Beamforming with Kinect

Zhoujie Fang, JerryFan, XuXiao Ma, and Muhammad Haky Rufianto

(Double Project)

DT2118 Speech and Speaker Recognition Project Report

May 18, 2015

Abstract

The audio stream is one of the most important features for the Microsoft Kinect. A good sound capture brings a lot of benefits such as the better communication, the better playback and record, and the better speech recognition for the voice controls. Beamforming is one of the solutions for improving the sound capture quality. For this project, we performed two ways to test the beamforming for Kinect. The first way is to compare the result with different recording methods to analyze how beamforming affects the speech recognition accuracy. The second way is to compare the feature of noise reduction when using beamforming capability when receiving the sound.

Base on the experiment, we can conclude that beamforming technology used by Kinect could increase the recognition ability of the ASR at a certain rate. With beamforming, the audio sound will have a better noise reduction and better focus on certain sound from a specific direction.

1. Introduction

Since the ability to speak and communicate comes naturally to humans, speech technologies are evolving to into a standard way to communicate between humans and machine, thus these technologies are impacting in people's life. A fast growth of Speech technologies can be seen during last decade with the development of speech recognizing system, remote accessing, dictation and many others. However, speech has demonstrated that it is not always an efficient mechanism to interact because of certain limitation when creating the interaction system between human and machine. One of the best way method to promote the speech technology is to increase the accuracy and performance of speech recognizing in order to make machine more agile and could use the speech as

another modality as the input to the system.

The beamforming technology is one of many development that has a main goal to create a better sound recording to get the voice as the source of the input data. Beamforming is the concept to determine the relevant audio source by analyzing the sound stream from multiple array of microphone, with this method, it can increase the particular audio stream while suppresses the other audio sources, and in the end we would get an input audio which reduces the noise of an audio signal [1].

The aim of this project is to create an experiment using Kinect devices and test the beamforming capability of the devices to gather some of the data and make some measurement about the performance of the ASR system using beamforming technology. In this project, we compare the result of the recording using Kinect devices regarding the noise suppression capability and the ASR performance. We will also mention the tools we use when conducting the experiment. Finally, we will demonstrate how to utilize the beamforming with Kinect devices and show the increase of performance when using beamforming technology.

2. Background

2.1 Automatic Speech Recognition

The automatic speech recognition can be defined as the independent, computer-driven transcription of spoken language into readable text in real time. It's basically a speech technology allowing computers to identify the speech that humans speak into a microphone or telephone.

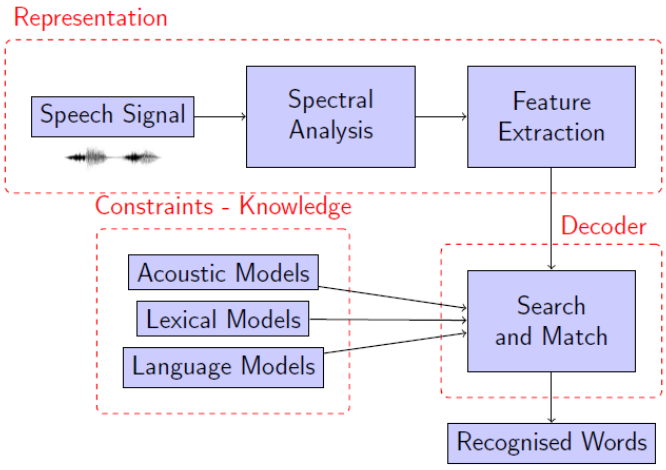

Fig 1. Component of ASR System [2]

The ideal ASR system is to allow the computers recognize the speech in real time, with 100% accuracy. However, there are always a bunch of issues reduce the accuracy. Except the traditional issues such as limited vocabularies or speaker characteristics, the most important issue is the noise. The Robustness of an ASR system refers to the degree of the system can function correctly in the presence of invalid inputs or stressful environmental conditions. Normally the natural environments have several conflicting sound sources come from different direction since the commonly used microphones are all omnidirectional microphones. So it's very important to reduce noise and reverberation in order to improve the word error rate.

One proposed way of doing the noise suppression is the beamforming which is mentions in following section.

2.2 Kinect Beamforming Algorithm

Simply to say, Beamforming refers to the technique that using a sensor array (microphones) to collect the signals with time delay estimates (TDE) based methods which uses the fact that sound reaches the microphones with slightly different times. The distance between each microphone gives the delay and the decay, where the signal decay due to energy losses in the air is actually negligible for the working distance [3].

Beamforming, also as known as spatial filtering brings the result that the signal from desired direction is reinforced and signals from other directions are attenuated. Therefore, the relevant signals are amplified, and the noise is suppressed.

Beamforming techniques can be broadly divided into two categories: conventional beamformers and adaptive beamformers. The conventional beamformers use a fixed set of weightings and time-delays to combine the signals from the sensors in the array, it primarily uses only information about the location of the sensors in space and the wave directions of interest where the adaptive beamformers combine the information with the properties of the signals actually received by the array. It is more effective for the rejection of unwanted signals from other directions.

The Kinect beamforming algorithm considers the effects of the electronic noise and it is actually close to the beamforming algorithm designed by Ivan

Tashev and Henrique S. Malvar in 2005. From the report “A NEW BEAMFORMER DESIGN ALGORITHM FOR MICROPHONE ARRAYS” [4], it shows the captured signal contains two sources of noise, one is isotropic acoustic noise and the other one is instrumental noise.

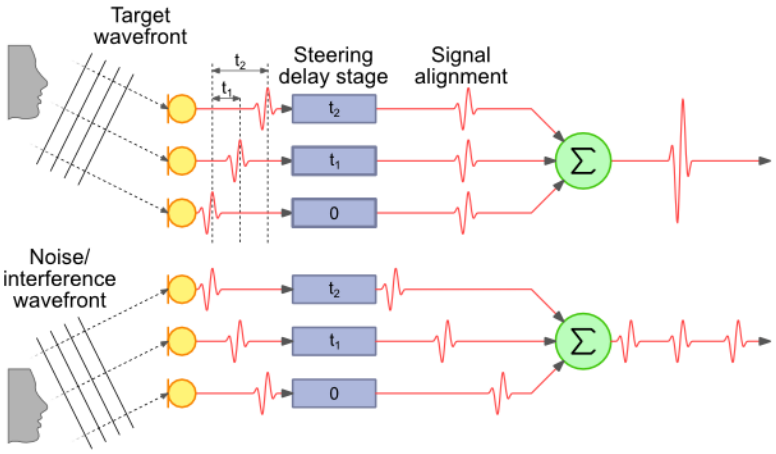

Fig 2. Beamforming workflow [5]

Technically, the canonical form of the time invariant beamformer in frequency domain is just a weighted sum. For each weight matrix it has the corresponding shape of the beam. The goal of the beamforming algorithm is to find the optimal weights matrix for giving geometry and beam direction. Then for each frequency bin find weights to minimize the total noise in the output. Also, a set of constrains is imposed for this solution which is equalized gain and zero phase shift for signals coming from the beam direction.

Although the Beamforming Algorithm is designed for arbitrary microphone array geometry (Either linear or circular with 4-8 microphones), but the Kinect uses four-microphone linear array which will be mentioned in following section.

2.3 Kinect Audio Sensors

For this project we used the Kinect for windows version 1.8. It has several sensors, including RGB camera, infrared (IR) emitter, IR depth sensor, and the multi-array microphone, etc.

For this project, we only used the multi-array microphone which is a four-microphone array with 24-bit mono pulse code modulation at 16 kHz sampling rate. It has the linear array geometry looks as follow:

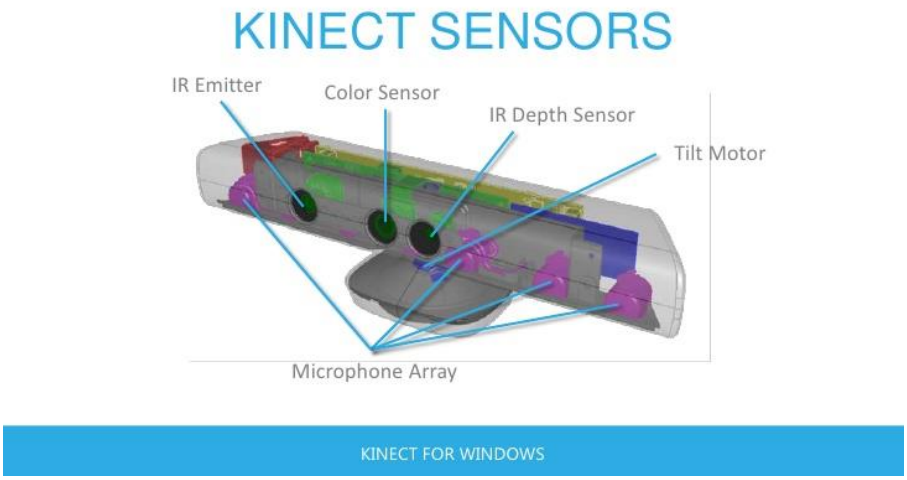

Fig 3. Kinect Sensors

All the microphones are identical and there are fixed distances between each two microphones. The microphone array enables several user scenarios such as high-quality audio capture, focus on an audio coming from a particular direction with the help of beamforming algorithm, and identify the direction of the sound sources [6]. It also allows the raw voice data access, which greatly simplified the task of recording audio sample in this project.

2.4 Hidden Markov Model Toolkit

The Hidden Markov Model Toolkit (HTK) is a free and portable toolkit which is mainly designed for building and manipulating Hidden Markov Models (HMMs) for speech recognition researching, although it has been widely used for other topics such as speech synthesis, character recognition and DNA sequencing (Wikipedia). The HTK consist of the following:

- **HTKLib** is a set of library modules which is used for building the HMM tools
- **HTKTools** is a set of command line tools which is used for building HMM models
- **HLMLib** is a set of library modules which is used for building the HLM tools
- **HLMTools** is a set of command line tools which is used for building n-gram language models. The n-gram language model is used to exploit the ordering of words, in which the n represents any integer greater than zero.

- **HTKBook** is a reference document which includes all basic theories of HMM and n-gram language models, usage manual for various tools and tutorial examples

In other words, the HTK can be described as a set of modules which can be called both command line and script files. According to the HTKBook as a reference document, we know that there are two major processing stages involved in HTK such as Training Phase and Recognition Phase. Firstly, the training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools. The HTK processing stages are data preparation, training, testing/recognition and analysis.

In the actual processing, the HTK firstly parameterizes features which are provided of speech data in various forms such as Linear Predictive Coding (LPC) and Mel-Cespstrum. Secondly, the HTK will estimate the HMM parameters by using the Baum-Welch Algorithm for training. Recognition tests are executed by estimating the best hypothesis from given feature vectors and from a language model. Results are given by the value of recognition percentage, as well as numbers of deletion, substitution and insertion errors.

The HTK is an optimal tool kit for speech recognition research, although it is a free toolkit and well-known which has been widely used not only for speech recognition research. It can also use for other pattern recognition, for example, facial recognition, handwriting recognition. Although many speech recognition researches have already exploited the HTK for different types of speech recognition researches, so it is worth using in our project.

3. Method and Implementation

Microsoft Kinect already have built-in beamforming algorithm and provide the SDK to interact with this feature. For this project, we will utilize this feature to analyze beamforming performance when recording with Kinect device. In this implementation, we used C# to connect with the Kinect to record the voice from the speaker, and we analyze the result using Audacity software to compare the recording and HTK software to test the speech recognition performance with those recordings.

The speaker for this project is only one person, because we want to make sure that the result will be uniform when we conduct the testing. Also, with the same speaker on each test, we can make sure that the data is free from variation of speech articulation. For the testing environment will conduct inside a normal room with minimum noise and echo.

The implementation for this project can be divided into two parts. The first part is the test of noise reduction of Kinect Beamforming by using Audacity to analyze the frequencies. Another part is the test of Speech Recognition using HTK tools with different setting: Beamforming and Single microphone. By comparing the recognition results, we intended to find out how Beamforming algorithm affects the accuracy of the speech recognition. The details of the implementation will be shown in the following sections.

3.1 Noise Reduction Test

In order to test the noise reduction of the Kinect Beamforming, we recorded the sample by using the smart phone to make the sound resource keep the same. So, with this method, the source is uniform and the result is suitable for noise reduction test.

Firstly, we created the white noise which has amplitude value 0.8 by using Audacity. Then we played the white noise and the recorded sample at the same time to simulate a same noisy environment. Although this test is not the same with normal noise in the real world, but we can compare the result of beamforming feature to reduce the noise.

For this implementation test, we are using 2 methods to compare the result. The method described as follows:

1. Using Kinect single microphone.

   To record from each microphone using C# is very simple and straightforward process. It can be done by following these steps:

   - Create a buffer for recording.
   - Read the Kinect audio stream.
   - Write the buffer to the file system.

   To check the recording file, we are using Audacity software, The result consisted of 4 audio channel banks. Each bank is correspondence with a single microphone on the Kinect device. For this project, we will use the last channel as the reference.

2. Using Kinect Beamforming.

   We can use the KinectAudioSource class to activate Kinect Beamforming feature, it will capture the audio data from the sensor and take control over audio processing by Kinect, including Beamforming feature. We can set the Beamforming angle mode to manual, automatic and adaptive, also we can set other capability like noise suppression and echo cancelation.

   The recording result when using Kinect Beamforming feature is consisted by a single audio channel. It is the result of the built-in beamforming algorithm inside Kinect which has calculated all the audio stream from each microphone and process it to minimize the noise and echo on the audio stream.

3.2 Speech Recognition Test

In order to get a good result, we used four settings and single speaker to record the utterances and perform the speech recognition. The utterances are digits that randomly generated by HTK tool and the recording place is the same quiet environment to keep the unrelated factors constant.

The audio component of the Kinect audio sensor is a four-element linear microphone array. In order to record the utterances using a single microphone, we used a sample application called “AudioCaptureRaw” provided by Kinect for Windows SDK which uses the Windows Audio Session API (WASAPI) to capture the raw audio stream from the microphone array of the Kinect and

write it to a .wav file [7].

For the Kinect beamforming recording we have implemented a simple application to read the audio signals from the microphones into a data buffer and then simultaneously write the data to an output .wav file.

There was also another sample application from Kinect for Windows SDK called “AudioBasics” was used to get the values of the angles between the speaker and the Kinect, we also measured the distance between the speaker and the Kinect in order to have an exhaustive result.

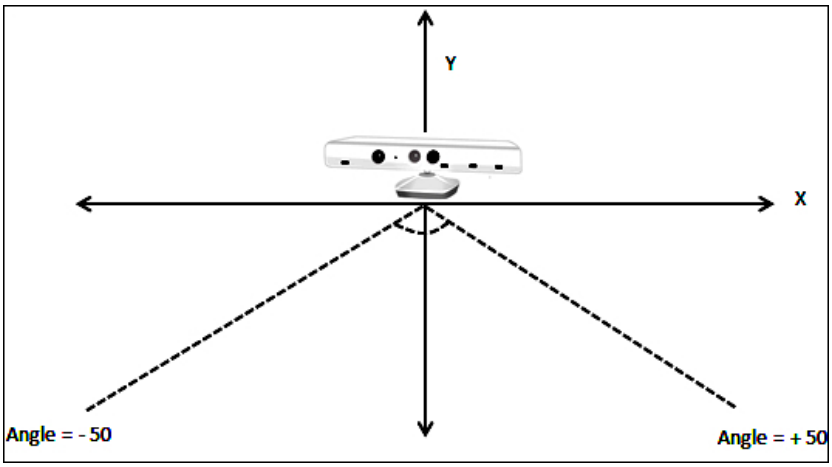

Fig 4. Kinect Angle Figure

So, according to the factors above, we designed four settings as follow:

1. Set 1: Recording uses Kinect Beamforming with positive angle (10 degree) and constant distance (0.5 m).
2. Set 2: Recording uses one single microphone channel of the Kinect with same angle and distance in set 1.
3. Set 3: Recording uses Kinect Beamforming with negative angle (-20 degree) and constant distance (0.5 m).
4. Set 4: Recording uses one single microphone channel of the Kinect with same angle and distance in set 3.

For each setting, we performed both for the training task and testing task.

4. Result

4.1 Noise Reduction Test

To compare the noise parameter in the recording, we use Audacity software to analyze the waveform of audio recording. If we check the single microphone recording file with Audacity, we will find 4 audio channels. It has happened because Kinect has 4 microphone array on a single device, and all of them can record simultaneously. Furthermore, we can see a slight difference on each of the recording file, like the delay and amplitude of the sound showed in figure 5 below.

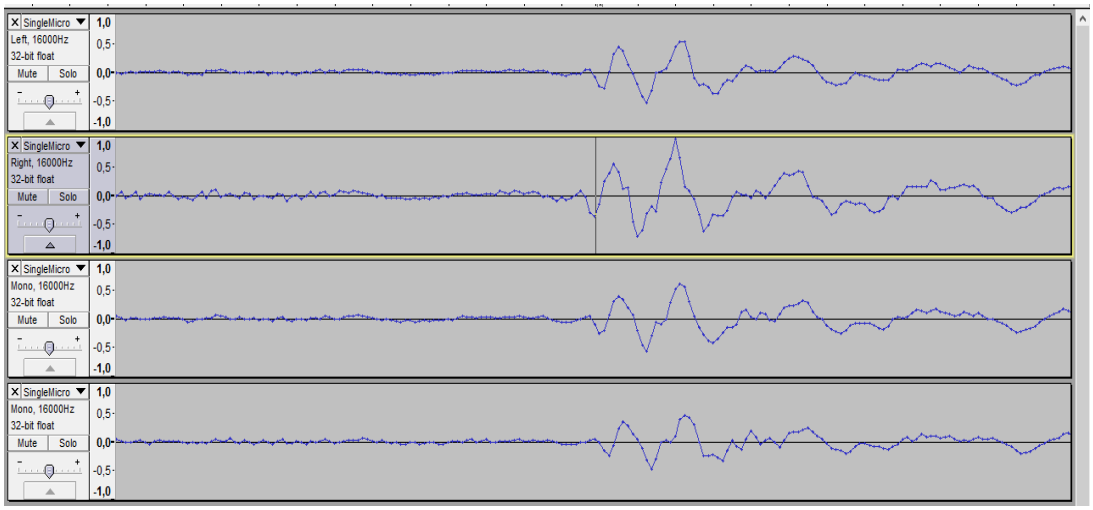

Fig 5. Single microphone record

From figure 5, we put the source of voice with negative angle (-20 degree) and 0.5 m distance. So, mic 2 will get first signal wave and then follow up with mic 1, mic 3 and lastly mic 4. In this project, we use mic 4 recording to compare the foreground voice and back ground voice to check the differences between real voice and the background noise.

To compare the noise reduction feature, we use the same clip of audio file to simplify the measurement. The result described as follows:

1. Using Kinect Single Microphone

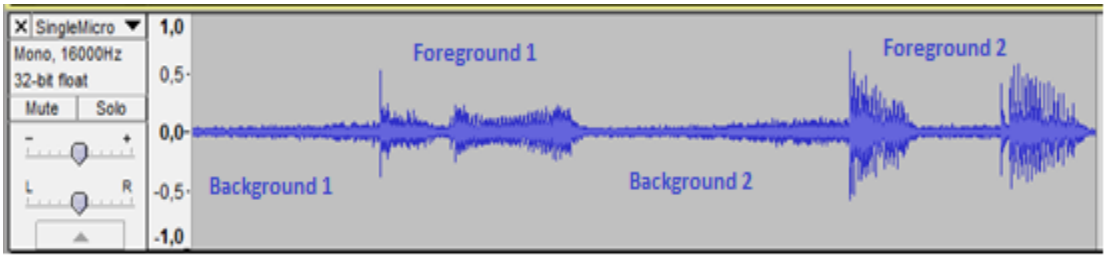

Fig 6. Audacity record analysis for Kinect single mic

In the figure 6, we clip some utterance from the recording to make it more simple and easy to analyze. Sample on figure 6 contain 2 utterances, and there is some silent phase before the utterance start. Hence we labelled it “Background 1” and “Background 2”, and for the utterance itself, we labelled it “Foreground 1” and “Foreground 2”.

To analyze further, we are using contrast function on Audacity, it will compare the differences of volume in each phase, and calculate the average from single mic recording.

- The contrast between Background 1 and Foreground 1 is 10 dB.
- The contrast between Background 2 and Foreground 2 is 11 dB.

Therefore, the average differences with 2 utterance and 2 silent phase is around 10.5 dB.

2. Using Kinect Beamforming

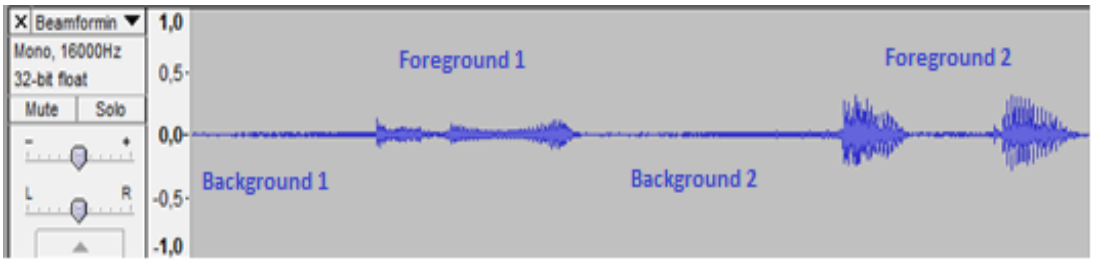

Fig 7. Audacity record analysis for Kinect single mic

In the figure 7 shows the audio clip recording using Kinect Beamforming. The condition is the same like previous recording in figure 6 with 2 utterances and silent phase. In this experiment, we also use Audacity to analyze the differences of volume in each phase. The result described below:

- The contrast between Background 1 and Foreground 1 is 15.6 dB.
- The contrast between Background 2 and Foreground 2 is 19.8 dB.

Therefore, the average differences with 2 utterances and 2 silent phases is around 17.7 dB.

4.2 Speech Recognition Test

To conduct a speech recognition test, we are using the HTK Toolkit to analyze the audio stream that produces by Kinect device. The form of audio file is .wav that already saved using a sample application called "AudioCaptureRaw" provided by Kinect for Windows SDK. In this project we test the recognition performance by using 4 sets of condition as we proposed in implementation parts.

In this implementation, we compare each of set of training data with another set of test data and takes a note of each performance. For example, we compare set 1 as training data with the set 1 to 4 test data, and record the accuracy performance when recognizing the utterance. To analyze the result, we are using HTK Tools, and the table below will summarize the result of the experiment.

No	Training speaker	Test speaker	Accuracy	Insertion	Deletion	Substitution	Average Set
1	Set1	Set1	100,00	0	0	0	89,38
2	Set1	Set2	77,50	0	0	9	
3	Set1	Set3	97,50	0	0	1	
4	Set1	Set4	82,50	0	0	7	
5	Set2	Set2	77,50	0	0	9	65,00
6	Set2	Set1	52,50	0	0	19	
7	Set2	Set3	35,00	0	0	26	
8	Set2	Set4	95,00	0	0	2	
9	Set3	Set3	97,50	0	0	1	82,50
10	Set3	Set1	95,00	0	0	2	
11	Set3	Set2	62,50	0	0	15	
12	Set3	Set4	75,00	0	0	10	
13	Set4	Set4	100,00	0	0	0	79,38
14	Set4	Set1	55,00	0	0	18	
15	Set4	Set2	95,00	0	0	2	
16	Set4	Set3	67,50	0	0	13	
Set 1: Beamforming with Kinect, distance 0.5m, Audio Beam Angle 10deg, Audio source Angle 5 deg							
Set 2: Single microphone with same distance and angles as Set1							
Set 3: Beamforming with Kinect, distance 0.5m, Audio Beam Angle -20deg, Audio source Angle -15 deg							
Set 4: Single microphone with same distance and angles as Set3							

Table 1. Table of experiment comparing the recognition accuracy of each set.

From the table 1, we can compare the accuracy based on the model on set 1 to set 4. Looking at the average accuracy of each set we can rank them from the highest accuracy to the lowest accuracy. Therefore, the highest is achieved by set 1 and followed by set 3, set 4 and lastly by set 2.

If we grouped each set based on the Beamforming feature, then we have 2 separate groups, one with Beamforming feature and the other using single microphone on Kinect. So, the average of each test is:

- Beamforming with Kinect

$$\frac{set1 + set3}{2} = \frac{89.375 + 82.5}{2} = 85.94 \sim 86\%$$

- Single Mic with Kinect

$$\frac{Set2 + Set4}{2} = \frac{65 + 79.375}{2} = 72.19 \sim 72\%$$

From calculation above, the accuracy using Beamforming is clearly much higher comparing to single microphone recognition accuracy.

5. Discussion

Based on the experiment above, using Beamforming with Kinect devices is clearly can reduce the noise from the surrounding environment. In the test above, we can conclude that when using Beamforming, the differences in noise level and the actual voice data is almost 18 dB. On the other hand, the single microphone result is only 10.5 dB and that means the differences between the voice and the noise is not clear enough. Based on the Audacity audio contrast tool manual, the differences between foregrounds and background at least 20 dB to make sure people understand the speech (foreground) [8]. So it is clear that the Beamforming is indeed reducing the noise.

The second experiment is to analyze if the Beamforming feature on Kinect device can improve the accuracy to recognize the utterance from human. From the simulation above, we can get the accuracy comparison between the result with Beamforming feature and with single mic feature when recording via Kinect device. The difference between both features are almost 15% with the Beamforming result as the best result. This performance might be affected by the noise reduction and echo cancellation feature that can be set in Beamforming configuration. If we correlate with the previous experiment, then the result is aligned, because with Beamforming, It can reduce all the noise that is not needed for speech recognition.

Therefore, the final result shows that Beamforming should perform more accurate comparing the result without using Beamforming feature.

Finally, we can conclude that Beamforming feature in Kinect devices could increase the capability of the Kinect device to deliver clearer sound to process with speech recognition. Even though, when we conduct the experiment is not using a real crowd noise but using white noise. Also, we are working in labolatory room, which is more ‘quiet’ comparing the real room noise. However, our experiment result should align with real world implementation. So, for the next experiment, we suggest that this method can be tested in real room condition with more various noises and utterances to be tested.

6. Conclusion

We have shown with our experiment that the Beamforming feature in Kinect device could increase the performance of speech recognition. Our method when conducts the experiment is using built-in features of Kinect Beamforming and test the output result using Audacity and HTK toolkit as tools to analyze the output. Although the test is not conducted in real world, but it is still valid because the result is aligned and have correspond with our initial though.

In this project, we also describe how we can set up the environment for the experiment. The application we used for this experiment is available free on the internet, also if you need to build the tools to record the output from the Kinect is not so complicated.

7. Reference

1. Beamforming, available from: < http://en.wikipedia.org/wiki/Beamforming >. [18 May 2015]
2. Salvi, G (2015) Lecture notes distributed in Speech and Speaker Recognition at KTH, Stockholm.
3. Microsoft Research, Microphone Array. Available from: < http://research.microsoft.com/en-us/projects/Microphone_Array >. [18 May 2015]
4. Tashev, I. and Malvar, H.S. (2005) A new beamformer design algorithm for microphone arrays, Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP '05). IEEE International Conference on, vol. 3, pp. 101-104.

5. Greensted, A (2012) Delay Sum Beamforming. Available from: < http://www.labbookpages.co.uk/audio/beamforming/delaySum.html >. [20 May 2015]
6. Microsoft ISDN, Audio Stream. Available from: < https://msdn.microsoft.com/en-us/library/jj131026.aspx >. [20 May 2015]
7. AudioCaptureRaw Walkthrough: C++, Available from: < http://download.microsoft.com/download/f/9/9/f99791f2-d5be-478a-b77a-830ad14950c3/audiocaptureraw_walkthrough.pdf >. [20 May 2015]
8. WCAG 2.0 Audio Audio Contrast Tool Help For Success Criteria, Available from: < http://www.eramp.com/WCAG_2_audio_contrast_tool_help.htm >. [20 May 2015]