PERFORMANCE COMPARISON BETWEEN GENERALIZED CROSS-CORRELATION TIME DELAY ESTIMATION AND FINGERPRINTING METHOD FOR ACOUSTIC EVENT LOCALIZATION

Anantha Alifia Putri1, Dirgantoro Kevin Putra1 and Wansu Lim2
1Telkom University, Bandung, Indonesia
2Department of IT convergence, Kumoh National Institute of Technology, South Korea
E-Mail: wansu.lim@kumoh.ac.kr

ABSTRACT
Using acoustic signals for localization is useful for any applications such as detecting trapped people during emergency situations by processing the sounds that are emitted in the face of danger. Different acoustic localization methods have been developed and proposed; each with its own advantages and disadvantages. Thus, this paper presents a performance comparison between two common Time Delay of Arrival (TDOA) based acoustic event localization methods: Generalized Cross-correlation Time Delay Estimation and fingerprinting. The experiment results show that the accuracy of sound source localization using GCC-PHAT TDE is high and PHAT is effective weighting function for GCC.

Key words: acoustic event localization, GCC time delay estimation, fingerprinting.

INTRODUCTION
Acoustic source localization plays an important role in many applications such as Global Positioning System (GPS), telecommunication system, seismology, navigation, video games, hearing aid, tracking personal possessions, radar, sonar, musical control. [1] [2] [3] [4] [5] [6]. In certain situations, localization information may be a matter of life and death, for example, tracking people trapped in emergency situations such as in a fire [7]. The aim of acoustic localization is estimating and determining the position of radiating sound sources.

Acoustic localization has been used since 1974 [8]. During the recent decades, various localization methods have been proposed to estimate the location of the targets [9]. These methods include fingerprinting, Doppler shift, evolutionary optimization, Time Delay Estimation (TDE), etc. [5][7][10][11]. The implementation of these different methods has been a research topic to find which method is more effective to find the acoustic source.

Fingerprinting method is the most popular acoustic localization based on Wireless Sensor Network (WSN) [12]. WSN provide reliable methods for monitoring physical environments in real-time due to their unique characteristics. In more detail, the WSN is formed by nodes containing an embedded processor (Arduino), an acoustic transducer (microphone), and Bluetooth shield [7]. Most fingerprinting localizations conducted have used signals like WLAN (Wi-Fi), RF and ultrasound signals, but audible sound is used less often [13]. Fingerprinting localization method comprises of an offline phase and an online phase. In the first (offline) phase, the area of interest is divided by a grid, and for each section range measurements from multiple WSNs are collected. The resulting vector of measurements is called the fingerprint of that location. In the second (online) phase, the target composes a sample vector of measurements at its current position and reports it to a server. The server then matches the sample with the fingerprints generated in the offline phase and estimates the target’s location [7].

The other method is Time Delay Estimation. Time Delay Estimation is the first and also most important step in the acoustic source localization algorithm. Subsequently, mathematical methods are used for data processing to estimate the relative time delay of signal arrival to each sensor. Finally, the time-delay estimate is used to calculate the acoustic source location. There are many time-delay estimation algorithms, but Generalized Cross-correlation (GCC) function is more widely used in practice and relatively simple. GCC Time Delay Estimation has many weighting functions such as Cross Correlation (CC), Smoothed Coherence Transform (SCOT), Phase Transformation (PHAT), Roth Processor (ROTH), Eckart Filter, and Maximum Likelihood (ML) estimator [14]. The best and commonly used Time Delay Estimation algorithm based on the direction of arrival estimation is the Phase Transformation (PHAT) algorithm [15]. This is because of its small fluctuation, sharp peak and strong anti-jamming ability. Computation of the time delay between signals from any pair of microphones can be performed by first computing the cross-correlation function of the two signals [16]. The lag at which the cross-correlation function has its maximum is taken as the time delay between the two signals. It then computes the signal of the acoustic sound by using a sample signal of Digital Fourier Transformation (DFT) [16]. After the sample signal is computed, the cross-correlation signal will be defined by the inverse of DFT from the measured time delay.

This paper compared the performance of the two most commonly used methods for localizing the source of an acoustic event. Both of these methods don’t have any big differences on their configurations. Fingerprinting localization requires several WSNs that connected to the server, and so does Time Delay Estimation. The parameter that was used to compare the methods are range detection, acoustic source node detection speed, and sensitivity of...
low frequency. The reason why sensitivity of low frequency is used as a parameter here is because in [17] it was explained that low-frequencies are cannot be localized, but in [18] it was turned out that low-frequency is possible to localize. Therefore, this paper proved are those two methods capable to localize low frequency or not and compare which one is doing it better.

RELATED WORK

In the acoustic localization field, the processing of acoustic signals using fingerprinting and Time Delay Estimation method is a very common thing. Several works like [7] [15][19]are examples for the implementation of both methods.

For instance, M. Perez and E. Carrera proposed an acoustic event localization using Bluetooth which implemented in the sensor network due to its easy connection procedure and low cost. Their experiment was done inside an area of 2x2 m with six wireless sensor network nodes and a server. The node density was set to 6 according to the amount of available hardware and the reduced physical space at the deployment location. At the end, experimental results showed that fingerprinting is a suitable method for locating acoustic events due to its lower localization error of 22.7 cm obtained with a 121-point database inside an area of 2x2 m [7].

On the other hand, [15] proposed acoustic source localization based on Generalized Cross-correlation Time-delay Estimation. In order to make the time delay estimation not affected by the characteristics of the signals themselves and to suppress reverberation and noise as much as possible, weighting function is needed. The paper analyzed the performance of several weighting functions in the generalized cross-correlation time-delay estimation algorithm and the result showed that PHAT weighting is the best choice for acoustic source localization in the generalized cross-correlation time-delay estimation algorithm due to its small fluctuations, sharp peak and strong anti-jamming ability.

For a clearer explanation, [19] discussed about GCC-PHAT TDE in more details. They implemented sound source localization with TOA (Time of Arrival) estimation and was able to determine the delay difference in sound arrival at two microphones based on location. The TOA (Time of Arrival) estimation value is used to determine exact angle of incidence of sound. Their result showed that GCC-PHAT calculation is fast and efficient for real time applications.

Furthermore, the implementation of several frameworks mentioned before would be implemented also in this paper.

SYSTEM DESIGN

Fingerprinting

The fingerprinting localization is a method of predicting location on basis of pre-recorded measurements of received signal strength (RSS) in its database. The measured RSS values are treated as random variables, which are statistically dependent on the location. The task is to predict the unknown location of the tracked WSN. This proposed acoustic localization system determines the target’s position through the interaction of four main frameworks such as, the parameter measured, the hardware/software infrastructure, protocols that coordinate the localization process, and the localization techniques.

The first framework in the proposed system is considering power measurements obtained from the intensity of acoustic signals as the observed parameter. The second framework refers to the infrastructure which is based on a wireless sensor network. In this method, the acoustic event is defined as the sound target that should be localized by each WSN. Each WSN consist of an acoustic transducer or microphone; that can detect precisely where the sound target is, a wireless communication device (Bluetooth shield); that is used to communicate to other node and server, and an embedded processor; that handles all the computation and logical operation, storing and retrieving data, and processing data. WSNs are integrated in order to establish connectivity among the whole network and also implement a synchronization technique.

Addressing the third framework, the protocols that coordinate the localization process primarily refer in this system to the synchronization technique to be implemented in the WSN. The server is able to order and associate the synchronization values in a way that time-related measurements can be passed to the localization techniques. These synchronization values can be gathered from every network that has global clock synchronization, the information about which node is detected the acoustic event together with its local timestamp will be sent from that node to the server. The clock differences for each system is important to determine how the observed/measured variables change over time.

At the last framework, the corresponding position estimation will be finally displayed on a two-dimension Cartesian coordinate plane after the information of a certain acoustic event is effectively related to its real time of occurrence.

There are two steps how fingerprinting method can localize the acoustic event target precisely.

Offline phase

This phase (also known as calibration phase) involves building a Received Signal Strength (RSS) database and creating a map (divided by a grid). Reference RSS points will be chosen after creating an accurate database of every WSNs locations are done. The distances are calculated from the known locations to the WSNs.

Online phase

This phase (also known as positioning phase) is process of position estimation measured unknown position of the acoustic event source. It then compared with RSS records in database and searched for the closest matching RSS reference. The closest matching RSS reference from the database is then treated as position estimation.

The basic idea of how fingerprinting works is to show the process of online localization estimation of the enhanced classical Euclidean based on weighted k-Nearest
Neighbors algorithm. With looking the similarities between the appropriate RSS vectors between tracked WSN and the selected reference points from the database. The similarity is calculated as the Euclidian distance $p_j$ defined by the following equation:

$$p_j = \sqrt{\sum_{i=1}^{N} (s_{ti} - s_{ji})^2}$$  

(1)

The smallest $p_j$ has a nearest reference point to the WSN, when it has a nearest reference point to the WSN then it would be having the most similar RSS vector too with the RSS vector of the tracked WSN. It is also important to determine the number of reference points (k) that is already saved in the database (offline phase) and it is used for an accurate coordinate localization.

The Weighted k-Nearest Neighbor (WkNN) algorithm is a common fingerprint-based method to increase accuracy for predicting changes of data and it has its own rule. This algorithm has advantage of high accuracy and easiness of realization. It is suitable too for fingerprinting localization technique because it acts as the standard if measured position is included in multiple cells. We can get the value of WkNN from this equation:

$$w_j = \frac{1}{p_j^2}$$  

$$\sum_{i=1}^{k} \frac{1}{p_i^2}$$  

(2)

With the unknown coordinate of the tracked WSN can be estimated by this expression $x = \sum_{j=1}^{k} w_j x_j$

**GCC Time delay estimation**

Generalized cross-correlation is a technique that obtained an estimated delay by finding the time-lag that maximizes the cross-correlation between the filtered versions of the two received signals. The filters have a function to ensure a large sharp peak in the obtained cross-correlation thus ensuring a high time delay resolution in generalized cross-correlation method.

The basic principle of the generalized cross-correlation function is: obtain the cross-power spectrum between two groups of signals, then give different weighted operations in the frequency domain and finally inverse transform to the time domain and obtain the cross-correlation function between the two groups of signals, the corresponding time of the extreme of the cross-correlation function is the time delay between the two groups of signals.

For Direction of Arrival (DOA) estimation, Generalized Cross-correlation method especially PHAT (Phase Transform) is commonly used pre-filter for GCC. A frequency-domain implementation of GCC-PHAT will necessarily compute 2N correlation points at once. One embedded processor is connected to the microphone array (consist of three microphone module).

Let the signals on two points (microphones) be $x_i(n)$ and $x_j(n)$ where $n$ is a time-sample index and for the Discrete Fourier Transform of the signals let be $X_i(k)$ and $X_j(k)$ where $k$ is a frequency-sample index. By doing a convolution of Fourier Transforms, a Cross-Power Spectrum will be obtained and can be expressed:

$$c_{x_i,x_j}^{(g)}(\omega) = (H_i(\omega)X_i(\omega)) \cdot (H_j^*(\omega)X_j^*(\omega))$$  

(3)

When the inverse Fourier transform applied to the equation (3), we can get the GCC function as follows:

$$\hat{c}_{x_i,x_j}^{(g)}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_{x_i,x_j}(\omega)X_i(\omega)X_j^*(\omega)e^{-j\omega \tau}d\omega$$  

(4)

The number of samples of delay between the two signals can be determined when the time lag maximizes $c_{x_i,x_j}^{(g)}(\tau)$:

$$\hat{t}_{ij} = \arg\max_{\tau} \hat{c}_{x_i,x_j}^{(g)}(\tau)$$  

(5)

$c_{x_i,x_j}(\tau)$ is used to analyzes the similarity between two different signals for every time delay that computed ($\tau$) with a convolution we can determine it with:

$$c_{x_i,x_j}(\tau) = \alpha \cdot c_{x_i,x_j} \ast \delta (\tau - \tau_{ij}) + \epsilon_{n_i,n_j}(\tau)$$  

(6)

From equation (3), for uncorrelated noises $c_{x_i,x_j}(\tau) = 0$ the equation (6) can be a represented as a delta function but spread by the Fourier transform. By choosing a pre-whitening filter is an optimal strategy for the Fourier transform become a delta function but with no spread. PHAT pre-filtering is showing more interesting results, it has been shown to be effective in real situations, when the Phase Transform (PHAT) is being used as the pre-filter of GCC it:

$$\Phi_{x_i,x_j}^{PHAT}(\omega) = \frac{1}{|c_{x_i,x_j}(\omega)|} = \frac{1}{|X_i(\omega)X_j^*(\omega)|}$$  

(7)

By placing equal emphasis on each frequency, the PHAT weighting is sub-optimal under reverberation-free conditions.

**IMPLEMENTATION**

According to the previous section, the acoustic event localization system was physically implemented under the following specifications.

**Infrastructure**

Our proposed acoustic event localization method is using WSN. It consists of three network nodes arranged to resemble an equilateral triangle with a distance of 35 cm between the nodes. The network topology is based on a centralized scheme. This centralized scheme is attractive for our prototype due to its simplicity and lower programming complexity because it works with point-to-
point links between each node and the server. The real implementation should be done using network nodes which consist Arduino Uno hardware as the microcontroller and analog sound sensor V2 as the acoustic transducer. But in this paper, we just did a simulation using MATLAB and Arduino IDE.

Bluetooth was proposed in this paper because it has an easy connection procedure, low cost, secure, and easily administered through specific communication interfaces that guarantee reliability in the exchange of information [7]. A JY-MCU (HC-05) Bluetooth shield was attached on every network node and a USB Bluetooth adapter was included on the server. Bluetooth here has function as a link between network nodes with the server in order to communicate with each other.

Figure-1 shows the general scheme for the acoustic event localization. The sound source is represented by the speaker symbol. When sound source detected by WSN nodes, the information about which node is detected the acoustic event and its TOA will be sent from that node to the server wirelessly. Based on the location, node B will have a smaller value of TOA compared to the other two nodes, because node B has the closest distance to the sound source. At the end, the result of the localization (the position of sound source) will be displayed in the server on two-dimension Cartesian coordinate as represented by yellow dots in the figure.

Fingerprinting

Acoustic signal intensity

The analog sound sensor that attached in the network nodes receive acoustic signals as analog audio inputs into the Arduino, which in turn converts these inputs into 8-bit digital signals by employing the ADC hardware. Then, after collecting samples, the acoustic power is computed and transmitted along with its local timestamp to the server every 10 ms on average. The expression used to calculate the acoustic power \( p_i \) at each node \( i \), where \( x[n] \) is the vector of digital samples and \( N \) the total number of samples.

\[
p_i = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2
\]  

Coordination protocols

The first step in the coordination process is synchronization. The server sends periodic reset clock messages to all the network nodes every 5 seconds in order to maintain a common time reference. Each node manages its local time by increasing a local time counter every 10 ms, and clearing it after receiving a reset message. Therefore, the time between consecutive reset messages influences the synchronization accuracy and the exchange of acoustic event information among the whole network.

In addition, acoustic power measurements and their corresponding timestamps arrive to the server in disorder due to the network latency variance and the message queuing of the Bluetooth protocol. Considering these weaknesses, it is necessary to find a way to solve the problems, the server implements an ordering functionality to associate acoustic event information from all network nodes in a precise time related way, and therefore communicate consistent information to the localization methods.

Localization methods

In fingerprinting method case, database of reference points was created. The vector of acoustic power for each reference point was obtained from different measurements of acoustic power recorded for each of the 3 network nodes at the corresponding reference position. Furthermore, the WkNN(2) was also implemented here in order to determine the closest point to the occurrence of an acoustic event. WkNN algorithm estimates the difference between any acoustic power vector formed upon the occurrence of an event and the previously recorded vectors. As result, the k nearest neighbor classifier figures out the k reference points with lower differences and determines position estimation based on its corresponding coordinates.
GCC Time Delay Estimation

Segmenting signals

Before localizing the sound source, the GCC TDE starts by segmenting these signals into blocks first and after that applying the Discrete Fourier Transform (DFT) to each of the microphones. The discrete-time microphone signals are denoted into $x_i(n)$...$x_M(n)$ with $M$ is the number of the used microphones in the system (i.e. three microphones). Any source localization algorithm will operate on the DFT of each data block to produce a precise location estimate.

Figure-2. Experiment of comparison.

This is an expression of discrete-time microphone signals with the DFTs signal of each microphone after being segmented into blocks where the length of $N$ is:

$$\hat{\tau}_i = \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{ij}[k]X_{i,b}[k]X_{j,b}[k]e^{j2\pi kN}$$

$$\hat{\tau}_i = \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{ij}[k]C_{ij,b}[k] e^{j2\pi kN}$$ (11)

Capturing signals by the microphones

After that, combine the equation (11) with Discrete Correlation Theorem that expressed using Fast Fourier Transform algorithm for efficiently computed.

$$\hat{\tau}_i = Re\left[IFFT\left(\Phi_{ij}[k]X_{i,b}[k]X_{j,b}[k]\right)\right](\hat{\tau})$$

$$\hat{\tau}_i = Re\left[IFFT\left(\frac{X_{i,b}[k]X_{j,b}[k]}{X_{i,b}[k]}\right)\right](\hat{\tau})$$ (12)

Expression (12) shows how to implement the GCC-PHAT function based on the FFT by estimating the TDOA that captured by the microphones. For estimating the TDOA between those two microphones can be found by searching the lag from GCC-PHAT function ($\hat{\tau}_{ij,b}(\hat{\tau})$) which the value is maximum.

Experiment of comparison

The GCC and fingerprinting performances were compared using three parameters such as range detection, acoustic source node detection speed, and sensitivity of low frequency. Those parameters applied in both methods. A clearer explanation is described below.

Range detection

The result of the comparison with this parameter is the inability of a network node from both methods to detect the sound source within a certain distance. To know the comparison’s result of this parameter, the experiment was done by placing sound source in different distances. As we can see from the Figure-2, the first trial was done by placing the sound source in position number 1, which means 5 cm from the midpoint between network node B and network node C. For the second trial, we changed the distance for another 5 cm (further than before), and we’ve done the same thing continuously until the n trial, as far as possible to find how far the capability of network node from two different methods can localize the sound source precisely. So, the result of the comparison is based on the maximum distance (represented by $x$ in the figure) or maximum capability of fingerprinting and GCC TDE method to localize the sound source. The further the
distance of the sound source still detected, the better the performance of that method.

**Acoustic source node detection speed**

Besides determining how far the distance that each method can localize the sound source, we also determined how fast the responsible node from both methods that we use can localize the sound source. The second experiment didn’t have much differences from the experiment of previous parameter, the difference is the comparison was done by comparing each method’s information values of Time Delay of Arrival (TDOA) from the responsible node that sent to the server. So, we need to collect the information values of TDOA in each distance from each method. The smaller the TDOA value of the responsible node, the better the performance of that method.

**Sensitivity of low frequency**

The last parameter is the sensitivity of each method to detect low frequency sound source, we proved whether the two methods are capable to localize it or not. The experiment was done by placing the sound source at position number 2 in the figure. Then, the comparison was done by reducing the frequency of the sound source until it reached average lowest human voice which is 300 Hz. The more the ability of that method to detect it, the better the performance of that method.

**Performance evaluation**

**Fingerprinting**

Algorithm 1 shows fingerprinting localization pseudo code. A song from library was implemented here to be used as the audio signal. Line 9 explained the process of building references as a part of offline phase. Those references were stored in the database and will used in line 21 for matching process on the online phase. After did the matching process using the references and the data on the real time, the result of successful match location estimation will be found.

<table>
<thead>
<tr>
<th>Algorithm 1 Pseudo Code for Fingerprinting Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 input IModelService modelService = new InMemoryModelService // store fingerprints in RAM</td>
</tr>
<tr>
<td>2 input IAudioService audioService = new NAudioService //use NAudio audio processing library</td>
</tr>
<tr>
<td>3 input private readonly IFingerprintCommandBuilder fingerprintCommandBuilder = new FingerprintCommandBuilder</td>
</tr>
<tr>
<td>4 input IQueryCommandBuilder queryCommandBuilder = new QueryCommandBuilder</td>
</tr>
<tr>
<td>5 for StoreAudioFileFingerprintsInStorageForLaterRetrieval(string pathToAudioFile)</td>
</tr>
<tr>
<td>6 input track = new TrackData</td>
</tr>
<tr>
<td>7 if trackReference = modelService.InsertTrack(track)</td>
</tr>
<tr>
<td>8 then//create hashed fingerprints</td>
</tr>
<tr>
<td>9 hashedFingerprints = fingerprintCommandBuilder BuildFingerprintCommand From(pathToAudioFile) UsingServices(audioService)</td>
</tr>
<tr>
<td>10 BuildFingerprintCommand</td>
</tr>
<tr>
<td>11 From(pathToAudioFile)</td>
</tr>
<tr>
<td>12 UsingServices(audioService)</td>
</tr>
<tr>
<td>13 Hash</td>
</tr>
<tr>
<td>14 Result</td>
</tr>
<tr>
<td>15 if hashedfingerprints created</td>
</tr>
<tr>
<td>16 then//store hashes in the database for later retrieval</td>
</tr>
<tr>
<td>17 modelService.InsertHashDataForTrack(hashedFingerprints, trackReference);</td>
</tr>
<tr>
<td>18 for TrackData GetBestMatchForSong(string queryAudioFile)</td>
</tr>
<tr>
<td>19 secondsToAnalyze ← 10 // number of seconds to analyze from query file</td>
</tr>
<tr>
<td>20 startAtSecond ← 0 // start at the beginning</td>
</tr>
<tr>
<td>21 if queryResult = queryCommandBuilder.BuildQueryCommand From(queryAudioFile, secondsToAnalyze, startAtSecond) UsingServices(modelService, audioService)</td>
</tr>
<tr>
<td>22 Query()</td>
</tr>
<tr>
<td>23 Result</td>
</tr>
<tr>
<td>24 then</td>
</tr>
<tr>
<td>25 return queryResult.BestMatch.Track // successful match has been found</td>
</tr>
</tbody>
</table>
GCC TDE

The figures below are the result of GCC TDE estimation using MATLAB with our proposed design structure. There are three microphones arranged to resemble an equilateral triangle with a distance of ±35 cm between them. An audio file from library was also implemented on the code to be used as the sample audio signal. Before computing the audio signal with Fourier transform, the longer time signal should be divided into shorter segment that usually called Short Time Fourier Transform (STFT). After getting the STFT’s parameter, we can get the captured signal using (12). At the end, the TDOA’s estimation was obtained based on the reference maximum TDOA of each microphone.

Figure-3. GCC TDE simulation in distance 1.

Figure-4. GCC TDE simulation in distance 2.

Figure-5. GCC TDE simulation in distance 3.

Figure-4 shows the result of the simulation in distance 1, Figure-5 in distance 2, and figure 6 in distance 3. The results show that there’re quite differences between the exact sound source location (green circle) and the estimated location (red circle) after using the GCC-PHAT TDE. Even the result of the estimated location wasn’t precise, but still, we can say that the result of the estimation is close to accurate.

CONCLUSIONS

After did the simulation with MATLAB, we can take a conclusion that the accuracy of sound source localization using GCC-PHAT TDE is high, which is 94.86%. The result is taken from the average of several trials that has been done. This result also justifies previous researches that has been done which stated that PHAT is the best weighting function for GCC. For further works, a real time sound source might use for localization instead of using a sample signal audio from the library.

ACKNOWLEDGEMENTS

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES


