASSP*, vol. ASSP-24, no. 4, pp. 320–327, 1976.

[3] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environments," in *Proc. IEEE ICASSP*, Salt Lake City, UT, USA, May 2001.

[4] D.B. Ward and R.C. Williamson, "Particle filter beamforming for acoustic source localization in a reverberant environment," in *Proc. IEEE ICASSP*, Orlando, FL, USA, May 2002.

[5] J.B. Allen and D.A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[6] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Commun., Radar & Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[7] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[8] M. Isard and A. Blake, "Condensation–Conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.