Beamforming using TDOA and Microsoft Kinect

Sajjadali Hemani
shemani@kth.se

Rickard Eriksson
rieri@kth.se

Abstract

The receptive part of beamforming is investigated in this report using two methods. The first one puts the Kinect software and hardware under the microscope to test the capabilities of the microphone array while the other one is implemented using a self-written Matlab algorithm. The experiment was carried out in a controlled environment with a degree of human and systematic errors taken into consideration. .Wav files were recorded, used as inputs to the program and by making use of time difference of arrival algorithms, a position and angle could be deduced.

1 Introduction

This report investigates the Kinect's ability to correctly find the source of sound. This was done by comparing two different implementations which made use of the Kinect. The first method was to make use of Kinect's inbuilt software provided by Microsoft which displayed the beam and source angle of the source. The other method was to record the sound from each microphone and deduce from that, using an algorithm, the angle at which the sound came from.

Among many sources, Microsoft's own support website for their Kinect SDK was very informative as it contained detailed description of how their programs were made[1].

1.1 Background

The main concept that this report will revolve around is beamforming. Beamforming is a signal processing technique to find the source or target in order to direct the receiver or transmitter. The desired result is to increase the effectiveness with which the signal is transmitted or received. In this report we will primarily deal with the receptive nature of this concept however it is worth mentioning its value on both ends.

Beamforming on the receptive end works by using differences in phase or time when receiving the signal to calculate the source of the signal[3]. In terms of sound the time at which the signal is received is of vital importance as this is used to calculate the delay between the difference microphones. This information is then used to introduce a delay to properly synchronize the sound. To achieve this, an array of microphones is needed[2]. When the angle of the sound has been set the receiver produces a 'beam' which represents an area with confidence that the device will amplify. The more confidence the receiver has of the sound coming from a certain direction, the more narrow the beam can be formed and the signal can be further amplified.

1.2 Kinect

To be able to experiment with beamforming, the Kinect was made use of. The Kinect is a motion sensing device by Microsoft equipped with a webcam and an array of microphones[5]. The camera can be used to track humans through the 'skeletal stream' which calculates the joints of the human body and can give coordinates of the user. For the purpose of this experiment, only the microphones will be considered. The version used for the experiment was 1.8. From the SDK webpage[1], the maximum and minimum angle were specified at -50°to 50°which was also confirmed experimentally. The location of the microphones and the measurements are shown in figure 1.

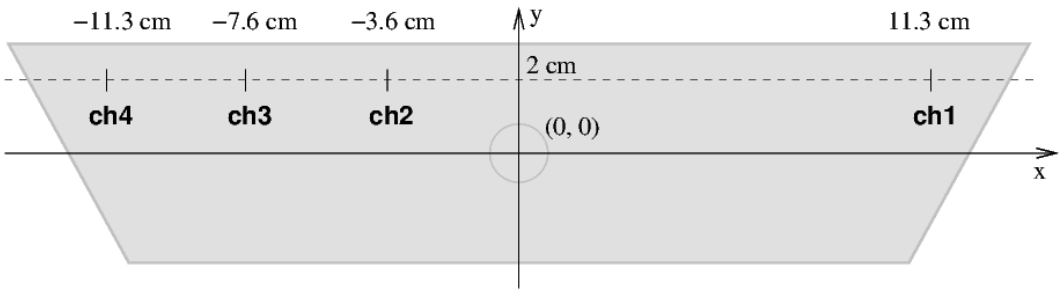

Figure 1: Geometry of the Kinect from a topological point of view[7].

2 Method

There are many methods that can be used for locating the source of a sound such as sonar (usually used under water), biologic echo location (bats and dolphins use this method)[4] and TDOA (Time Difference of Arrival). For this report the TDOA algorithm was the most favoured method to use. This algorithm was preferred over other algorithms because of its' efficiency and accuracy. The TDOA algorithm is used in many organic lifeforms to position a sound source, for example us humans use with our ears and is called interaural time difference[4] and can be visualized in figure 2.

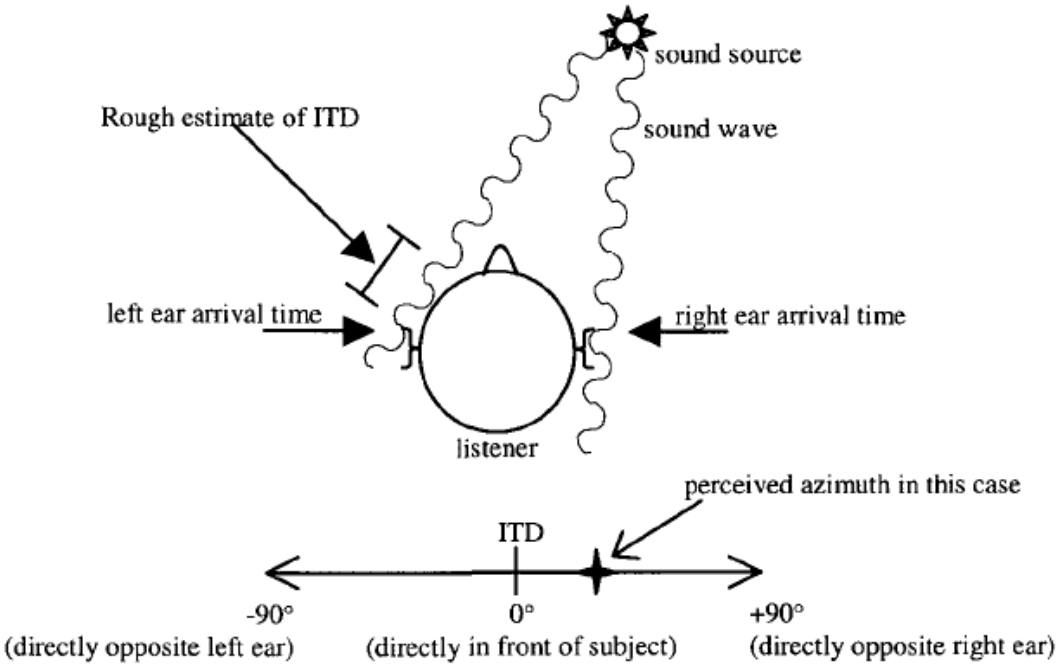

Figure 2: Image showing the principle of interaural time difference [6].

By using two omnidirectional microphones or receivers the general direction of a sound can be located, thanks to cross-correlation[9]. The sound is received by the two different receivers have a slight time shift between them. The soundwaves recorded from the two receivers are cross-correlated to confirm how much the signal is delayed between them. When the time shift has been calculated it can be transferred into an angle by trigonometry. The formula[9] for cross-correlation can be described as in equation (1).

$$Cross_{x_1,x_2}(\tau) = \sum_{k=0}^{n} x_1(k)x_2(k-\tau) \tag{1}$$

In equation (1) x_1 and x_2 are the arrays containing the signals that are being investigated. The variable τ represents the difference in time as we explore a range of different time stamps.

By summing the τ for each of the samples we get a vector representing the different correlation points. After this is done the highest value in the vector is the best assumption for the angle of the sound. When the best τ is found we can use equation (2) to get the angle of the sound source.

$$\theta = arcsin(\frac{\Delta t \times v}{x}) \tag{2}$$

In equation (2) the angle θ represents the angle from where the sound is coming from.

Considering all different time delays from negative infinity to positive infinity is not computationally feasible. We assume that the medium the sound is traveling through is air and we know the distance between the two microphones is D. By using the speed of sound v we can calculate the longest delay in time between the microphones and sort out most of the range of τ. Before this can be used it has to be converted from a delay in time to a delay in number of samples as the algorithm works with samples rather than time.

$$\tau_{max}(time) = \frac{D}{v} \tag{3}$$

$$\tau_{max}(samples) = \tau_{max}(time) \times frequency \tag{4}$$

Using equations (3) and (4) the range of τ is now from $-\tau_{max} \leq \tau \leq \tau_{max}$ (it extends to $-\tau_{max}$ because the it is unknown which signal of the two that is received first).

For each of the samples in the signals the algorithm compare the signals in the range specified by τ. These are summed up and compared to each other where the winner is considered the time shift of the correlation and can be converted into an angle or radians. The following code runs through all samples and sums up all the correlation variables τ by following the equations (1) and (2). At the end of the code the angle θ in calculated using equation (2).

```
% loop through the signal samples
for n = ( 1 + ITDmax ) : ( length(leftMic) - ITDmax )
    % loop through all possible ITD values
    for Tau = -ITDmax : ITDmax
        % calculate the cross-correlation between signal leftMic ...
            and rightMic
        cross(Tau + ITDmax + 1) = leftMic(n) * rightMic(n - Tau);
    end
    % sum the cross-correlations with earlier samples
    sumCross = sumCross + cross;
end

% find the index 'I' at which the correlation function has a maximum
[¬, I] = max(sumCross);
% compute the corresponding maximum in samples
I = I - ITDmax - 1;
% compute radians
theta = asin(soundspeed * I / (Mdistance * freq));
```

The TDOA and its cross-correlation is useful for detecting from what angle the sound is coming from, but to decide the exact x and z position of the source in the 2D plane more information is needed. To calculate the coordinates of the sound source we need at least 3 microphones. In this case the channels 1, 2 and 4 from figure 3 can be used. By using the TDOA algorithm to decide the angle of the source with the two combinations *channel 1 - channel 2* and *channel 2 - channel 4* two different angles and vectors can be found. The vectors can be translated into linear equations. With these two lines the intersection which represents the position of the sound source is found. By calculating the euclidean distance from the sound source to the origin of *channel 2* (the center microphone) the approximate distance from the sound source can be found.

3 Experiment

Experimenting with sound has innate sources of error as background noise will always be a factor. Therefore it was of vital importance to keep the environment as controlled as possible. The room was chosen to be open in order to be able to move the sound source freely. The Kinect was placed on a table 90 cm from the ground and the sound source was placed 1m away. The source was to be moved around in a 180° fashion in order to explore the extent to which the Kinect could detect sound sources. To make sure the source was constantly at 1m from the Kinect a half-circle was made with the Kinect at the center. The angles at which the source should be measured from were marked

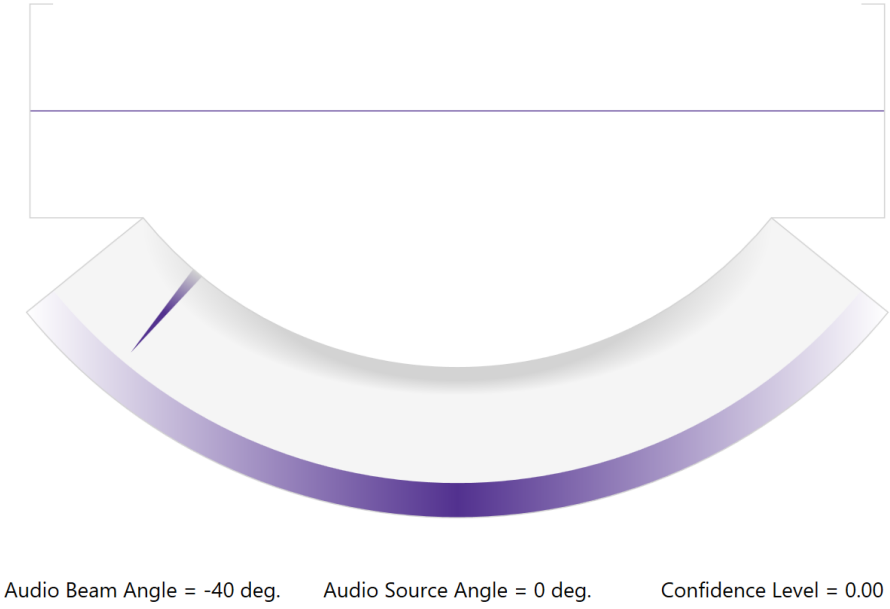

Figure 3: Audio basics-WBF program

out. As mentioned earlier the angle was calculated in two different ways. The Kinect program from Microsoft was first made use of by simply turning it on, emitting a sound and recording the source angle. Figure 3 shows a beam and source angle. The source angle is the program's perception of the angle at which the source may be whereas the beam angle is the direction of the receptive beam formed by the microphones. The closer the two angles get, the closer the confidence level gets to 1.

The other method was to separately record the sound that each microphone picks up. To do this, audacity was made use of. Audacity has the option to extract the .wav[8] file with the sound that each microphone picks up. After the sound had been recorded it was analyzed using the Matlab program to locate the source.

4 Results

The angles found with the TDOA algorithm using only channel 1 and 4 can be seen in table 1. In this table the real angles are compared to the Matlab algorithm using TDOA and the algorithm of the kinect.

The second part of the experiment was calculating the exact position in 2D space using the intersection of the two pairs of channels. The result of the algorithm can be seen in table 2 where the distance from origo is shown along the angle of the sound source. As another control variable the angle from the assumed position of the sound source and the y-axis was calculated and can be seen in table 2 as well.

The vectors and intersection can be seen in figure 4 which combined with table 2 gives the position of the sound source. The distances found in figure 4 are in meters.

Table 1: Angles found with Matlab and Kinect

Measured angle(°)	Kinect software angle(°)	Matlab algorithm angle(°)
-90	-45	-54.9583
-45	-27	-33.0809
0	6	0
45	39	33.0809
90	—	54.9583

Table 2: Position of sound source from 3 microphones

Measured angle(°)	Matlab algorithm angle(°)	Matlab algorithm distance(m)
-90	-49.8662	0.16658
-45	-30.7965	0.83577
0	0	1
45	30.7677	1.256
90	51.7733	0.62405

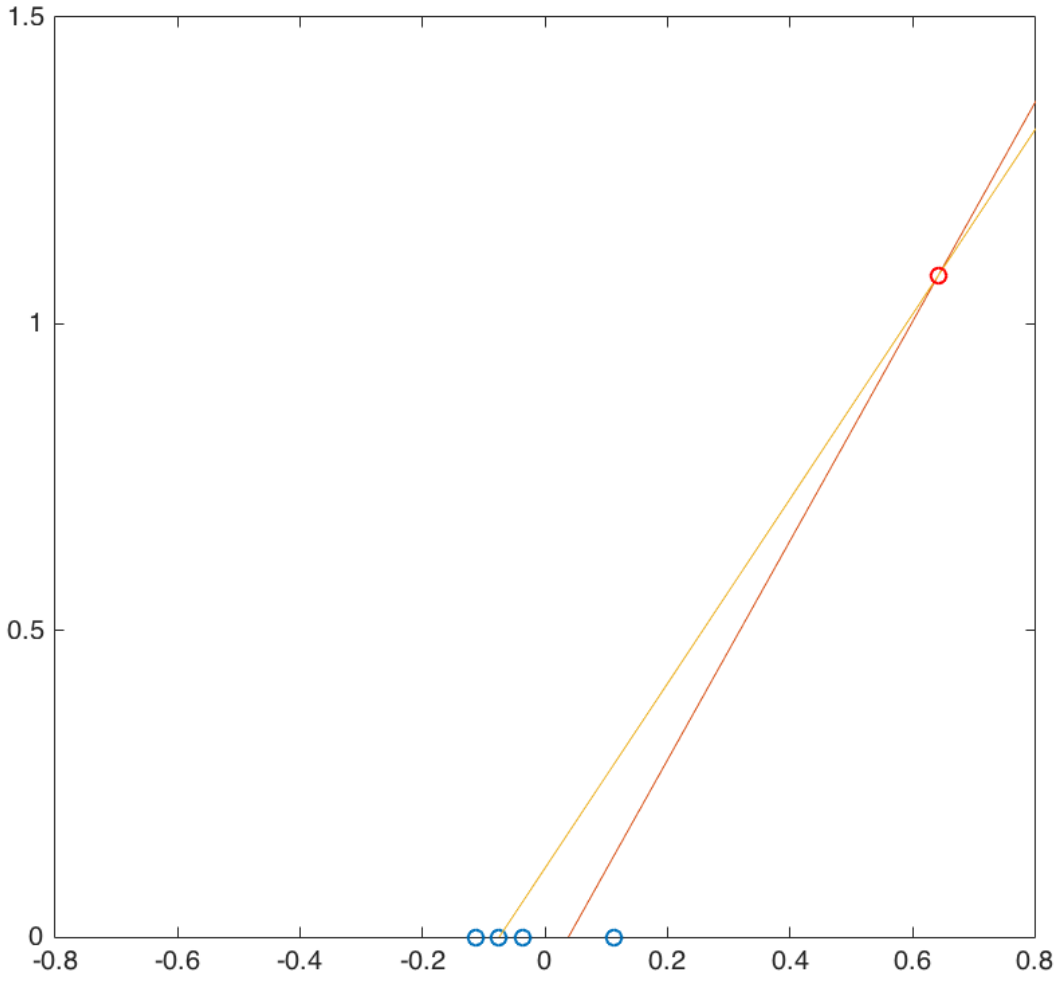

Figure 4: The vectors from the channel pairs 1-2 and 2-4 as well as the intersection between them shown as a red circle.

5 Discussion & Conclusion

5.1 Evaluating the results

While the experiments has inherent flaws and weaknesses which are further discussed below, it did yield some valuable results. Looking back at the original thesis of this report which was to compare two different methods for achieving the angle of the sound source, it has been reached to an extent. Table 1 compares the results from the different implementations which shows that the Matlab algorithm performs better and is more consistent in terms of symmetry. The second part of the result is perhaps more interesting and unique which was to be able to predict the approximate distance to the source. The results from table 2 show that the distance prediction is accurate within the angle range of -50°to 50°. This went beyond what the Kinect offered from it's software and from reading articles related to Beamforming, a clear absence of this conclusion is found which makes it unique as well.

The biggest remark to the result is in the external validity as the test set and experiment setup has obvious weaknesses.

5.2 Improvements to the experiment

As has been mentioned throughout the description of this experiment, there have been many sources of error and areas of improvement. One very obvious source is the human error involved in all the measurements. The measured angles which were held as reference points for determining the correctness of the data, were done using a protractor. This introduces a systematic error stemming from the instrument on top of the human error. Similarly the distance from the Kinect was measured using a folding ruler which also falls into the category of systematic errors.

An improvement that is more related to the interpretation of the experiment is the number of measurements taken. The external validity of this experiment is put into question with so few angles which is a major flaw in this report.

From a technical stand point there is an upgrade that could massively improve the quality of the results. Instead of separately recording the .wav files a live broadcast of the 4 channels would be a clear upgrade. This way the experiment will be a lot smoother to perform and the external validity of the overall experiment will be increased.

5.3 Future work

Sound localization is not only limited to a 2D space, with some extensions the algorithm could possibly calculate the 3D coordinates of the sound source. By using an extra two microphones, the 3D coordinates can be found. The only thing that has to be changed in the algorithm is to use the calculation to find out the angles of the vectors between the two new microphones and the middle microphone of the ones used in this report. After the vectors have been calculated the intersection of the four vectors can be extracted into 3D coordinates. The microphones could be arranged in the shape of a plus sign to calculate the angles. An example of a possible configuration of the five microphones can be seen in figure 5. The microphones are evenly spaced with the distance λ for easier calculation.

The configuration of the microphones are based on the same principle as the configuration in this report where microphones 1-2-3 were used to extract the coordinates. Microphone 1-2-3 are used to calculate the azimuth and transfer it to x and z coordinates while microphones 4-2-5 calculates the elevation and transfers it to x and y coordinates. This configuration is just one of many ways to achieve 3D localization. To find the most accurate one would have to experiment and change the different distances and positions of the microphones.

The kinect could extract the 3D coordinates from the skeletal stream (the camera that locates joints) by following the head of the user. This could be useful as all other sound in 3D space could be filtered away to pick up the sound from the user even better. The skeletal stream could also be used with the algorithm in this report to make the beamforming even more accurate.

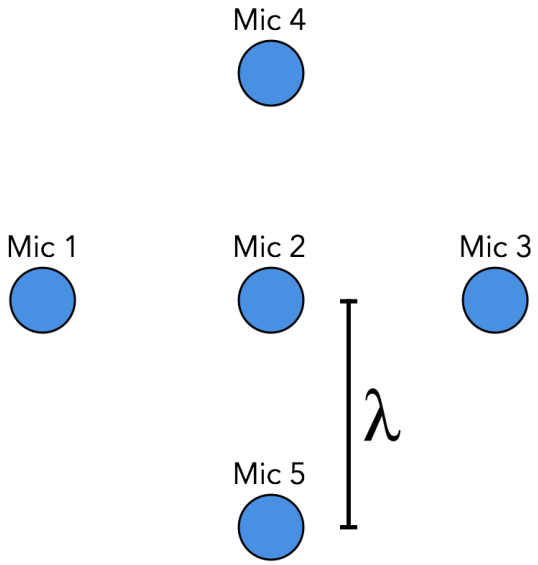

Figure 5: Five microphones seen from the front configured in a plus shape with distance λ from the center mic for 3D localization.

References

[1] Microsoft: Using the Kinect as an Audio Device. https://msdn.microsoft.com/en-us/library/jj883682.aspx.

[2] Y. Li, T. Banerjee, M. Popescu and M. Scubic. Improvement of acoustic fall detection using Kinect depth sensing. . *Preprint submitted to 35th Annual International IEEE EMBS Conference*, January 2013.

[3] Wikipedia: Beamforming. https://en.wikipedia.org/wiki/Beamforming.

[4] Wikipedia: Sound localization. https://en.wikipedia.org/wiki/Sound_localization.

[5] Microsoft:Kinect for Windows Sensor Components and Specifications. https://msdn.microsoft.com/en-us/library/jj131033.aspx.

[6] Minh Nguyen: The 3-D Sound Effect. https://cnx.org/contents/AOK1OZK3@2/Background.

[7] Kinect array geometry topological picture. http://giampierosalvi.blogspot.se/2013/12/ms-kinect-microphone-array-geometry.html

[8] Audacity: Audio track. http://manual.audacityteam.org/man/audio_tracks.html

[9] Lauren Calmes: Cross-correlation. http://www.laurentcalmes.lu/