A REAL-TIME TALKER LOCALIZATION IMPLEMENTATION USING MULTI-PHAT AND PARTICLE FILTER

Antti Löytynoja, Pasi Pertilä

Department of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, 33720, Tampere, Finland
phone: + (358) 3 3115 3788, fax: + (358) 3 3115 3857, email: antti.loytynoja@tut.fi, pasi.pertila@tut.fi

ABSTRACT

The estimation of a talker location in noisy and reverberant conditions using microphone arrays requires complex algorithms. However, the rapid increase in computational performance during the last two decades has opened the way for practical real-time talker localization applications. This paper presents an implementation of such system based on steered response power with phase transform (SRP-PHAT) combined with a particle filter (PF). The system utilizes the simplicity of SRP-PHAT and robustness of PF. The system is implemented using Pure Data software running on a standard laptop PC. Using realistic test data we show that accurate real-time talker location estimates can be produced even with a relatively low particle density.

1. INTRODUCTION

Acoustic source localization (ASL) using spatially separated microphone arrays has been an active research topic for at least two decades. The aim of ASL is to obtain the location of one or more acoustic sources using the audio signals acquired by the microphones. The location information can be used, e.g, in smart video conference systems to steer the video camera towards the speaker. In reverberant room environments the system must be able to cope with secondary signals reflected from the surfaces.

In [3] few popular methods for solving the ASL problem are presented. Lately the combination of steered beamforming (SBF) and time delay estimation (TDE) based methods have become popular due to their relatively low computational complexity. Among these methods the steered response power (SRP) with phase transform (PHAT) has been found to be robust in the presence of room reverberation [3, 11].

In the recent years, some real-time ASL implementations have been developed. Do et al. [4] based their system on SRP-PHAT, while Lehmann et al. [9, 5] implemented a system based on SBF and particle filter (PF), fused with a voice activity detector (VAD). Both of these implementations were found suitable for practical applications in terms of computational performance. The former method relies on finding the global maximum of the SRP using stochastic region contraction (SRC). However, there is no assurance that the location of the global maximum is actually the source location, as the maximum can be caused by a brief dominant noise peak due to reverberation. This problem can be overcome by using Bayesian recursive estimation, such as PF, to make use of the complete measurement history to provide the current source location estimate.

This paper discusses a real-time ASL implementation based on MULTI-PHAT and PF as described in [7, 10]. In the proposed implementation, MULTI-PHAT method derived from SRP-PHAT is used to generate a combined spatial likelihood function (SLF) that can be used as source evidence for a PF algorithm. The likelihood data is generated by estimating the temporal difference of the signals sensed by a microphone pair using PHAT-weighted generalized cross correlation (GCC-PHAT). The resulting data from each microphone pair is combined by multiplication instead of summation used in traditional SRP, hence the name MULTI-PHAT. It has been shown that combining the likelihood data by multiplication decreases the variance of the spatial likelihood distribution resulting in a significant reduction of the root mean squared error (RMSE) [10].

This paper is organized as follows. The next Section presents the TDE-function and a method for generating a combined SLF using multiplication. In the third Section the ASL problem is presented as a state-space filtering problem and the Bayesian solution to this filtering problem is presented in a form of a PF algorithm. In the fourth Section the implementation of the system is described and the complexity of the implementation is stated. In the fifth Section the setup and test data are described and the localization results using real data are presented. The last Section concludes the paper by summarizing the results.

2. LIKELIHOOD FUNCTION

Consider a room with an active sound source located at $\mathbf{r}$ and a pair of spatially separated microphones i and j. The signal emitted from the sound source is sensed by the microphone i after a propagation delay espressed as:

$$\tau_{i,\mathbf{r}} = |\mathbf{r} - \mathbf{m}_i| \cdot c^{-1},$$

where c is the speed of sound, $|\cdot|$ is the l^2 norm, and $\mathbf{r}$ and $\mathbf{m}_i$ are the Cartesian coordinate vectors of the sound source and the microphone i, respectively. Due to the different sound propagation paths to microphones i and j, a time difference of arrival (TDOA) can be calculated. Geometrically, assuming spherical sound radiation pattern, the TDOA between a microphone pair $b = \{i, j\}$ can be expressed in discrete samples as:

$$\tau_{b,\mathbf{r}} = Q((|\mathbf{r} - \mathbf{m}_i| - |\mathbf{r} - \mathbf{m}_j|) \cdot f_s \cdot c^{-1}), \quad (1)$$

where f_s is the sampling frequency and $Q(\cdot)$ is the quantization operator. From the measured signals, the TDOA between a microphone pair b can be estimated using a TDE-function $R_b(\cdot)$. A popular choice for a TDE-function is the so-called generalized cross correlation (GCC) function [6]:

$$R_b^{GCC}(\tau_b) = F^{-1}\{W_b(k)X_i(k)\overline{X}_j(k)\}, \quad (2)$$

Algorithm 1: SIR particle filter

Input: $\{\mathbf{S}^i_{\kappa-1}, w^i_{\kappa-1}\}^P_{i=1}, \mathbf{Z}_\kappa$
Output: $\{\mathbf{S}^i_\kappa, w^i_\kappa\}^P_{i=1}$
for $i = 1 : P$ **do**
  Take a particle:
  $\mathbf{S}^i_\kappa \Leftarrow q(\mathbf{S}_\kappa|\mathbf{S}^i_{\kappa-1}, \mathbf{Z}_\kappa)$
  Move the particle according to movement model:
  $\mathbf{S}^i_\kappa = g(\mathbf{S}_{\kappa-1}, \mathbf{a}_{\kappa-1})$
  Calculate a weight proportional to the likelihood:
  $w^i_\kappa \Leftarrow p(\mathbf{Z}_\kappa|\mathbf{S}^i_\kappa)$
end
Calculate the sum of weights:
$W \Leftarrow \sum_{i=1}^{P} w^i_\kappa$
for $i = 1 : P$ **do**
  Normalize the weights: $w^i_\kappa \Leftarrow W^{-1} w^i_\kappa$
end
Resample using algorithm 2:
$\{\mathbf{S}^i_\kappa, w^i_\kappa\}^P_{i=1} \Leftarrow \text{RESAMPLE}(\{\mathbf{S}^i_\kappa, w^i_\kappa\}^P_{i=1})$

Algorithm 2: Resampling algorithm

Input: $\{\mathbf{S}^i_{\kappa-1}, w^i_{\kappa-1}\}^P_{i=1}$
Output: $\{\mathbf{S}^{j*}_\kappa, w^i_\kappa, i^j\}^P_{i=1}$
Initialize the cumulative distribution function (CDF):
$c_1 \Leftarrow 0$
for $i = 2 : P$ **do**
  Generate CDF: $c_i \Leftarrow c_{i-1} + w^i_\kappa$
end
Start at the beginning of the CDF: $i \Leftarrow 1$
Draw a random starting point: $u_1 \sim \mathbb{U}[0, P^{-1}]$
for $j = 1 : P$ **do**
  Move along the CDF: $u_j \Leftarrow u_1 + P^{-1}$
  while $u_j > c_i$ **do**
    $*i \Leftarrow i + 1$
  end
  Resample by reassigning locations and weights:
  $\mathbf{S}^{j*}_\kappa \Leftarrow \mathbf{S}^i_\kappa$
  $w^i_\kappa \Leftarrow P^{-1}$
end

where $\bar{X}_j(k)$ is the complex conjugate of the Fourier transformed microphone signal $x_j(n)$, k is the discrete frequency, $F^{-1}(\cdot)$ denotes the inverse Fourier transform, and $W_b(k)$ is the weighting function. The TDOA between the microphones i and j can be determined by locating the global maximum of the real-valued TDE-function $R^{GCC}_b(\tau_b)$. PHAT weighting have been shown to produce an emphasized peak especially in reverberant rooms and therefore it is selected for this implementation.

Using the equation (1), each source location candidate inside the room of interest can be mapped into a set of microphone pair -specific TDOAs $\tau_{b,\mathbf{r}} = \{\tau_1 \dots \tau_N\}$, where N is the total number of the microphone pairs. With a TDOA value and corresponding TDE-function, a likelihood value can be assigned to a given source location candidate using the equation (2). Inversely, a TDOA value can be mapped into a set of locations. The set of locations corresponding to the $\arg\max R^{GCC}_b(\tau_b)$ traces out a hyperbola with a slightly higher likelihood value than other locations in the spatial domain. By combining several pairwise TDE-functions, a combined spatial likelihood function (SLF) can be constructed. In the SLF the hyperbolae intersect and through combination form a global likelihood maximum at the intersection point. In favourable conditions, the intersection point is the location of the source.

Several different combination methods for SLF have been introduced. Summation is used in [3], multiplication is used in [8, 7] and the determinant is used in [2]. It is shown in [10] that using multiplication instead of summation, the peak in SLF is emphasized, resulting in 45% reduction in the location RMS error.

Consider a set of microphone pairs Ω that contains the microphone pairs inside each array for all arrays, but not the inter-array pairs. The combined likelihood for a given source location candidate $\mathbf{r}$ using multiplication can be expressed as:

$$L(R_\Omega|\mathbf{r}) = \prod_{b\in\Omega} R_b(\tau_{\mathbf{r},b}) \tag{3}$$

Equations (2) and (3) form the basis for generating the SLF used as source evidence for the tracking algorithm.

3. TALKER TRACKING

Traditionally, the global maximum of the most recent SLF is considered as the source location. However, the measurement data is often corrupted by noise and reverberation, and dominant peaks can occur outside the actual source location resulting in false source evidence. In the proposed implementation, a sequential Monte Carlo method called particle filter (PF) is incorporated to provide past location information and to increase robustness against these outliers.

The estimation of the sound source location is essentially a special case of a problem of estimating the state of the system using noisy measurements. In Bayesian framework, the SLF represents the noisy measurement distribution $p(\mathbf{R}_\kappa|\mathbf{S}_\kappa)$, where $\mathbf{R}_\kappa$ is the noisy measurement obtained using GCC-PHAT, and $\mathbf{S}_\kappa$ is the state of the system at time index κ.

The aim is to estimate the current state $\mathbf{S}_\kappa$ using all the measurement data $\mathbf{R}_{1:\kappa}$ available so far. The subindexes indicate that past measurement data is taken into account while estimating the current state.

The solution to this state-space filtering problem is obtained by the two-step principle of prediction and update. Assuming that the posterior distribution $p(\mathbf{S}_{\kappa-1}|\mathbf{R}_{1:\kappa-1})$ is known at time index $\kappa-1$, the prediction of the state at time index κ can be calculated:

$$p(\mathbf{S}_\kappa|\mathbf{R}_{1:\kappa-1}) = \int p(\mathbf{S}_\kappa|\mathbf{S}_{\kappa-1}) p(\mathbf{S}_{\kappa-1}|\mathbf{R}_{1:\kappa-1}) d\mathbf{S}_{\kappa-1} \tag{4}$$

and when the new measurement $\mathbf{R}_\kappa$ becomes available, the predicted prior distribution of the system at the current time instance κ can be updated to posterior distribution using the Bayes' rule:

$$p(\mathbf{S}_\kappa|\mathbf{R}_{1:\kappa}) = \frac{p(\mathbf{R}_\kappa|\mathbf{S}_\kappa) p(\mathbf{S}_\kappa|\mathbf{R}_{1:\kappa-1})}{p(\mathbf{R}_\kappa|\mathbf{R}_{1:\kappa-1})}, \tag{5}$$

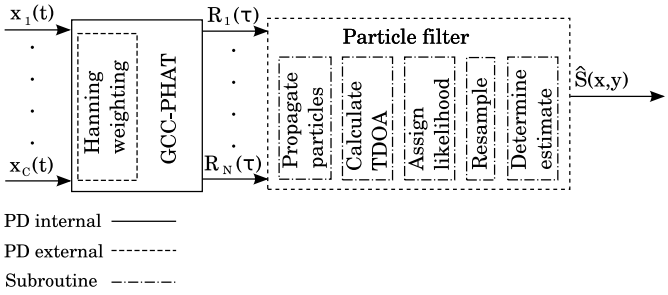

Figure 1: Diagram of the sound source localization system.

where the normalizing constant can be expressed as:

$$p(\mathbf{R}_\kappa|\mathbf{R}_{1:\kappa-1}) = \int p(\mathbf{R}_\kappa|\mathbf{S}_\kappa)p(\mathbf{S}_\kappa|\mathbf{R}_{1:\kappa-1})d\mathbf{S}_\kappa.$$

The recursive evolution of the distribution by prediction and update described in (4) and (5) is repeated at each time instance κ.

The principle of particle filtering is that the distribution is approximated by a set of P weighted particles $\{\mathbf{S}^\rho_{0:\kappa}, w^\rho_\kappa\}^P_{\rho=1}$, where $\{\mathbf{S}^\rho_{0:\kappa}\}^P_{\rho=1}$ is a set of points each with associated weights $\{w^\rho_\kappa\}^P_{\rho=1}$, and $\mathbf{S}_{0:\kappa}$ is the set of all states so far. The true posterior density at time instance κ can be approximated as:

$$p(\mathbf{S}_{0:\kappa}|\mathbf{R}_{1:\kappa}) \approx \sum_{\rho=1}^{P} w^\rho_\kappa \delta(\mathbf{S}_{0:\kappa} - \mathbf{S}^\rho_{0:\kappa}),$$

where $\delta(\cdot)$ is the Dirac delta function. As the number P increases very large, the discrete approximation becomes equivalent to the functional representation of the posterior distribution.

In our implementation, the state consists of a 3D coordinate, and the prediction and update of the sampled distribution is performed by recursively processing the particles according to sampling importance resampling (SIR) [1], presented in Algorithm 1. The algorithm propagates the particles (prediction) according to a motion model, and when a new SLF is constructed, assigns each particle a weight proportional to the likelihood value at the location of the particle. It is assumed that the speakers are seated and only move their heads randomly. Therefore Brownian motion is used to model the movement.

The algorithm also includes a resampling step (update), presented in Algorithm 2, to avoid the so-called degeneracy problem. In this step, particles with low weight are systematically replaced with ones with high weight. This causes the particles to clusterize at locations with high likelihood. In order to avoid the clusterization of every particle, seven percent of the particles are relocated in random locations inside the measurement space. The idea of this step is to expand the spatial range of measurement in the presence of a sound source, and to be able to "sense" the appearance of another sound source at another location.

After resampling, it is only the matter of selecting a point estimate based on the updated particle mass. One logical way is to look where the most particles reside. In our implementation, a 2D point estimate is drawn by calculating the median coordinate of the particles separately across two dimensions:

$$\hat{\mathbf{S}}_\kappa(x,y) = med\{\rho_1 \dots \rho_P\}_d,$$

where $\{\cdot\}_d$ is the set of P particles ordered according to dimension $d = \{x,y\}$.

Table 1: The asymptotic time complexity of different procedures in GCC-PHAT module.

Procedure	Asymptotic time complexity
Hanning weighting	$O(A \times m \times L)$
FFT	$O(A \times m \times L \log_2 L)$
PHAT	$O(A \times M \times L)$
IFFT	$O(A \times M \times L \log_2 L)$

Table 2: The asymptotic time complexity of different procedures in the PF module.

Procedure	Asymptotic time complexity
Normalization	$O(A \times M \times L)$
Particle propagation	$O(P)$
TDOA calculation	$O(A \times M \times P)$
Weight assignment	$O(A \times M \times P)$
Resampling	$O(P^2)$
Median estimation	$O(P^2)$

4. IMPLEMENTATION

The system proposed in this paper is developed and tested on a standard laptop PC with 2.2 GHz Intel Core Duo processor and 2 Gb of DDR2 SDRAM. The theoretical computational efficiency of the processor is 17.6×10^9 floating point operations per second (FLOPS). The operating system on the laptop is Kubuntu Linux with a real-time kernel version 2.6.20-16. For development environment, Pure Data (PD) software [12] was selected. PD is a real-time graphical open source programming environment for creation of computer music and multimedia. The PD distribution includes a vast collection of internal objects, which enable even the most advanced audio signal processing. The modular code base of PD also enables the programming of external custom objects in C programming language.

Our system is implemented using both internal objects of PD and external objects programmed in C. Briefly, the calculation of GCC-PHAT is performed using the internal objects, and an external object was programmed to implement the particle filter algorithm. Figure 1 illustrates the different modules and the main procedures of the system. Before running the localization system, the microphone coordinates, measurement space dimensions, particle number and the variance of the movement model need to be given as parameters. The latter parameter determines the spread of the particles at the prediction stage.

The system employs $A = 3$ microphone arrays, each consisting of $m = 4$ microphones. The acquired audio data is first read in frames of $L = 1024$ samples and then passed to GCC-PHAT module. There the frames are Hanning-weighted in order to avoid spectral leakage, and then processed into source evidence by making the $M = 6$ inter-channel comparisons within an array using the equation (2), thus resulting in $A \times M = 18$ GCC-PHAT vectors. The asymptotic time com-

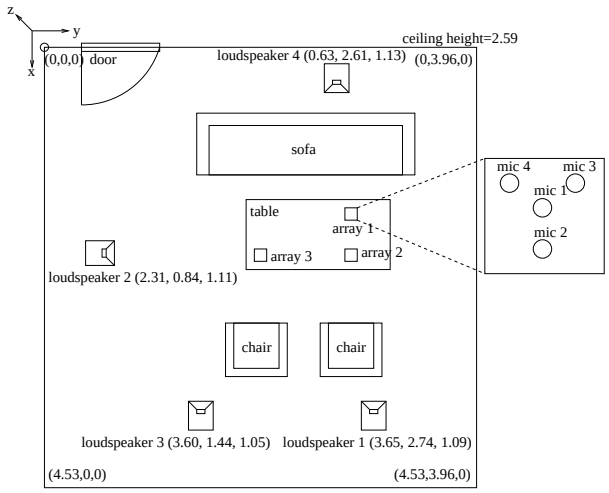

Figure 2: Recording room configuration. The microphones are located on the wooden table, surrounded by the sofas and the loudspeakers acting as sound sources. The microphone arrays consist four microphones in Y-shaped geometry.

plexity of generating the source evidence at each time step κ is presented in Table 1. The exact algorithmic implementation of the fast Fourier transform (FFT) in PD is unknown, and therefore an estimation of the time complexity based on a well-known and widely used algorithm (Radix-2) is presented.

After generating the source evidence, the GCC-PHAT vectors are passed to the particle filter module. The particles are iteratively processed according to Algorithm 1. The GCC-PHAT vectors are used as look-up tables to assign a weight to each particle. The likelihood values from different GCC-PHAT vectors are combined using equation (3). The asymptotic time complexity of the main procedures in the implemented particle filter is presented in Table 2.

Using $P = 500$, $A \times M = 18$, and $L = 1024$, the CPU load was around 50% during computation as opposed to 5% load of idle time. This implies a calculational load of approximately 8.8×10^9 FLOPS, which is high compared to, e.g., the method presented in [5], but nevertheless computable using modern PCs. Moreover, the focus in this implementation was more on simplicity rather than computational efficiency. For example, in the estimation of median coordinate, sorting of particles was performed using computationally costly insertion sort algorithm.

5. REAL-DATA RESULTS

The most important properties of a sound source localization application are the ability to localize the sound source accurately and to quickly adapt to the change of the source location. In real-time applications location estimates need to be generated at high rate.

The system proposed in this paper was tested using real speech samples from the TIMIT database consisting of male and female speech. Three different multi-channel sequences were created from the samples and played through four Genelec 1029 A active loudspeakers with sound pressure of 80 dB, each loudspeaker transmitting separate channel. The active loudspeaker changed between the different speech samples to emulate a human discussion. The first sequence consisted of male speech, and the second sequence of female speech. The third sequence consisted of both male and female speech with slightly overlapping samples. During playback, the soundscape was recorded using the microphones. The acquired signals were then sampled at 48000 Hz using 32 bits per sample and saved as separate mono WAV files.

Table 3: 2D tracking RMSE values.

Sequence	Content	2D RMSE [m]
1	Male speech	0.297
2	Female speech	0.278
3	Male+Female speech	0.249
Mean		0.275

The recording took place at the audio laboratory, located at the Tampere University of Technology, Department of signal processing. The room dimensions are $4.53 \times 3.96 \times 2.59$ m and the T_{60} reverberation time of the room is approximately 0.26 seconds. The room interior consists of a table, sofas, loudspeakers and other equipment. The microphones were placed on the table in three arrays in Y-shaped geometry. DPA 4060-BM prepolarized omnidirectional condenser microphones were used with a 48 V phantom feed. The recording setup is illustrated in Figure 2.

The recorded WAV files were used as input data for the tracking system. During the tracking procedure, the source location estimates (x- and y-coordinate) were saved in a text file for evaluation of tracking accuracy. 500 particles were used for the tracking and the measurement space height was limited to 2 meters resulting in particle density of approximately $14/m^3$. The variance of the movement model was 5×10^{-4}.

The tracking accuracy was evaluated by calculating the 2D root mean squared error (RMSE) of the source location estimate against the actual source position across the whole duration of the sequence. The tracking results of each sequence were averaged over three test runs. The results are presented in Table 3. A typical 2D tracking plot of a 4-source scenario is presented in Figure 3.

The mean RMSE of the tracking results is slightly less than 30 cm. However, RMSE is a very strict measure; for example, a small delay in the adaptation causes a brief but significant difference between the estimate and the ground truth. Based on the visual evaluation of the tracking plots the estimates remain mainly close to the ground truth, except for a slight systematic bias in the x-coordinate of loudspeaker 3, also visible in Figure 3. This can be caused by e.g. reverberation. During the slightly overlapping portions of the sequence 3, the location estimate oscillates between the locations of the active sources.

To estimate the effect of different parameter setups on the computational load, the processing time of a 24 second signal (sequence 1) was measured while using different number of particles and frame lengths, and averaged over three test runs. The averaged processing times were then normalized by the length of the input signal. Figure 4 illustrates the normalized processing times as a function of particle number, value 1 indicating real-time computing.

When using a frame of 512 samples the PF algorithm is

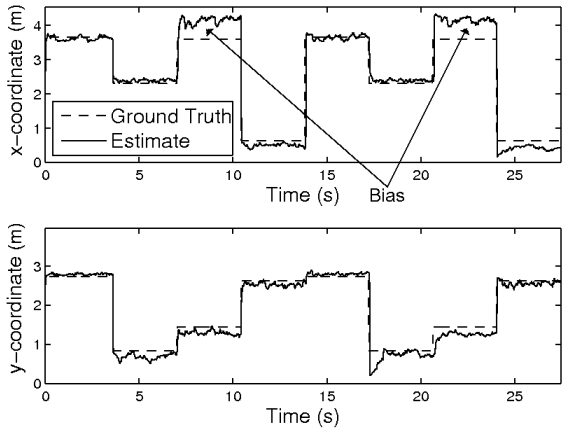

Figure 3: 2D tracking plot of using the data from sequence 2 as input. The different locations of sound sources are clearly visible and the adaptation to the change of location is rapid.

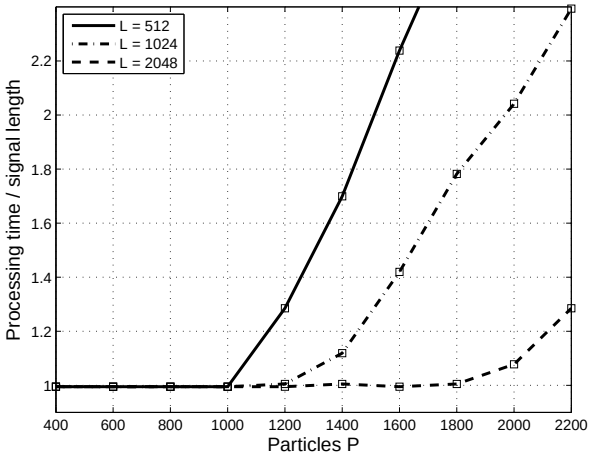

Figure 4: Normalized processing times for different frame lengths L as a function of particle number P.

executed approximately 94 times per one second of input signal as opposed to approximately 23 times when using a frame of 2048 samples. The use of a shorter frame thus increases the computational load, which must be compensated by decreasing the number of particles in order to generate location estimates in real-time.

6. CONCLUSIONS

In this paper a real-time version of a talker localization system proposed in [7] is presented. Tests with realistic input data indicate that the system is able to localize a sound source simulating a human talker and rapidly adapt to the change of source location. The effect of different input parameters on the computational load is examined and the results show that by using a modern PC, accurate location estimates can be generated in real-time by adjusting the number of particles or frame length.

REFERENCES

[1] M. 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, 2002.

[2] J. Chen, J. Benesty, and Y. Huang. Time delay estimation using spatial correlation techniques. In *Proceedings of the 8th international workshop acoustic echo and noise control (IWAENC '03)*, pages 207–210, Kyoto, Japan, September 2003.

[3] J. DiBiase, H. Silverman, and M. Brandstein. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter Chapter 8. Robust localization in reverberant rooms, pages 157–180. Editors: Michael Brandstein and Darren Ward, Springer-Verlag, 2001.

[4] H. Do, H. Silverman, and Y. Yu. A real-time srp-phat source location implementation using stochastic region contraction(src) on a large-aperture microphone array. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 1:121–124, April 2007.

[5] A. M. Johansson, E. A. Lehmann, and S. Nordholm. Real-time implementation of a particle filter with integrated voice activity detector for acoustic speaker tracking. In *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS'06)*, pages 1004–1007, Singapore, December 2006.

[6] C. Knapp and G. Carter. The Generalized Correlation Method for Estimation of Time Delay. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 320–327, 1976.

[7] T. Korhonen and P. Pertilä. TUT Acoustic Source Tracking System 2007. In *Second Annual International Evaluation Workshop on Classification of Events, Activities and Relationships*, 2007.

[8] E. A. Lehmann. *Particle Filtering Methods for Acoustic Source Localisation and Tracking*. PhD thesis, Research School of Information Sciences and Engineering, Department of Telecommunications Engineering, The Australian National University, Canberra, ACT, Australia, July 2004.

[9] E. A. Lehmann and A. M. Johansson. Particle filter with integrated voice activity detection for acoustic source tracking. NICTA/WATRI Technical Report PRJ-NICTA-PM-008, Western Australian Telecommunications Research Institute, Perth, Australia, December 2006.

[10] P. Pertilä, T. Korhonen, and A. Visa. Measurement Combination for Acoustic Source Localization in a Room Environment. *EURASIP Journal on Audio, Speech, and Music Processing*, 2008, 2008.

[11] C. Zhang, D. Florencio, and Z. Zhang. Why Does PHAT Work Well in Lownoise, Reverberative Environments? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2565–2568, 2008.

[12] J. M. Zmölnig. Pure Data Community Site. http://puredata.info. Last checked: 5.2.2009.