A Real-Time Application for Sound Source Localization Inside a Spherical Microphone Array

Tim Lübeck[1], Johannes M. Arend[1,2], Christoph Pörschmann[1]
[1] TH Köln, Institut of Communications Engineering, Cologne, Germany
[2] TU Berlin, Audio Communication Group, Berlin, Germany
Email: tim.luebeck@th-koeln.de

Introduction

Dynamic binaural synthesis is a common method to present a *virtual acoustic environment* (VAE) over headphones. For auralization, anechoic audio signals are convolved with HRIRs (Head-Related Impulse Responses) or BRIRs (Binaural Room Impulse Responses) according to the head orientation of the listener. In the ongoing research project NarDasS (Natürliche raumbezogene Darbietung selbsterzeugter Schallereignisse in virtuellen auditiven Umgebungen, english: "Natural room-related reproduction of self-generated sound in virtual acoustic environments"), we developed a system based on dynamic binaural synthesis to reproduce any self-generated sound in a VAE[1].
To realize an adequate reproduction, a 32-channel surrounding spherical microphone array captures the direction-dependent sound of an acting user. By convolution of these captured signals with specific BRIRs, a binaural room response is synthesized in real-time, and presented over headphones. The system is designed for a sound source located in the center of the array. However, in real life, the user and in particular the sound source might be slightly off-center. To compensate for this offset, the level of the microphone signals needs to be adjusted according to the 1/r distance law. This requires knowledge on the exact position of the sound source.
This work presents the algorithm and C++ implementation for sound source localization inside the microphone array in real-time. The *time differences of arrival* (TDOAs) between the microphone signals are calculated by a cross-correlation with phase transform weighting, and a linear equation system is set up. This equation system is then solved applying the least square method with QR-decomposition. The calculated position can be used for level adjustment of the microphone signals.
The paper is structured as follows. First, we give a brief overview of common sound source localization methods. Next, we outline the principles of TDOA-based localization. Then follows a section about the actual real-time implementation in C++. Finally, we analyze the localization accuracy for several test cases and real-life scenarios in a technical evaluation and conclude the paper with a short summary and outlook.

The Microphone Array

A user is surrounded by a spherical microphone array to capture the self-generated sound. The fiberglass rod framework of the array has a diameter of 2 meters and holds 32 Rode NT5 microphones. As described by Arend et al.[1], the array is constructed based on a pentakis dodekaedron. This array geometry results in a microphone spacing of about 70 cm. The entire structure is set up in the anechoic room at TH Köln.

Passive Source Localization

Passive acoustic source localization is an important area of ongoing research and has a wide field of applications. Human-computer interaction, speech enhancement, or noise suppression, for example, all require knowledge about the sound source position. Equally, non-acoustic position estimations used for mobile communications or GPS are based on similar localization methods.
Broadly speaking, there are three common methods for passive localization [2],[3]. Two possible ways are *beamforming* and *high-resolution spectral estimation* approaches, which are appropriate for far-field localization of multiple sources. These methods require a lot of computational power and most of all a narrow sensor spacing to prevent spatial undersampling. The third group are *TDOA-based* approaches, which are quite popular because of their small computational performance requirements. These low requirements are a significant advantage, especially with regard to real-time implementation. Furthermore, the given microphone distance of about 70 cm make a good case for choosing the TDOA approach for our project. In the next section, the approach will be explained in more detail.

TDOA-Based Localization

The propagation time from source to microphone, or for non-acoustic localization to any sensor, can be described by the *time of flight* (TOF) or *time of arrival* (TOA). Since there is no information about the absolute transmission time from source to receiver in passive localization, the TDOA between two sensors is considered. The distance traveled by the sound during this time difference can be calculated by multiplying this time difference with the speed of sound c (assumed as 343 $^{m}/_{s}$). Based on these distances, the source position can be estimated, as shown in section Setup of the Equation System.

Time Delay Estimation

The accuracy of the final position estimation depends on the proper determination of the TDOAs. The cross-correlation function is commonly used for this purpose [3]. Figure 1 depicts the basic principle of the *time delay*

estimation (TDE). The time shift between two similar signals (shown in Figure 1 a and b) can be obtained by calculating the cross-correlation of these signals (c). The cross-correlation is maximal at exactly this point where both signals have no phase shift. The deviation from the origin to the maximum point corresponds to the time delay between signal one and two.

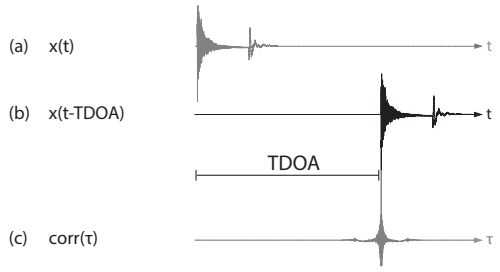

Figure 1: Principle of TDOA determination. The second graph shows the same signal as the first graph with a time delay. This time delay can be obtained by finding the maximum of the cross-correlation (corr(τ)) and determining the deviation from the origin.

Phase Transform

In case of noise, reflections, and other background sounds, the correlation exhibits large side lobes. As a result, it might be hardly possible to determine an explicit time delay. To prevent this effect, Knapp and Carter [4] introduced a pre-filtering technique, which is applied in the frequency domain. Thus, the cross-correlation is first transformed to frequency domain:

$$s_{12}(\tau) = (x_1(-t)^* * x_2(t))(\tau) \quad (1)$$

$$S_{12}(\omega) = X_1^*(\omega) \cdot X_2(\omega). \quad (2)$$

Next, the resulting *cross power spectral density* (CPSD) can be weighted with a weighting function $\Psi(\omega)$:

$$S_{12_{weight}}(\omega) = \Psi(\omega) \cdot X_1^*(\omega) \cdot X_2(\omega). \quad (3)$$

The most typical weighting function is the *phase transform* (PHAT) weighting $\Psi_{PHAT}(\omega)$. It is basically a division by the absolute value of the CPSD:

$$S_{12_{PHAT}}(\omega) = \frac{X_1^*(\omega) \cdot X_2(\omega)}{\left|X_1^*(\omega) \cdot X_2(\omega)\right|} \quad (4)$$

In general, the weighting results in a flat magnitude response (pre-whitening) but does not have any effect on the phase response. Since the phase response contains all information about the time shift, and the PHAT-weighting does not modify the phase, this weighting delivers a single peak at the accurate time delay point and a more precise time delay estimation is possible even under bad conditions. Knapp and Carter denote this pre-filtered cross-correlation technique as GCC-PHAT.

Setup of the Equation System

There are several ways to use the determined TDOAs for sound source localization. Approaches which estimate the direction of the sound incidence are called *direction of arrival* (DOA) or *angle of arrival* (AOA) methods. However, for this work, we used a method introduced by Mahajan et. al. [5]. They showed a feasible way to set up a linear equation system, which will be derived in the following. Figure 2 shows exemplarily the basic arrangement of two sensors and one source.

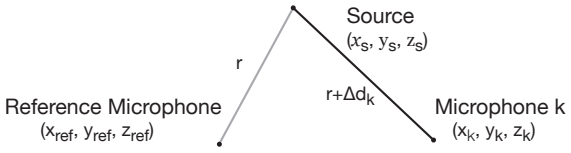

Figure 2: Simplified microphone arrangement of two microphones and a source to derive the basic equations to setup a equation system.

The distance between the source with the coordinates x_s,y_s,z_s and the reference microphone is denoted by r. The distance from the source to another microphone, denoted by index k, is about Δd_k longer than r, where Δd_k can be obtained by $\Delta d_\mathrm{k} = c \cdot TDOA_\mathrm{k}$. These distances, expressed by the Pythagoras theorem, lead to Equation 7.

$$r^2 = (x_\mathrm{ref} - x_\mathrm{s})^2 + (y_\mathrm{ref} - y_\mathrm{s})^2 + (z_\mathrm{ref} - z_\mathrm{s})^2 \quad (5)$$

$$(r + \Delta d_\mathrm{k})^2 = (x_\mathrm{k} - x_\mathrm{s})^2 + (y_\mathrm{k} - y_\mathrm{s})^2 + (z_\mathrm{k} - z_\mathrm{s})^2 \quad (6)$$

$$\Rightarrow x_\mathrm{k}^2 - 2x_\mathrm{k}x_\mathrm{s} + y_\mathrm{k}^2 - 2y_\mathrm{k}y_\mathrm{s} + z_\mathrm{k}^2 - 2z_\mathrm{k}z_\mathrm{s} - 2r\Delta d_\mathrm{k} = x_\mathrm{ref}^2 + y_\mathrm{ref}^2 + z_\mathrm{ref}^2 + \Delta d_\mathrm{k}^2 \quad (7)$$

Substituting Equation 5 in Equation 6 has the effect that there is no unknown squared variable in Equation 7 and a linear equation system can be set up. This equation contains three unknown source coordinates and additionally the unknown distance r. An explicit solution requires four equations and thus five TDOA measurements. These equations can be expressed by an equation system in the form of $A \cdot x = b$ with $A \in \mathbb{R}^{4\times4}$ and $b, x \in \mathbb{R}^4$ such as:

$$A = \begin{bmatrix} 2(x_\mathrm{ref} - x_2) & 2(y_{ref} - y_2) & 2(z_\mathrm{ref} - z_2) & -2\Delta d_1 \\ 2(x_\mathrm{ref} - x_3) & 2(y_{ref} - y_3) & 2(z_\mathrm{ref} - z_3) & -2\Delta d_2 \\ 2(x_\mathrm{ref} - x_4) & 2(y_{ref} - y_4) & 2(z_\mathrm{ref} - z_4) & -2\Delta d_3 \\ 2(x_\mathrm{ref} - x_5) & 2(y_{ref} - y_5) & 2(z_\mathrm{ref} - z_5) & -2\Delta d_4 \end{bmatrix}$$

$$x^T = \begin{bmatrix} x_\mathrm{s}, y_\mathrm{s}, z_\mathrm{s}, r \end{bmatrix}$$

$$b = \begin{bmatrix} x_\mathrm{ref}^2 + y_\mathrm{ref}^2 + z_\mathrm{ref}^2 + \Delta d_1^2 - x_2^2 - y_2^2 - z_2^2 \\ x_\mathrm{ref}^2 + y_\mathrm{ref}^2 + z_\mathrm{ref}^2 + \Delta d_2^2 - x_3^2 - y_3^2 - z_3^2 \\ x_\mathrm{ref}^2 + y_\mathrm{ref}^2 + z_\mathrm{ref}^2 + \Delta d_3^2 - x_4^2 - y_4^2 - z_4^2 \\ x_\mathrm{ref}^2 + y_\mathrm{ref}^2 + z_\mathrm{ref}^2 + \Delta d_4^2 - x_5^2 - y_5^2 - z_5^2 \end{bmatrix}$$

Solution of the Equation System

Since the array provides 32 microphones, it is technically possible to determine 31 TDOAs leading to 31 equations. Usually, this modified, overdetermined equation system $\tilde{A} \cdot x = \tilde{b}$ with $\tilde{A} \in \mathbb{R}^{4\times31}$ and $b \in \mathbb{R}^{31}$, which has no unique solution, can be solved with the *least square* (LS) method. As described by Damen and Reusken [6], a regression curve can be approximated by minimizing the sum of all squared deviations (the residuals) of measurement data and regression curve points. This

minimization can be expressed by Equation 8. Rearranging this equation leads to Equation 9.

$$||\tilde{A}x - \tilde{b}||^2 \tag{8}$$
$$\Leftrightarrow A^T Ax = A^T b \tag{9}$$

To get a more stable and well-conditioned equation system, it is beneficial to perform the minimization using the QR-decomposition. An arbitrary matrix A is decomposed into the orthogonal matrix Q for which $Q^T Q = \widetilde{1}$ (with $\widetilde{1}$ = unit matrix) holds and an upper triangular matrix R.
Inserting the QR-decomposition in the minimization (Eq. 8) leads to Equation 10 where R_0 and c_1 denotes the upper four lines in R and $Q^T \cdot b$.

$$\begin{aligned}
&\| QRx - b\|^2 \\
=&\| Rx - Q^T b\|^2 \\
=&\| \begin{bmatrix} R_0 \\ 0 \\ s \end{bmatrix} x - Q^T b\|^2 \\
=&\| \begin{bmatrix} R_0 x \\ 0 \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \|^2 \\
=&\| R_0 x - c_1\|^2 + \|c_2\|^2
\end{aligned}$$

The minimization problem can be simplified to $R_0 x - c_1$ and solved by the explicit resolvable equation system:

$$R_0 x = c_1 \tag{10}$$

Real-Time Implementation

To integrate the localization application into the NarDasS-system, it has to be real-time capable. The implementation of the localization algorithm is based on the JUCE Framework[1] and in particular on the audioDevice-callback class. This class periodically provides a buffer of samples. The number of samples delivered per period depends on the buffers size. As a good compromise between computational workload and overall round-trip latency, we set the buffer size to 128 samples (at a sample rate of 48 kHz) during development.

Figure 3 shows the signal flow of the localization application. As depicted, the localization module is part of the NarDasS-Processor, which additionally performs adaptive filtering of one input signal to get the reverberation excitation signal as well as the level adjustment of the microphone signals depending on the sound source position (see Arend et al.[1] for a more detailed description). In general, the NarDasS-Processor passes the current Fourier-transformed buffer to the localization application algorithm and gets back the current position of the sound source in cartesian coordinates.

[1]JUCE Framework: version: v5.1.2, access: 26.10.17, `https://www.juce.com`

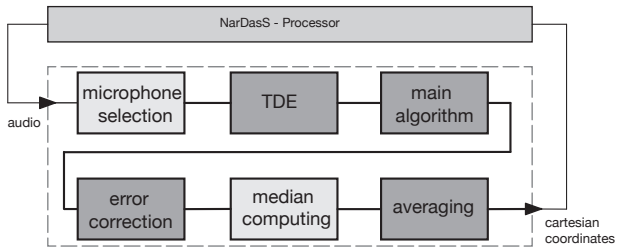

Figure 3: The block diagram shows the signal flow of the C++ application. The NarDasS-Processor passes the input signals to the localization application. As a result, the localization algorithm delivers the estimated position in cartesian coordinates.

The choice of the buffer-length needs to allow the detection of all TDOAs occurring in typical situations. The array diameter of 2 m leads to a maximum TDOA of 5.8 ms and thus requires at least 512 samples (at a sample rate of 48 kHz). Therefore, a sufficiently large process buffer must be filled, before passing it to the localization algorithm. We achieved the best results with a minimum size of 4096 samples.
The first block of the localization algorithm is the microphone selection, which is related to the block median computing (as illustrated by the coloring). As described so far, it is possible to determine 31 TDOAs with respect to one reference microphone and estimate one position, based on these 31 TDOAs. However, we achieved more stable results by choosing more reference microphones, determining 31 TDOAs respectively, and computing one source position for each reference microphone. Subsequently, the median is calculaded over all estimations. The number of reference microphones and thus the number of position estimations per block can be adapted according to requested performance and accuracy. The current implementation uses nine reference microphones on the horizontal plane.
The TDE calculates as many TDOAs as determined by the microphone selection and passes them to the main algorithm. This block sets up the linear equation system and solves it using the introduced LS method with QR-decomposition. To implement this solving procedure, we used a Householder transformation from the EIGEN library [2]. The Householder transformation is a widely used tool to implement stable and fast decompositions. Result of the main algorithm is one position estimation for each reference microphone.
The block error correction checks the plausibility of the estimated position and compensates for inaccurately determined TDOAs. According to a fixed buffer size and an assumed maximum velocity of the tracked source, it is possible to suppose a maximum distance the source can move during one computation. If the distance of the current estimation to the last estimation exceeds this maximum distance, the current estimation is assumed to be improbable. Furthermore, this block checks if the estimated position is located inside the array. Estimations outside the array are assumed to

[2] EIGEN library: c++ library for linear algebra and numeric , version: 3.34, access: 27.10.17, `http://eigen.tuxfamily.org`

be improbable too. In case of these unsuitable results, the algorithm discards them and passes the last valid estimation to the median computing.
The error correction is followed by the median computing-block, which delivers the median value from all estimations, as described above.
The final averaging-block is especially relevant for the localization of moving sources. It calculates the average of the preceding estimations. As shown in Figure 4, the averaging weakens the outliers and yields to a smoother tracking of the source. Most of all, for the following level-adjustment, it is highly relevant to prevent large position jumps.

Technical Evaluation

Since the application of the NarDasS-system aims at reproducing natural sound sources, the technical evaluation of the localization algorithm is based on the signals speech, guitar, drums, and white-gaussian noise. All signals were normalized in level and played back via a JBL Clip consumer loudspeaker (dimensions: $8\,\text{cm} \times 8\,\text{cm} \times 3\,\text{cm}$). Table 1 shows the localization results for all signals separatly. For each result the average over localizations at four different array test positions was calculated. The localization error is obtained by computing the Euclidian distance between measured and estimated source position. Additionally, the table lists the *mean absolute error* (MAE) over all signal types. To investigate the influence of the averaging, we compared the results without averaging (denoted as a_{none} in the table) with the same calculations with an averaging over the last two (a_2) and four (a_4) positions. As can be seen, the algorithm reaches a mean accuracy of 4.8 cm. Considering the dimensions of the test source (the JBL loudspeaker) and inevitable measuring inaccuracies, a deviation of about 5 cm is a quite satisfying result. Averaging over the last estimations (as shown in Table 1 for condition a_2 or a_4) yields even more precise results.

Table 1: Estimation results with different averagings. MAE = Mean Absolute Error, avg = averaging over all positions, $a_{\text{none/2/4}}$ = averaging over the preceding estimations.

	avg error [cm] a_{none}	avg error [cm] a_2	avg error [cm] a_4
Drums	0.1078	0.1060	0.1045
Flamenco	3.9523	3.8947	3.8713
Speech	10.4693	9.927	9.7203
Noise	4.8062	4.6885	4.6558
MAE	4.8339	4.6541	4.5879

Moving Sources

The measurements in the previous section referred to static sources only. To examine the performance of the algorithm for moving sources, a pendulum motion of the sound source has been tracked. For this, the loudspeaker played back the speech signal while swinging from the side to the origin of the array. Figure 4 shows the localization results without averaging (a) and the same results with an averaging over the last four estimations (b). As can be seen, the averaging compensates for some outliers. Overall, the results make clear that the algorithm can achieve accurate results for slow moving sources.

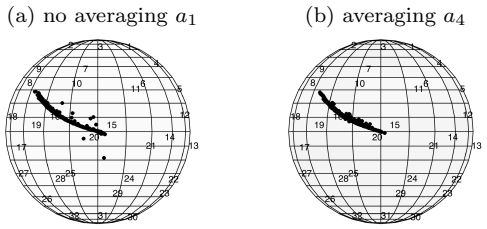

Figure 4: Localization results of a moving loudspeaker inside the array without averaging (a) and with averaging over four preceding estimations (b).

Conclusion

In this paper, a real-time capable TDOA-based localization algorithm has been presented. We tested the localization algorithm for the signals drums, speech, guitar, and white-gaussian noise and achieved an accuracy of about 5 cm for non-moving sources. Moreover, it is possible to track slow movements. The application has been integrated into the NarDasS-Processor and produces sufficiently accurate results to adjust the level of the microphones in reasonable time.
Up to now, it is impossible to track multiple sources at the same time. This might be useful to realize a conversation of two persons in a VAE, for example. Thus, the tracking of multiple sources could be a first refinement of the present implementation. In addition, the localization algorithm has been tested inside an anechoic room without multipath propagation. For a wider field of application, the algorithm has to be checked in various environments.

References

[1] Arend, J. M., Stade, P., and Pörschmann, C., "Binaural reproduction of self-generated sound in virtual acoustic environments," *Proceedings of Meetings on Acoustics*, 30(1), pp. 1–14, 2017.

[2] Brandstein, M. S., *A Framework for Speech Source Localization Using Sensor Arrays*, Ph.D. thesis, Massachusetts Institute of Technology, 1990.

[3] Swartling, M., *Direction of Arrival Estimation and Localization of Multiple Speech Sources in Enclosed Environments Mikael Swartling*, 2012.

[4] Knapp, C. H. and Carter, G. C., "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4), pp. 320–327, 1976.

[5] Mahajan, A. and Walworth, M., "3-D position sensing using the differences in the time-of-flights from a wave source to various receivers," *IEEE Transactions on Robotics and Automation*, 17(1), pp. 91–94, 2001.

[6] Dahmen, W. and Reusken, A., *Numerik für Ingenieure und Naturwissenschaftler*, Springer-Verlag, Berlin Heidelberg, 2nd edition, 2006.