Robust Speech Direction Detection for Low Cost Robotics Applications

Samyukta Ramnath
Department of Electrical and Electronics
BITS Pilani K.K. Birla Goa Campus
Goa, 403726
Email: samyuktaramnath@gmail.com

Gerald Schuller
Dept. of Media Technology
Ilmenau University of Technology
Ilmenau, Germany - 98693
Email: shl@idmt.fraunhofer.de

***Abstract*—Previous efforts in sound source localization have extensively studied algorithms for localization and separation with differing kinds of microphone arrays, and with sophisticated separation algorithms with high computational complexity. The basic goal of this study is to implement a system that iteratively changes its direction to move towards the source. Low-complexity is a requirement, since the platform used is a Raspberry Pi. The first objective in this project was to identify the location of a single source. The second objective was to distinguish voice from noise or instrumental music, accomplished using a band-pass filter. Classification using a Support Vector Machine was found to be too slow to be a viable method to run on the Raspberry Pi. The system thus created is a low-cost, low-computational alternative to sound source localization. Future work could consider using more robust, rule-based methods that are computationally viable to run on the Raspberry Pi.**

***Keywords*—Acoustic signal processing, Source separation, Digital filters.**

I. Introduction

The aim of this study was to implement a computationally inexpensive, iterative system to follow a human voice in real time, using low cost electronics and processors. Sound source localization does not require direct line of sight, and can be implemented relatively more easily than vision-based localization methods [7]. Thus, acoustic-based source localization methods can work well in environments with less-than-ideal conditions. Various methods have been used in acoustic sound source localization, including the use of microphone arrays, transfer functions that model the human system of hearing [6], or machine-learning based methods [10]. Implementing a standalone method for direction estimation with low cost electronics would mean using as simple an algorithm as possible, with as few resources as possible, even if it means compromising on precision and accuracy. Enabling the system to follow the voice iteratively, and get more accurate as it moves closer to the sound, would compensate for this lack of precision. Various algorithms were tested on the Raspberry Pi, and the most suitable one was chosen for the task.

Some constraints and points that should be considered while approaching the problem of acoustic source localization with robots are [1]:

- *Echoes and Reverberation:* Reverberation confuses the localization algorithm, as the sound reflected off a surface from a frame of samples could reach the microphone and interfere with the next frame of audio, causing an inaccurate calculation of time delay between each channel. Thus, in very reflective environments, we expect the algorithm to have low performance. Literature mentions the use of reverberation filters [2]; however, in this paper, a simple power comparison has been used, checking whether the power in the left and right channel is comparable, using the fact that there will be some attenuation in the signal after reflection.

- *Noise:* High noise levels reduce the Signal-to-Noise Ratio of the signal, making the algorithm less accurate. Noise suppression algorithms are available in literature, [3], but in this paper, a simple power threshold has been used. The power of each frame has been computed. The minimum power amongst all frames so far has been computed, and this has been assumed as the background noise power level. A ratio of the current power to the background power distinguishes between stray noise and an acoustic event.

- *Source Specificity:* The robot should be able to distinguish between as well as separate different acoustic sources. Existing methods to do so have referenced methods such as Non-negative Matrix Factorization, which is fairly computationally expensive. The approach used in this paper is to have a simple frequency cutoff, assuming that each source occupies a specific part of the spectrum of the frame.

- *Latency:* Latency is defined as the time difference between stimulus and response to stimulus. It is essential for the system to have a low latency for our application, else the algorithm would take too long and thus be too discontinuous to be of any practical use.

- *Computational Expense:* The amount of computational expense acceptable in the system depends on the platform used. For a robot, limited hardware capabilities mean that algorithms should try and reduce the computations performed. This is particularly true on the Raspberry Pi, which has limited real-time computational capabilities. There is a trade-off between computational power and accuracy.

II. Previous Efforts

Previous approaches to sound source localization have extensively studied algorithms for localization and separation, both on platforms with vast computational resources as well as those with limited computational power [4]. Some approaches have taken inspiration from the human hearing system, which is binaural and uses pinnae to give additional cues about direction [5]. Further approaches have studied the effect of the shape of the human head on localization, and have attempted localization with a humanoid shape [6]. Some literature has gone beyond binaural source localization, and have used multiple microphones to localize a source in 3 dimensions with high accuracy [7]. Source localization in one plane is enough for most ground robots found in literature. Two microphones can only localize a sound in one plane, with an inherent ambiguity present in whether the sound is coming from the front or from behind the robot. This can be removed either by using three non-aligned microphones [8], or by using two microphones, listening and finding an angle, rotating the robot by some angle, listening again and finding the correct direction of the sound in 360^o. One attempt to take inspiration from the binaural human system mentions this method, to resolve the ambiguity between the front and back position of sound [9]. This approach, however, used a robot with a higher processing power than the Raspberry Pi, and does not attempt source separation. One very unique approach to this problem was to use a single microphone and a pinna-like structure to learn the direction of sound in 3 dimensions [10]. Some efforts involving auditory signal processing go beyond the simple task of localization and look at predicting human responses to auditory events. The 'Two!Ears' project at TU Berlin notes that 'while many models that mimic the signal processing involved in human visual and auditory processing have been proposed, these models cannot predict the experience and reactions of human users' [11].

The methods described so far have used omnidirectional microphones (microphones which record the same signal in all spatial directions). Literature extensively describes an algorithm called Generalized Cross-Correlation (GCC) in which the delay estimate is obtained as the time-lag which maximizes the cross-correlation between filtered versions of the received signals [12]. Another method is the Steered Response Power (SRP), which is a function generally used to aim a beamformer. The beamformer acoustically focuses the array to a particular position or direction in space [13]. The SRP algorithm has been shown to be more accurate than the GCC algorithm, but often at a high computational expense (sec. 8.1, [13]).

III. Methodology

A. TDOA Estimation

The direction of the sound source was estimated using the InterAural Time Difference (ITD) method, as opposed to the InterAural Level Difference (ILD) method. This is accomplished using time difference between the left and right channel of a pair of microphones mounted on a vacuum cleaner robot (Roomba). Cross-correlation of the signals in the left and right microphones is used to calculate the time delay of the arrival of signal at the left and the right microphones.

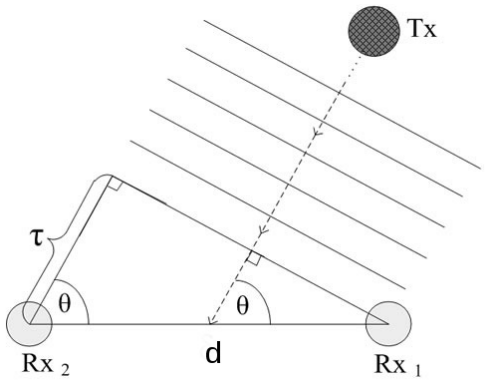

Fig. 1. Diagram to calculate angle of arrival of acoustic source

The Cross-correlations r_{xy} of two signals x and y of length N is expressed as (Section 2.6, [14]):

$$r_{xy}(l) = \sum_{n=l}^{N-|k|-1} x[n]y[n-l] \tag{1}$$

where $k = 0$ for $l \geq 0$ $k = l$ for $l \leq 0$

The peak in the cross-correlation signal indicates the index at which the two signals are similar. Using the sampling rate of the signal (fs), the time difference can be computed as :

$$t = \frac{ceil(\frac{2N-1}{2}) - \arg\max(r_{xy})}{fs} \tag{2}$$

where t is the computed delay in seconds, and $ceil(x)$ is a function which returns the smallest integer greater than or equal to x. The output signal r_{xy} is of length $2N-1$.
$\arg\max()$ are the points of the domain (arguments) of some function at which the function values are maximized.

B. Angle Finding

In order to compute the angle of arrival from the time delay of arrival of the signal between the left and right channel of the stereo microphone pair, an assumption is made that the distance between the source and the system is much more than the distance between the left and right microphones. Thus, the angle made by the source at both the microphones is approximately the same. The angle of arrival can thus be computed by using the time delay with the help of fig 2, as :

$$\cos\theta = \frac{\tau \cdot c}{d} \tag{3}$$

Since only the angle of arrival of sound is known, and not the plane in which the source is, the source could be localized anywhere on a cone with an aperture angle equal to θ. This is called the 'cone of confusion' [15]. For 3 dimensional localization applications, at least 4 microphones would need to be arranged in a tetrahedron, to get measurements from at least 3 pairs of microphones. Once three angles are computed, three cones of confusion are obtained, and the intersection of these three surfaces will indicate the actual direction of the source. This method cannot be used to determine the distance of the source from the microphone.

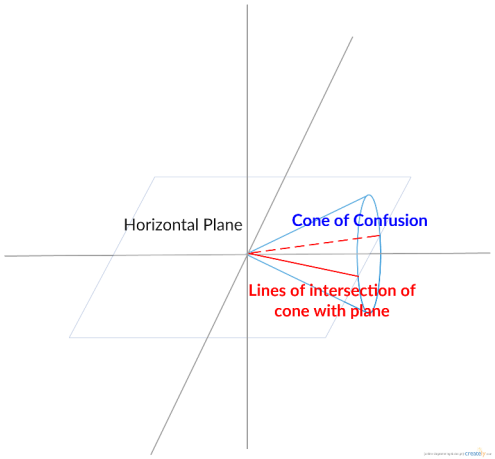

Fig. 2. Intersection of cone of confusion with the horizontal plane

In our application, the robot can only move in the 2 dimensional horizontal plane, and so the only relevant angle would be in the plane parallel to the ground. Thus, the possible direction of the source can be found by finding the intersection of the cone of confusion of the two microphones with the horizontal plane.

The intersection of the cone with the horizontal plane gives two possible directions of arrival of the audio source : one 'in front' of the Roomba robot, and one 'behind' it. Thus, the cross-correlation method can only localize the sound in 180^0; in order to localize the sound in a plane in 360^0, the entire system is rotated by 5^0 once after taking an initial reading of the time difference. This allows the system to determine whether the sound source is behind it, or in front, which allows it to determine the angle of arrival of the source in the horizontal plane. This takes inspiration from the way humans localize sound.

C. Frequency Limitations

If the acoustic source is a single pulse, finding the time difference at different microphones is a simple task. However, if the acoustic source is a signal which is continuous in time, such as a cosine of fixed frequency, then we need a phase difference of at most π between the two signals, in order to determine which signal is leading the other. To satisfy this condition, the distance d between the microphones should be less than half of the wavelength of the incoming sound, so that (eqn. 4.13, [17])

$$d \leq \lambda/2 \tag{4}$$

With a spacing of about 15 cm between our ears, humans can use Inter-Aural Phase Difference to localize sound up to a frequency of about 1kHz. For frequencies of incoming sound greater than that, humans use Inter-Aural Level Differences to localize sound. In the experiment, the microphones were placed at a distance of about 8cm, which allows for a higher frequency cutoff of about 2kHz.

D. Voice-Music Discrimination

The task of voice-music discrimination was done using a spectral flatness measure and a low-pass filter on the individual channels of the stereo microphone pair.

Music has its spectral energy concentrated at certain particular frequencies, whereas white noise has it's spectral energy spread over most of the spectrum. The Spectral Flatness Coefficient computed over the frequency spectrum for each window indicates the 'flatness' of the spectrum, thus serving as a useful way to distinguish between tonal sounds (voice or music) and atonal sounds (noise).

Consider a signal x of length N samples, framed into m segments of size N/m.
The Spectral Flatness Measure (SFM) is then defined as: [18]

$$SFM = log_{10}[\frac{AM(X(m))}{GM(X(m))}] \tag{5}$$

for the mth frame of the signal.

IV. Our New Approach

Voice-Music discrimination was done using a frequency cutoff, in the form of an elliptic low-pass filter, with a 60 dB stopband attenuation and a frequency cutoff of 300 Hz. This frequency was chosen because it is close to the fundamental frequency of the human female voice [16]. It was assumed that the music used would be of a much higher frequency than 300 Hz, and of a much higher frequency than the human voice. Instruments like the higher notes of a violin, a piccolo or a flute would satisfy this assumption. Thus, the high frequency music was filtered out, and the direction of the voice was estimated.

A. Filtering

Using a frequency cutoff near the fundamental frequency of the human voice (usually below 300 Hz for a human female voice) is a low-complexity approach to the problem of separating voice from mixed speech-music signals. The method would work if the same filter is applied to the left and right channels of the microphone. If voice and music are playing simultaneously, voice would be dominant in the low frequency region although there will be some spectral content from the music signal. Thus, the Robot would take the dominant signal as the voice signal, compute the angle and move towards the voice. When music is playing without any voice, there will still be some signal content in the low frequency region. Thus, the Roomba robot will pick up this signal in the low frequency region, compute the angle of arrival and move towards it. The computation of the angle should not be affected by the application of the low-pass filters, if exactly the same filter is applied to both channels. Low-pass filtering was found to be a reasonably accurate, low-complexity solution for speech-music separation.

The elliptic filter has a higher ripple than the butterworth filter, but provides the minimum required attenuation in the stopband and maximum admissible attenuation in the passband at a lower order than the butterworth filter. This makes it more suitable for real-time applications. (Section 7.6, [19]).

The elliptic filter used in the project was a fourth order bandpass digital filter, with a lower cutoff of 65 Hz and a higher cutoff of 350 Hz. It has a passband attenuation of 5 dB and a stopband attenuation of 60 dB. It was realized with

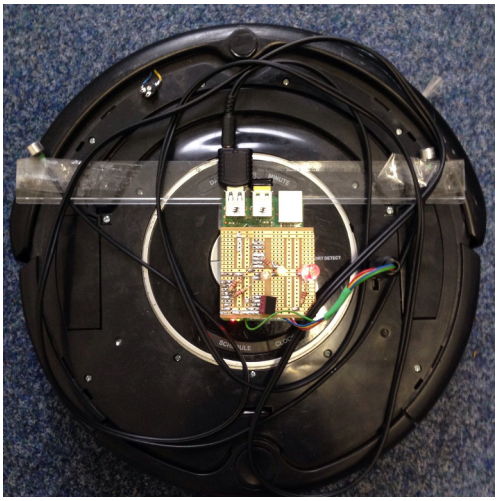

Fig. 3. Experimental Setup

the 'ellip' filter from the SciPy package from Python. The parameters were chosen according to the background noise conditions and frequencies of voices and music in the place of testing.

B. Experimental Setup

Two separate microphones were connected as a stereo pair, via a USB sound card, to a Raspberry Pi. The RPi was placed atop a Roomba robot. The RPi was connected to the Roomba robot via a serial interface board [20]. The setup is shown in fig 3.

Python's PyAudio module was used to record audio from the microphone pair. Once the PyAudio object was instantiated and the stream was opened, a frame of 1024 samples was acquired and processed. Once the frame was acquired, the stream was closed, and the audio was separated into left and right channels by separating odd and even samples. An identical digital elliptic filter, as described in IV-A, was applied to the left and right channels of audio samples. Cross-correlation of the two signals gave an estimate of the time-delay between the left and right channel signals. Using the time delay between the left and right channels, the angle of arrival was computed. Once the angle was found, the Robot was turned anticlockwise by 5^0 and the audio stream was opened once again. Another frame of data was acquired, and once again, the angle was computed. Based on whether the angle increased or decreased, the Robot turned in the clockwise or anticlockwise direction toward the source, and moved forward by 150 cm. The process repeated until the program was stopped.

Source code for the project can be found on GitHub, [21], along with a brief explanation of each code.

V. Results

The system was tested with an instrumental version of a song on the flute playing from a device kept at one end of the room on the ground, and a singing voice at the other end of the room. It was observed that the robot would iteratively move towards the singing voice when both the instrumental music and the singing voice were played. When the singing voice was paused, the robot would move towards the instrumental music.

The Roomba was found to turn in the direction of the voice, with noise and music playing from elsewhere, when the voice was above a certain volume level and within some distance from the Roomba. If only music was played, the Roomba moved towards the music. The accuracy of localization increased as the Roomba got closer to the source - such that the Roomba eventually reached the sound source, effectively following it. These tests were recorded and can be found on the TU Ilmenau website [22] :

- Roomba iRobot responding to foot-tapping on the ground
- Roomba moving towards voice despite music playing in the background

VI. Conclusion

This paper aimed to investigate the applications of acoustic source localization and source separation in robot control, and implement a system for source localization on the Raspberry Pi, a processor with limited computational power. The aim was to take an approach that would have the least complexity in terms of number of elements and complexity of algorithms. In this aspect, the binaural microphone system implemented on the Roomba iRobot with the Raspberry Pi satisfies the goal of the study. The biologically-inspired method of using two microphones accompanied by a turning movement enabled us to localize the sound source in 360^o without having to use more than two microphones, which avoided the trouble of synchronizing multiple microphones. The solution of using a low-pass filter on each frame of data instead of a complex source separation algorithm enabled the system to run smoothly and with an acceptable latency of a few seconds. The resultant system iteratively changes its direction and moves toward the source, and becomes more accurate as it gets closer to the source. The system works with less accuracy in a noisy,reverberative environment, but the robot still reaches the source, albeit after a larger number of iterations. Due to the final goal of an iterative system, it is not needed to know the distance of the source from the robot. The system can also distinguish between a low frequency tonal sound such as voice and a higher frequency tonal sound such as instrumental flute music.

The current system still has a number of limitations. It cannot distinguish between two tonal sounds which share the same or mostly similar space in the frequency spectrum. Although it does work in a noisy environment, it is still fairly susceptible to noise and reverberations. Future work in this direction could look closer into developing more sophisticated algorithms with low computational complexity for source-separation. Noise suppression algorithms and reverberation filters could be included to improve the accuracy of the system.

Acknowledgment

This work was done during an internship at the Ilmenau University of Technology, Germany, at the Group for Applied Media Technology.

REFERENCES

[1] M. Durkovic. "Localization, Tracking, and Separation of Sound Sources for Cognitive Robots". Diss. Universitatsbibliothek der TU Munchen, 2012.

[2] T. Yoshioka et al, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition." *IEEE Signal Processing Magazine* 29.6 (2012): 114-126.

[3] R. Bentler and L.K. Chiou, "Digital noise reduction: An overview." *Trends in Amplification* 10.2 (2006): 67-82.

[4] S. Ramnath, "Stereo Voice Detection and Direction Estimation in Background Music or Noise for Robot Control". Undergraduate Thesis. BITS Pilani, K.K. Birla Goa Campus, 2016.

[5] P. Hofman and A. Van Opstal, "Binaural weighting of pinna cues in human sound localization." *Experimental brain research.* 148.4 (2003): 458-470.

[6] G. Athanasopoulos, H. Brouckxon and W. Verhelst, "Sound source localization for real-world humanoid robots." *Proceedings of the 11th international conference on Signal Processing* Vol. 12. 2012.

[7] J-M. Valin et al., "Robust sound source localization using a microphone array on a mobile robot," Intelligent Robots and Systems, 2003.(IROS 2003). *Proceedings. 2003 IEEE/RSJ International Conference on.* Vol. 2. IEEE, 2003.

[8] H-Y. Gu and S-S. Yang, "A sound-source localization system using three-microphone array and crosspower spectrum phase," *2012 International Conference on Machine Learning and Cybernetics.* Vol. 5. IEEE, 2012.

[9] J.C. Murray, H. Erwin and S. Wermter, "Robotics sound-source localization and tracking using interaural time difference and cross-correlation," *AI Workshop on NeuroBotics.* 2004.

[10] A. Saxena and A.Y. Ng, "Learning sound location from a single microphone," *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE,2009.

[11] Two!Ears Team. (2017). Two!Ears Auditory Model 1.4. Doi: 10.5281/zenodo.238761

[12] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.4 (1976): 320-327.

[13] J.H. DiBiase, "A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays," Diss. Brown University, 2000.

[14] J.G. Proakis and D.G. Manolakis, "Digital Signal Processing," 1st ed. Upper Saddle River, N.J.: Prentice Hall, 1996. Print.

[15] V. Pulkki and T. Hirvonen, "Localization of virtual sources in multi-channel audio reproduction," *IEEE Transactions on Speech and Audio Processing* 13.1 (2005): 105-19. Web.

[16] R.J. Baken and R.F. Orlikoff, "Clinical measurement of speech and voice," *Cengage Learning,* 2000.

[17] C.F. Scola, "Direction of arrival estimationA two microphones approach," Diss. Blekinge Institute of Technology, 2010.

[18] Y.Ma and A. Nishihara, "Efficient Voice Activity Detection Algorithm Using Long-Term Spectral Flatness Measure," *EURASIP Journal on Audio, Speech, and Music Processing* 2013.1 (2013): 21. Web.

[19] A.V. Oppenheim and R.W. Schafer, "Digital Signal Processing," 1st ed. Englewood Cliffs, N.J.: Prentice-Hall, 1975. Print.

[20] G. Schuller, "Raspberry Pi Serial Interface," *Raspberry Pi Serial Interface.* N.p., n.d. Web. 01 Dec. 2016. *https://www.dk0tu.de/blog/2016/06/25_Raspberry_Pi_Serial_Interface/*

[21] S. Ramnath, "Hale2bopp/Audio-Source-Localization." GitHub. N.p., 20 Feb. 2017. Web. 04 Mar. 2017. *https://github.com/hale2bopp/Audio-Source-Localization*.

[22] "TU Ilmenau (Homelink)." Forschung am Fachgebiet Angewandte Mediensysteme. N.p., n.d. Web. 04 Mar. 2017. *https://www.tu-ilmenau.de/index.php?id=44974*.