tocdepth6

Declaration of Authorship

We, Shuang Wang, Chenyao Diao and Manan Lamba, declare that this project titled, **"Investigating the perception of a nearby wall by exploring the virtual room with self-produced oral sounds"** and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in partial fulfillment of a research degree at this University.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given.
- We have acknowledged all main sources of help.

Signed:

Shuang Wang: ____________________

Chenyao Diao: ____________________

Manan Lamba: ____________________

Date: ____________________

Abstract

This study investigates the realization of human echolocation in virtually acoustic environment(VAE). Capabilities of fully sighted people are examined to perceive obstacles and surfaces by using auditory senses, just like the visionless people. For this purpose, human's own voice is used as an input in real-time. Software based convolution system *Pybinsim* has been altered to process the real time oral speech.

Characterization of any room is done by measurement of room impulse response from the mouth to the both ears of same head, which is termed as Oral binaural room impulse response, OBRIR. A test person provided with near mouth microphone and headphones can speak and hear their voice simultaneously, with a minimal delay. For this purpose, an algorithm is created and implemented, which processes the tester's oral sounds in real time. This experience complies to the being present in a place like any normal room while actually being in a Virtual environment. According to the corresponding tracking data, the OBRIR filters are updated for convolution with tester's voice. Integrating a real time audio input is challenging in terms of the system delay. One goal of this realization is to attain the minimum postponement(latency) to enable a correct representation of a nearby virtual wall.

HTC Vive is used in order to conduct the listening test on human echolocation in a Virtual Acoustic Environment (VAE). The contributors to the listening tests are asked to rate their experience on the parameters of own voice perception by localize a nearby virtual wall. An informal listening test was conducted and the results suggest that the shorter the distance between the speaking listener and the wall, the more accurate the localization.

The implementation and testing reveals that integration of real-time self produced oral sounds in virtual room is a cumbersome task. The delay of mouth produced sounds is highly influenced by the hardware. Large time delays are encountered in the process.

Kurzfassung

Diese Studie untersucht eine Realisierung von menschlicher Echoortung in virtuellen akustischen Umgebungen. Die Fähigkeiten von Menschen mit normalem Sehvermögen, Hindernisse und Oberflächen wie Blinde mit Hilfe des auditiven Sinnes wahrzunehmen, werden untersucht. Zu diesem Zweck wird die menschliche Stimme als Input in Echtzeit genutzt. Das Software-basierte Faltungswerkzeug PyBinSim wurde angepasst, um Live-Sprache in Echtzeit verarbeiten zu können.

Die Charakterisierung eines Raums erfolgt durch Messung einer Raumimpulsantwort vom Mund zu den beiden Ohren, welche Oral-Binaurale Raumimpulsantwort (engl. Oral Binaural Room Impulse Response, OBRIR) genannt wird. Eine Testperson, ausgerüstet mit einem Mikrofon nah am Mund and Kopfhörern, kann gleichzeitig sprechen und die eigene Stimme mit einer minimalen Verzögerung hören. Dafür wurde ein Algorithmus geschaffen, welcher die oralen Geräusche der Testperson in Echtzeit verarbeitet. Dieses Erlebnis gleicht der Präsenz in einem Ort wie beispielsweise einem normalen Raum, während man sich in einer virtuellen Umgebung befindet. Entsprechend der getrackten Kopfposition und –orientierung werden die OBRIR-Filter kontinuierlich aktualisiert. Eine Integration von Echtzeit-Audiosignalen am Eingang ist besonders anspruchsvoll im Hinblick auf die Verzögerungen durch das System. Ein Ziel dieser Realisierung ist es, eine minimale Laufzeitverzögerung zu erhalten, um eine korrekte Abbildung naher virtueller Wände zu ermöglichen.

Eine HTC Vive Brille wird genutzt, um ein psychoakustisches Experiment zur Echoortung in einer virtuellen akustischen Umgebung durchzuführen. Die Teilnehmer des Versuchs werden gebeten, ihr Erlebnis der eigenen Stimme anhand der Lokalisation einer virtuellen Wand zu bewerten. Im Rahmen des Projektes wurde ein informaler Hörtest durchgeführt. Dessen Ergebnisse deuten an, dass die Genauigkeit der Lokalisation der virtuellen Wand mit geringerer Distanz zwischen sprechendem Hörer und Wand besser wird.

Die Implementierung und die Tests zeigen, dass die Integration von selbst-erzeugten oralen Geräuschen in Echtzeit in den virtuellen Raum ist eine schwierige Aufgabe. Die Verzögerung der oral erzeugten Geräusche ist stark von der Hardware abhängig. Große Zeitverzögerungen wurden in dem Vorgang entdeckt.

Acknowledgements

First and foremost, we would like to extend our gratitude and appreciation towards our Supervisor, **Dipl.-Ing. Annika Neidhardt** for her constant support and guidance. Without her words of encouragement we would not have achieved these results in this field of study.

Many thanks to **Prof. Dr.-Ing. Dr. rer. nat. h.c. mult. Karlheinz Brandenburg** as he is a source of inspiration for many generations of audio engineers.

Mr. Matthias Döring was also very helpful in providing us the proper equipments and giving us useful demos.

Contents

List of Figures

List of Tables

Abbreviations

BRIR	Binaural Room Impulse Response
OBRIR	Oral Binaural Room Impulse Response
BRS	Binaural room scanning
VAE	Virtual Acoustic Environment
SPL	Sound Pressure Level
DRR	Direct to Reverberation Ratio
HMD	Head Mounted Display
HRIR	Head Related Impulse Responses
HRTF	Head Related Transfer Function
ILD	Interaural Level Difference
ITD	Interaural Time Difference
ITDG	Initial Time Delay Gap
DRR	Direct to reverberation ratio
OSC	Open Sound Control
RP	Repetition Pitch

Chapter 1

Introduction

Virtual reality is flourishing commercially since the last decade and it is still thriving for more expansion. From gaming industry to healthcare, everyone is in competition to invent the next unbelievable device. With higher consumer expectations and huge financial investments, manufacturers are in quest to leave the audiences awestruck.

Virtual reality is a computer generated three dimensional experience taking place in a simulated environment that involves visual, auditory and other sensory elements. Virtual Reality has been described as I^3 for *Immersion-Interaction-Imagination* [9].

Any virtual environment which is able to evoke these three aspects of a human mind is successful. The *immersion* of a virtual environment can be described as the degree of involvement in the virtually created real world, of the testing person sensorially or emotionally. *Interaction* is how the virtual setup responds to the user inputs such as voice commands, gestures or through input devices. *Imagination* reflects the human imagination, realness of problem solving application or closeness to the real world.

Human Echolocation is a very useful and quality of life improving technique, generally used by the many blind people. It is an art of navigating or scanning the surroundings through the echos. It can be safely said that through this technique, the blind are able "*see*" the world.

This gives rise to the research question for this study:

What would be the results for comparison of recorded voice vs own voice in virtual room?

The basis of this study being echolocation, the main focus is on the perception of one's own voice i.e *autophonic perception* and the effect the room environment has on it. The research addressing the characteristics of autophonic perception in rooms has been fairly limited [10]. As a result, most studies that analyzed some facet of autophonic perception within real room environments have been limited by the number/variety of rooms tested, for a statistically meaningful number of participants [11]. By autophonic perception, it would be clear how good a listening speaker can recognize an obstacle or a surface, in this case a virtual wall.

Oral Binaural Room Impulse Response (OBRIR) filter according to the user's head orientation is chosen and is convolved in *Pybinsim* [12] with the input voice signal of listening speaker, then played back in real time.

Chapter 2

Fundamentals

The goal in virtual acoustics is to deliver any acoustical message (voice, music, noise) in a virtual reality system from the source to the receiver as it would happen in a real or imaginary acoustical situation [13]. In this study, the source and the receiver are at fixed distance from each other. To achieve the stated goal, understanding the fundamentals of sound localization is crucial. Echolocation is of greater interest as well as know how of room acoustics and dynamic binaural synthesis is also desired for the presented study.

According to C. Pörschmann [14], autophonic perception in a room consists of the following three pathways:

a) Sound conducted directly from the mouth to the ears of the same head (direct-airborne conduction).

b) Sound conducted through the internal structures of the human head to the cochlea (bone or body conduction).

c) Room reflected sound, which includes reflections from relevant surfaces in the room's environment (indirect-airborne conduction).

For this study, the source and the receiver are concentric i.e within the same head. Therefore, only the third path is considered as the first two conduction paths are not simulated for being already present with the speaking listener.

2.1 *Echolocation*

The authors feel the obligation to discuss echolocation in detail in order to understand the procedure and results of this study.

Echo and location together make the word echolocation. Clearly, it is the ability of some animals like bats, dolphins and whales to make use of echos from the surroundings to navigate around and hunt. Humans, specially the blind ones use their hearing and touch as a substitution to vision. The clicking or tapping sound produces echo by being reflected from the surfaces. However it's not so common to see the blind making clicks while walking, rather many prefer tapping their canes on the surface while walking in order to find an obstacle-free path for themselves. According to Thaler[15], no detailed data is available but anecdotical reports suggest that only 20 to 30% blind people use echolocation in daily life, which is also mentioned by *Neidhardt et al* [1].

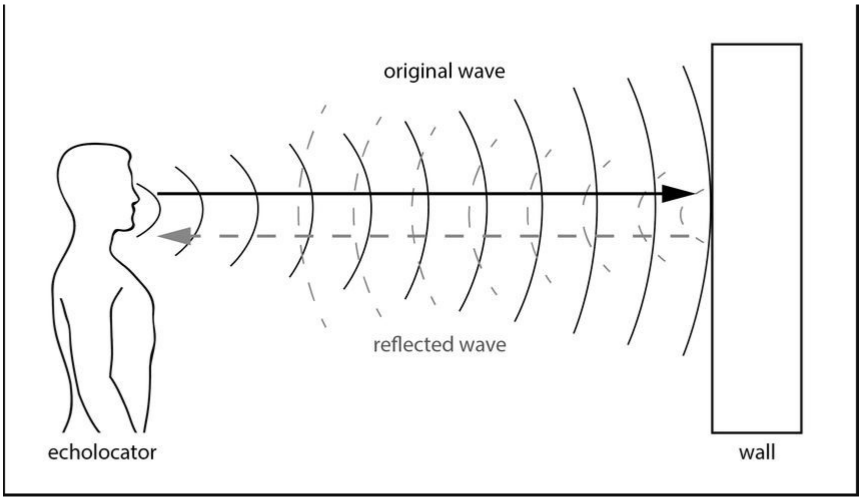


FIGURE 2.1: Basic principle of Human Echolocation[1]

Sound localization is the backbone of echolocation as the sound pressure level (SPL) in nearer ear towards the obstacle is higher and it helps the blind people to echolocate. But echolocation doesn't happen equally well for all types of sounds. It is a common belief that the use of high frequency sounds actually help humans echolocate[16]. It might be because high frequency sounds have smaller wavelength, which makes it better to reflect from surfaces and objects. However, the high frequency sounds might not work well for greater distances as well as for the curved or tilded surfaces as they tend to scatter[17]. The fact that the high frequency signals are easily suppressed by low frequency, therefore using low frequency sounds for echolocation in noisy surroundings can be a better choice[17].

Besides object detection, echolocation may give information about object's distance and azimuth, shape, material, internal structure, size, motion, and time to contact [18]. The quality of the perceived reflections from a surface is influenced by the type of orally produced sound, the surface which will produce an echo and the receiver, in this case the ears.

There are five basic specifications of any sound perceived by humans: *directionality, pitch, timbre, intensity and envelop.* The oral sounds or clicks produced for echolocation should have a *directionality* in order to know where the echos come from. Humans are able to distinguish a large scale *pitches* as the dynamic range of human ear is from 20hz to 20khz [19]. *Timbre* is the uniqueness of the sound and is often usefull in noisy environments. It is easier to echolocate in a crowded place if a unique sound is produced. *Intensity* is the loudness of the sound generated and is measured in decibels (dB). Another parameter is *envelope*, consisting of three prerequisites: *rise time, sustain time and decay time.* The *rise time* or *onset*, is the amount of time a sound signal needs to reach its peak from zero. The *sustain time* is how long the signal stays at its mean intensity. *Decay time* being the time taken to drop from the mean of intensity to zero. Practically in mathematical terms, envelope is the contour of any signal. It helps the person to memorize the type of sound and to locate it better.

In order to perceive a virtual wall by the user's own voice, the focus is only on the surface detection aspect of echolocation. Perception of the presence of a surface by the reflected echos is of great significance, as an echo will only be produced if a surface is physically or virtually present. As stated earlier, the shorter the distance between testing person and the wall, the better the reflection. Hence there will be a chance of better echolocation. In addition to this, size of the surface also matters in proper echolocation.

Self produced sounds work the best for the purpose of echolocation as they can be easily modulated by the user and do not need maintenance like external sound producers for example electronic beeping device or a cane. Flexibility of the produced signals is for sure better as compared to the signals from electronic devices as they have some additional advantages as mentioned above. Commonly used sound signals by blind echolocators are hand claps, finger clicks, oral clicks or vocalizations. Although using hands has an advantage of strong intensity but it lacks directivity. Another positive point in support of orally produced sounds is that the source i.e mouth is located close to the ears, hence it should increase the chances of perceiving a surface. Therefore, many blind echolocators use oral clicks.

To conclude, it can be seen that self produced oral sounds prove to be vital in echolocation. Due to the same reason, the voice of the testing person has been used in real time in this study.

2.2 *Sound Localization*

To get familiar with the phenomenon of echolocation, sound localization, which serves as the most basic element to identify the direction of the sound, should also be reviewed. Therefore, the ability of estimating the direction and distance of the sound source falls under sound localization. Spatial hearing can be termed as the efficiency of the auditory system to render or decode unique paths of sound reaching the listener's ears.

Figure 2.2 represents the three dimensional co-ordinate system consisting of *horizontal, frontal* and *median planes.* The horizontal plane is parallel to the eyes, meaning the sound source locating at an angle of 0° i.e *azimuth angle($\boldsymbol{\varphi}$)*, will be right in front of the eyes. Median plane encircles the horizontal plane from azimuth angle 0° to 180°. The angles above and below the horizontal plane are known as *elevation($\boldsymbol{\delta}$).* The frontal plane is perpendicular to median and horizontal planes.

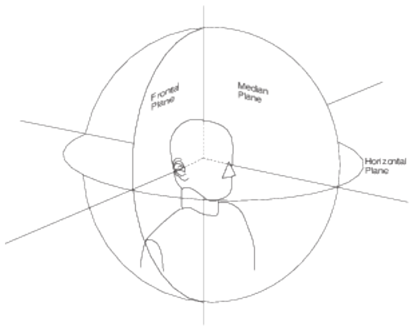


Figure 2.2: The co-ordinate system consisting of Horizontal,frontal and median planes.[2]

2.2.1 *Binaural cues*

Binaural means localization using both the ears. These cues come under two categories: *Interaural Time Difference(ITD)* and *Interaural Level Difference(ILD).* ITD through its name suggesting the perception caused difference in arrival times of a sound signal between the ears. ILD, on the other hand occurs due to the shadowing effect caused by circumfrence of the head, is the difference in intensities of the same sound entering the two ears of the same person. in [20], it is described that ITD is more prominent for low frequency and ILD is influenced by high frequency sounds, indicating towards the existence of *Duplex theory* given by Lord Rayleigh(1907), who in his study also stated that ITD and ILD overlap at 1KHz.

However,even if binaural cues are the primary source of localization of sound, still a listener can be sometimes puzzled about the location of the source. This is due to *cone of confusion*[21] caused by the the sound sources positioned at the base of an imaginary cone. This opaqueness may induce *front-back confusion*[22]. This can be overcome by *dynamic binaural cues* i.e tilting the head so as to refine the ability to localize a sound source.

2.2.2 *Monoaural cues*

Head, pinna and torso in human beings cause shadowin, angle dependent refraction, differaction and reflections of the sound waves,hence helping in the extraction of localization cues using single ear. Pinna has a greater role than often thought, according to *M. B. Gardner and R. S. Gardner* in [23], the cavities of the pinna act like resonators of the different frequency bands.The findings by Alan D. Musicant and Robert A. Butler in [24] also back the claim. The latter even proposing that monoaural cues help in determining the elevation angle, further solving the front- back confusion by providing information in the absence or minimal presence of binaural cues. However, pointing out the position of sound source in horizontal plane, meaning with binaural cues is always advantageous as to monoaural cues.

2.2.3 *Repetition pitch and Perceived loudness*

Spectral cues help in localization of sound. These are the cues with which, there is a minimal change in the artifacts of the sound. Sudden variance in pitch, timbre, tonality, and intensity can be apprehended by the human ear. These cues are very useful in the awareness of the surroundings. Two different kinds of spectral cues which can be useful for echolocation are discussed below.

As the person advances towards any reflecting surface, the nature of the sound varies. The fusion of direct and reflected sounds cause a coloration of the perceived sound called repetition pitch(RP) [25]. It can be also known as reflection tone, sweep pitch or time difference tone.

Thus, the fusion of reflections with the direct sound leads to a rise in overall energy at the ear of the perceiver and thereby perceived loudness is also increased([26]). Both pitch and loudness increase with decreasing distance to a surface. These two sources of auditory information are correlated in real environments. In addition to pitch and loudness, information about objects and the ambient environment is also provided by the timbre of the echoes which are essentially based on the spectral envelope and its variations [27].

2.3 Room Acoustics

In an enclosed space, when an auditory event takes place, along with the direct sound also the reflections of the sound from the surfaces plays a vital role in determination of sound source as well how distant it is.

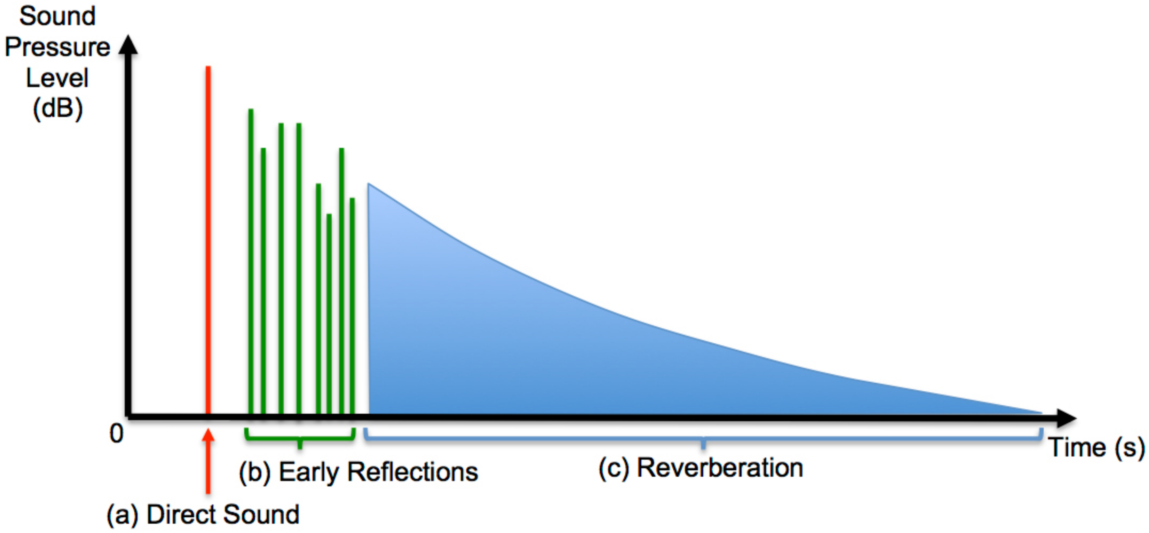


FIGURE 2.3: The room impulse response highlights the direct and reverberant sound energies w.r.t time and helps explore room acoustics. [3]

2.3.1 Room Impulse Response

Figure 2.3 explains the (omni-directional) impulse response of a normally enclosed room. J. Blauert in *The technology of binaural listening* says,*The filtering characteristics of a room for a combination of one sound source and receiver at certain positions can be simplified as a linear time-invariant (LTI) system [28].*

As displayed, the impulse response of any room has (a) direct sound(red), (b) early reflections(green) and (c) reverberations(blue). As explained by Blauert[22], the fusion of direct sound and the reflections takes place if they both arrive within limits(1-5 milliseconds for clicks, tens of milliseconds for complex music) in a reflective space. In this case, the listener tends to perceive a single auditory event due to *precedence effect.* If the delay is more than the above stated limit (a.k.a *echo* *threshold*), the listener is able to perceive *echoes.* Room acoustics are explained to a great extent through room impulse response of a given room. Further account of room acoustical features can be given by *Initial time delay gap and Direct to reverberant ratio* . The description is given in the following subsections.

2.3.2 Initial Time Delay Gap

As described before, every physical or virtual enclosed space consists of direct sound and its reflections from different surfaces. The difference of time between the direct sound and its first reflection from the close-by surface is called *Initial Time Delay Gap (ITDG)* [29]. For better perception, gap between direct sound and the first reflection should be as small as possible.

2.3.3 *Direct to Reverberant Ratio (DRR)*

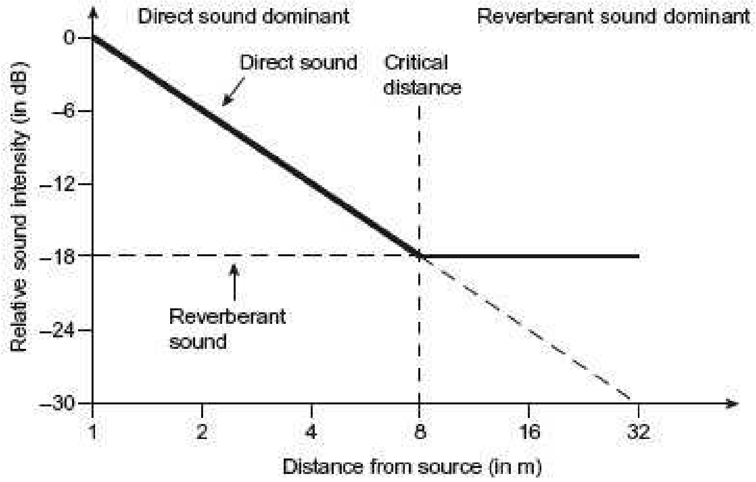


FIGURE 2.4: *Regions of dominance for direct and reverberant sound.*[4]

It is the ratio of the energies of the direct sound and the reverberant sound. It is expressed as:

$$DRR = 10\log_{10} E_d/E_r \tag{2.1}$$

where E_d and E_r are the energies of direct sound and reverberant sound respectively. *Reverberation radius or Critical distance* is the distance at which the intensities or energies of direct sound and reflected sound are equivalent [30]. Also, As one moves away from a source of sound in a space, the level of direct sound reduces but the reverberant sound stays constant. This means that ratio of direct sound to reverberant sound becomes less and so the reverberant sound becomes more dominant [31]. Figure 2.4 explains the statement given above.

2.4 *Binaural Synthesis*

BInaural synthesis being an outstanding technique for auralizing VAEs, is very useful for this project as the key concept behind it is to recreate the exact sound pressure level at the eardrums as in the physical world.

2.4.1 *Head Related Transfer Functions*

As explained by *Potisk [32]*, head-related transfer functions (HRTF) capture transformations of a sound wave propagating from the source to our ears. Some of the transformations include diffraction and reflections on the parts of our bodies such as our head, pinnae and torso. As a consequence, with these two functions we are able to create the illusion of spatially located sound [33]. HRTF is a Fourier transform of a head-related impulse response (HRIR). As a matter of fact it should be stressed that it is a complex function defined for individual ear containing the details of magnitude and phase shift. The HRTF is also highly dependent on the location of the sound source with respect to the listener, thus proving pivotal in localizing the sound source.

2.4.2 *BRIRs and OBRIRs*

If HRIR involves room acoustics then this is indicated by using the term BRIR or for its frequency domain equivalent BRTF [28]. BRIR stands for *binaural room impulse response* and consequently, BRTF implies *binaural room transfer* *function.* That means, filtering attributes of both *directional room impulse responses(DRIRs)* and HRIRs are included in BRIRs.

BRIRs are customarily measured by technique of using artificial head and torso simulator(HATS) with a steering mechanism identical to HRIR measurements. As it is known that practically and technically, an ideal impulse in terms of dirac distribution is not possible, hence sine sweep is used to evoke impulse responses. Software based measurement of BRIRs is also attainable with MCRoomSim, that fully synthesizes BRIRs [34]. However, the reliability of simulation softwares always needs to be considered, as there might be oblique differences of perception in listener's expectation and the simulated room.

Convolution of a BRIR with with a mono input sound signal in time domain or the multiplication of a respective BRTF with input in frequency domain results in binaural synthesis of a VAE. The input signal is replicated either via headphones or loudspeakers, headphones being the more economical solution out of both due to absence of cross-talk. Equalization of the headphones is mandatory.

OBRIRs are the *Oral binaural room impulse responses*, measured in a same way as BRIRs, only distinction being that the HATS has a mouth simulator. Binaural room impulse response is measured by the position of sound source at a random external positon in a room but in OBRIR measurement as indicated by oral, the sound source is considered to be in the same head as the receiver i.e ears. It is very interesting to observe here that the mouth is the source and ears are the receiver. Therefore, for the perception of self produced oral sounds, oral binaural room impulse responses are used. Rest comutational features remain same as BRIRs. OBRIRs, according to *Neidhardt et al [1]* were measured using G.R.A.S KEMAR 45BC. The detailed description of OBRIR measurement can be accessed in the mentioned paper.

Yadav et al [11] also talks about the measurement of OBRIRs with the help of ODEON [35] software for modelling the room acoustics. In order to measure the room reflected sound in the form of an OBRIR from the mouth to the two ears of the same head, *Cabrera et al. [36]* had suggested a method to perform a binaural

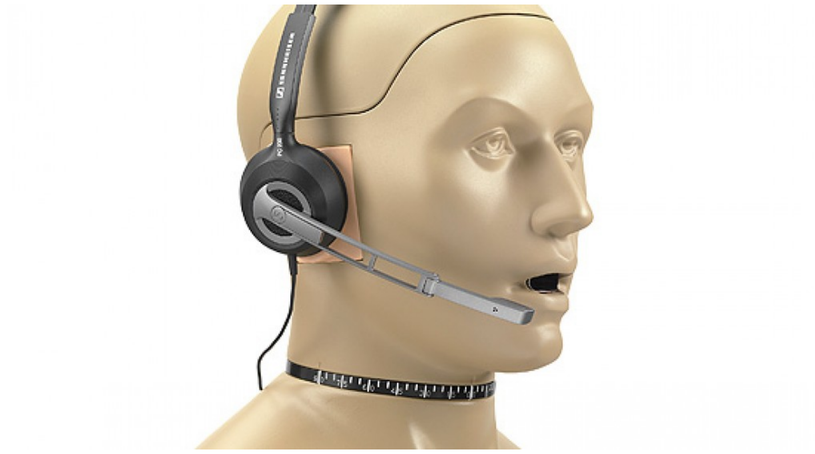


FIGURE 2.5: Head And Torso Simulator with mouth simulator. KEMAR 45BC[5]

room scanning with a swept-sinusoid measurement method, over any (or all) the six degrees of freedom of human head-movement seen in the figure below.

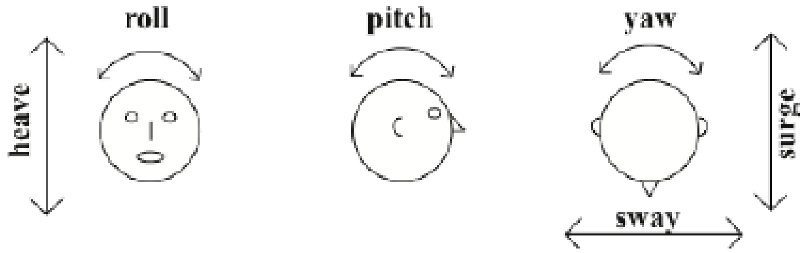


FIGURE 2.6: 6 degrees of freedom; 3 angular variations (yaw, pitch, roll) and 3 linear translations (sway, surge, heave)[6]

2.5 *Dynamic Binaural Synthesis*

Despite restricted degree of freedom, the *dynamic binaural synthesis* licenses the auralization of VAEs, in accordance to the movement of the listeners. Motion-tracking systems or conventional input devices restrain the movement in the virtual environment. According to the head-orientation of the testing listener, the relevant filter is selected and convolved with the signal. Due to dynamic binaural synthesis, the number of localization errors are reduced. This fact is well supported by the test results found by *Pauli Minnaar [37]*. In [38], Mackensen claims that dynamic binaural synthesis offers high quality listening for the headphone-based binaural room scanning (BRS) auralization system which further strengthens the explanation given before.

The set of BRIRs with specific angular details required by head rotation, are often limited to azimuth angle. The detailed view regarding dynamic binaural synthesis using oral binaural room impulse response has been discussed in [1, 11, 36].

Chapter 3

State Of Art

Many different techniques and methods are present for exact reproduction of binaural audio. Different aspects of binaural audio processing are being worked upon and the desire to improve the currently used systems is growing day by day. In this chapter, the authors attempt to give an overview of the current state of binaural technology. Firstly, a short description of state of the art in dynamic binaural synthesis with the tracking system is been given. In the second section, the previous studies to explore echolocation as well as the concepts related to autophonic perception with and without the use of VAEs is been discussed. In the subsequent section, introduction to *pybinsim* is given.

3.1 *Progression in Virtual Audio*

Development and evolution of virtual environment has brought new dimensions to the fields of audio and video. A virtual environment using audio i.e virtual auditory environment can be realized by using many loudspeakers, ear loudspeakers and even the headphones. Though there are some constraints of each of them. The listener position and head orientation plays a massive role as the sensation is always transmuted in the case of multi-loudspeaker usability. listener's displacement from the sweet-spot gives rise to transaural acoustic crostalks. Even if mechanisms for cancellation of cross-talk are in place, still its utility lies withing a specific area or sweet-spot for a listener. Ear loudspeakers have also been used for some experiments but headphones have proved to be the most trusted equipment as the listener using headphones is always in sweet-spot irrespective of the motion or head orientations. This is because the listener's position does not have any effect on the position of headphones position, it changes with the person. The facility of dynamic binaural synthesis along with the highly responsive motion tracking devices, any surface can be perceived in a virtual room. Not much research has been done in this direction to explore the facet of human senses.

3.2 *Previous studies on echolocation*

Numerous studies and researches have been conducted with distinct results about echolocation and voice perception in virtual environments. The chief element to perceive any sound source is its sound pressure. As it is known that in anechoic chambers, the intensity of any sound decreases by 6 dB, when the distance is doubled. But in real rooms the drop is approximately 4.25 dB [39]. This result can prove decisive in finding the surface through commands. Many initiatives have been taken by the researchers to find the degree of echolocation craft in humans in real enclosures. A detailed study is provided by Kolarik et al [40]. In the study "Echolocation versus echo suppression in humans" [41], statement that the blind people perform significantly better in an echolocation task is often reported. This claim has been given support by [42]. Additionally, it has been seen that people with normal vision can learn to distinguish between different distances of reflecting surfaces with a similar accuracy like the distance of sound sources [43].

Schenkmann and Nelson [44] used a loudspeaker positioned at the chest of a KEMAR facing large reflecting surface.Noise bursts were recorded at distances between 0.5 m and 5 m as well as without the surface. The same recordings were conducted in a conference room and in an anechoic chamber. Participants listened to these recordings in a listening test environment and had to identify the recordings without the surface in a choice of two. With increasing distance results became less accurate and the subjects performed better with the items recorded in the conference room.

Rowan et al. [45] uses an almost identical setup to study the effect of distance and orientation of reflective surface. A loudspeaker was placed 25 cm below and 5 cm in front of the intaural axis of a KEMAR dummy head in an anechoic chamber. Binaural impulse responses were recorded with a reflecting surface at distances between 0.9 m - 3 m and different orientations. The participants listened to bands of generated Gaussian noise, convolved with the measured impulse responses. The ability to distinguish left from right lateral position decreased with increasing distance. In [46], a system for simulating the room acoustical features in real world environment, making use of own voice is presented. It is shown that the measured OBRIRs are fed into a software based real time convolution system. The convolution system used in the study is implemented by hosting the commercially available VST plugin SIR2[47] in Max/MSP[48].

Echolocation Virtually

Audio and sensory researches are profoundly dependant on the binaural recording and recreational techniques. This paves the base of motivation to dive deeper into the world of blind people by imitating the world virtually to know about echolocation.

Diverting his attention to the commercial applications like teleconferencing, C. Poerchmann [14] contributed in the idea of virtual teleconferencing by integrating own voice in virtually created environments. In his study, it has been stated that bone conduction has no influence on the perception.An insertation loss on the transmission through the air is present while using headphones. This insursion loss when compensated, increases *presence* significantly. Though, these experiments were not thought to be a contribution to the studies related to echolocation directly but the awareness of ones own voice in VAEs is primitive. This provides a path towards the researches exploring echolocation using self generated sounds in virtual rooms. *Pelegrin Garcia et al.* [49] researched on interactive auralization of self generated sounds to address the features of human echolocation. In this study it has been concluded that experiences with the system by blind expert echolocators show that, the does not reproduce precisely the sensations of sound reflections occurring at close distances (closer than 3 m), but it provides helpful cues for the discrimination of distant walls or corners (beyond 3 m). The latency of the system was reported to be 3.5ms.

Neidhardt et al. [1] is the closest in relation to this project as the author has also touched upon the same technical aspects to explore surface perception by self produced oral sounds. The recorded oral sounds were used in this case.Three room scenarios were taken and the experiment was conducted in two phases, exploration phase in which the testing person was asked to walk towards the wall in order to get use to the virtual environment and testing phase in which the participants were asked to estimate the direction of the closest virtual wall. OBRIRs were measured and applied to *Pybinsim* which convolved the input sounds with OBRIRs.

3.3 *Pybinsim: A tool for binaural simulation in python*

Pybinsim [12] is dynamic binaural synthesizing open-source tool, designed for research and educational purposes. Pybinsim is a light weight application based on windows and Mac OS platform. Head orientation and other movements can be investigated easily in pybinsim. Due to the OSC-based interface, it is compatible with different kinds of tracking devices. Real-time binaural synthesis is achieved by applying chunks of directional filters on chunks of audio signals depending on the tracked listener data. Depending on listener's head orientation and position, appropriate binaural room impulse response(BRIR) is loaded. It convolves the BRIR with an audio sample, and reproduces resulting sound. Pybinsim is capable of processing the recorded speech signal in .wav format as input.

Chapter 4

Experimental Setup

The equipments that are used in this project are being described in this chapter.

A computer with pybinsim program installed, a head-tracking system, a near-mouth microphone(Sennheiser) to give the input voice signal and an open ear headphone(AKG K1000) to listen to the output audio.

4.1 *Sound Card*

External sound card i.e *RME Fireface UCX* is used to accommodate multiple input-output channels. The Fireface UCX is a highly integrated pro audio solution in an ultra-compact format for studio and live recordings[50].

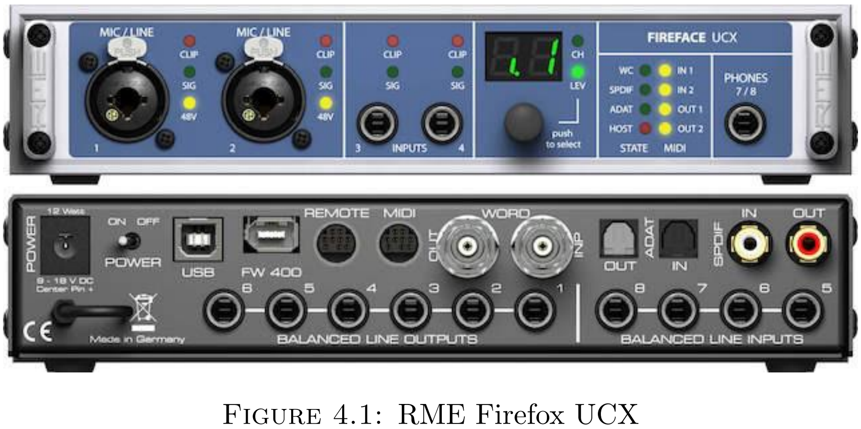


FIGURE 4.1: RME Firefox UCX

Sensitive in-ear microphones were used in both the ears one by one to measure the delay between the direct sound and its first reflection. Dummy head(HATS: G.R.A.S. KEMAR 45BA) was used to measure and confirm the delay from the mouth to each of the ears respectively.

4.2 *Motion Tracking Device*

For this project, HTC Vive was used as tracking device for user's motion. It is developed and manufactured by HTC corporation, a consumer electronics giant in collaboration with Valve corporation which is a video game developer and distribution firm in 2016 for immersive virtual reality applications.

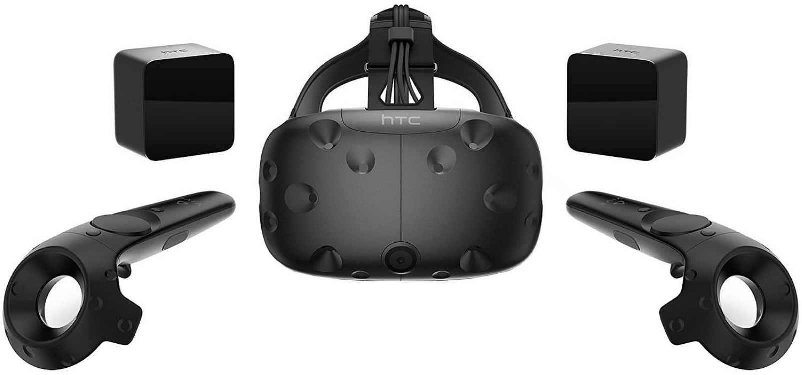


Figure 4.2: HTC Vive consists of HMD, a pair of controllers and a pair of Lighthouse base stations[7]

As shown in the figure above, HTC Vive comprises of Head Mounted Display (HMD), two lighthouse base stations as well as two controllers[51]. A display mounted on head in front of the eyes covers the entire field of vision and covers the view in 360°. [52] explains that the resolution of 1200 x 1080 for left and right eye is used for the two corresponding displays of HMD. Both the displays have refresh rate of 90 frames per seconds to avoid jitter. The base stations can track the physical position of the user with a play area up to 15ft x 15ft (4.6m x 4.6m). The base stations can be placed diagonally with a maximum distance of 16ft (5m) between them. The minimum play area is 6.5ft x 5ft (2m x 1.5m)[53]. Calibration of play area is done by using the two provided controllers. The controllers also serve the purpose of virtual hands in order to interact with the objects in the virtual fields. The safe area in the virtual zone is marked by a virtual blue colored grid. As the user gets to close to the boundaries of the play area, the grid appears, assisting the user to continue inside the assigned area.

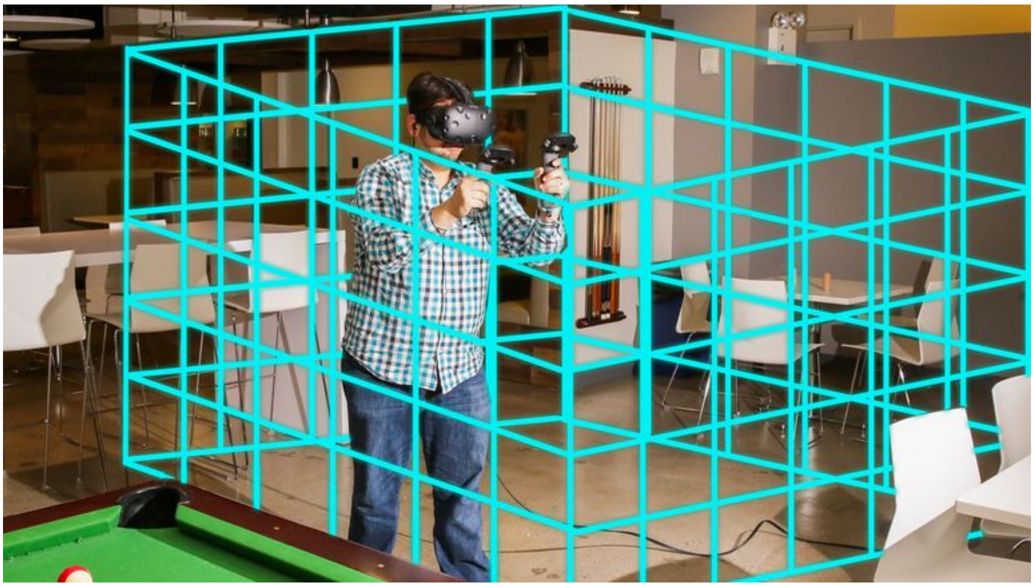


FIGURE 4.3: Blue grid: Calibrated area assigned for free user movement[8].

4.3 *System and Specifications*

Pybinsim, a python based software program is run on a desktop computer with the external sound card *RME Fireface UCX* connected to it. The computer with existing Windows 7 operating system has an Intel Core i7-6700K processor. A macbook computer with operating system macOS High Sierra 10.13.4 with intel core i7, 2.7 GHz and RAM of 16 gb is used. Python is used as the programming platform on which *Pybinsim* is based. An open source, cross-platform audio software *Audacity*[54] was used for the measurements and analysis of delay. *Adobe Audition CC*[55] was also tried for the comparison of measurements shown by Audacity but it produced the same results.

Chapter 5

Implementation

This chapter focuses on the in depth analysis of each step involved in the implementation. It involves detailed study and understanding about the operational procedure of rendering software used. This section also discusses the measurements of different types of delays and the use of delay data to develop new filters. Ultimately, a pre-listening test is talked about and its performance is shown.

5.1 *Pybinsim2: For real time self produced audio*

This section gives an overview of different parts and inner structure of *Pybinsim*. The *Pybinsim* already available, processes only .wav files as an input. For this project, there was a need for convolutional system which also works for live audio input. Therefore, *Pybinsim* is modified in order to process real time speech signal and is termed *pybinsim2*.

5.1.1 *Application Stack*

The diagram in the figure 5.1 illustrates the relationship between Pybinsim2, Pybinsim, Pyaudio, Portaudio, and the supported native audio API CoreAudio. The motivation of refactoring Pybinsim is to provide a convenient and general solution suitable for the research problems specific to this project.

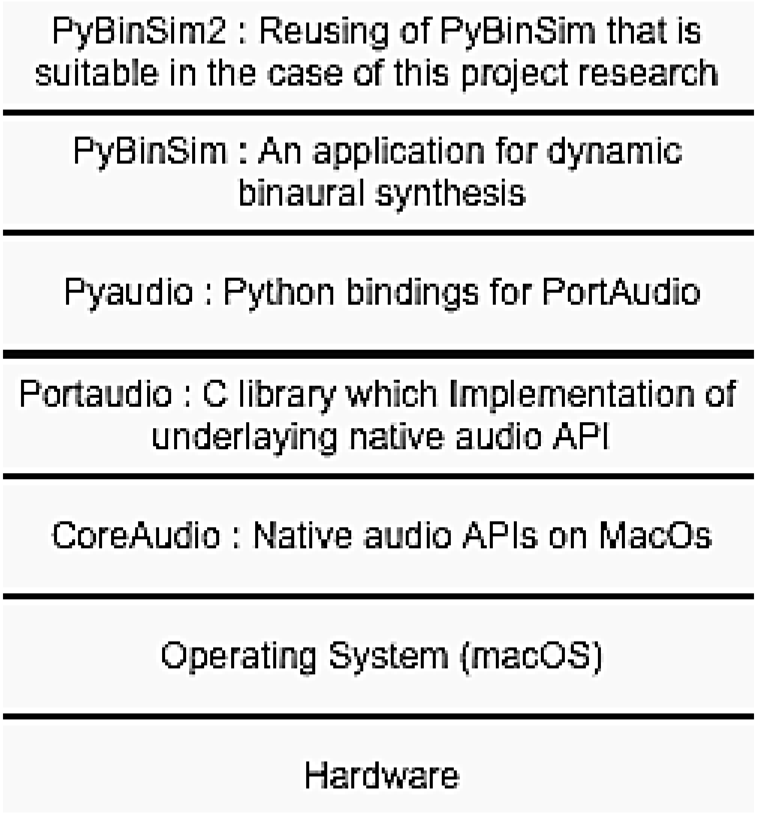

PyBinSim2 : Reusing of PyBinSim that is suitable in the case of this project research
PyBinSim : An application for dynamic binaural synthesis
Pyaudio : Python bindings for PortAudio
Portaudio : C library which Implementation of underlaying native audio API
CoreAudio : Native audio APIs on MacOs
Operating System (macOS)
Hardware

Figure 5.1: Application stack

The layered description provides an enhanced view about the different components which are responsible for the functionality of this system.

5.1.2 *Inner structure of Pybinsim2*

Figure 5.2 gives an insight to the inner file structure of this light weight rendering software. It shows the different stages of processing. Pybinsim2 performs main audio data processing as server side within the Client / Server mode, and it continues listening for incoming data, which are sent by client side. External tracking file acts as the client side that sends pose-data, which includes position and angle data of the tester. Whenever pose-data is about to change through tracking information, application.py works actively and the OBRIRs are refreshed.

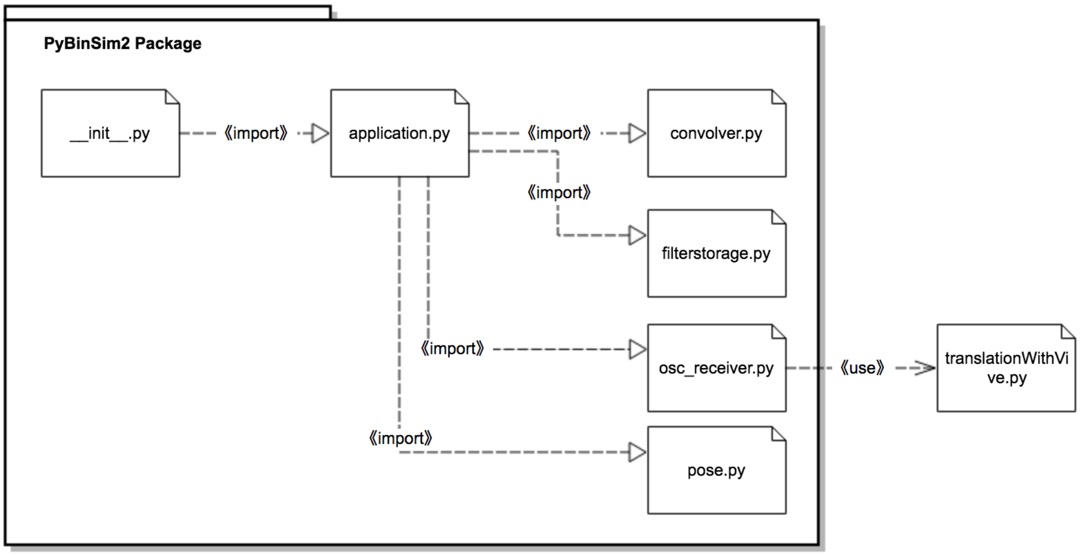


Figure 5.2: Inner structure of Pybinsim2

Data flow Diagram shown in fig. 5.3,on the next page precisely describes the path of detailed data processing between components of this system. The basic process happens in Callback function which executes the loop in a separate thread repeatedly until an outside interrupt occurs. The basic mechanism in the background of technique is interleaving and de-interleaving array processing of data.

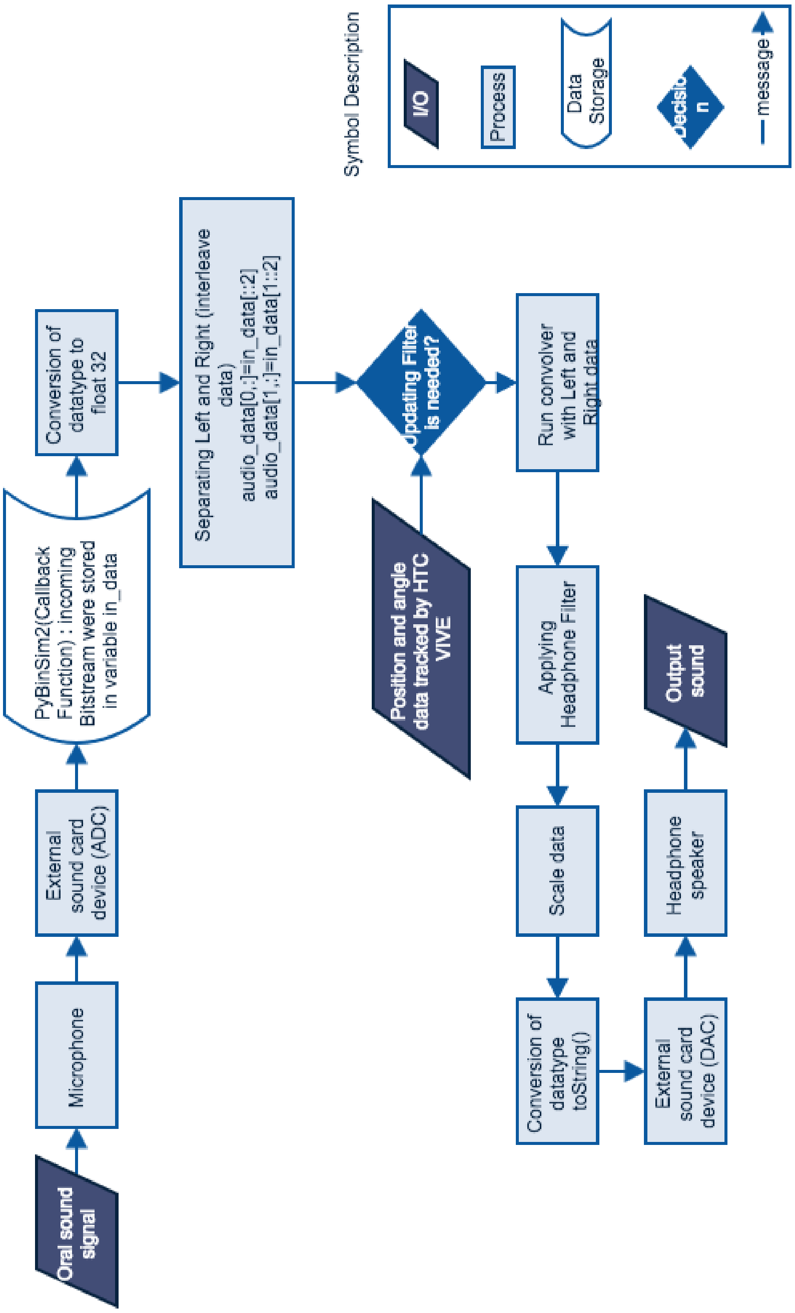

Oral sound signal
Microphone
External sound card device (ADC)
PyBinSim2(Callback Function) : incoming Bitstream were stored in variable in_data
Conversion of datatype to float 32
Separating Left and Right (interleave data)
audio_data[0,:]=in_data[::2]
audio_data[1,:]=in_data[1::2]
Updating Filter is needed?
Position and angle data tracked by HTC VIVE
Run convolver with Left and Right data
Applying Headphone Filter
Scale data
Conversion of datatype toString()
External sound card device (DAC)
Headphone speaker
Output sound
Symbol Description
I/O
Process
Data Storage
Decisio n
message

Figure 5.3: Data Flow inside *Pybinsim2*

Figure 5.4, showing self explanatory flow chart of *Pybinsim2* is given. A defined set of OBRIR (Oral Binaural Room Impulse Response) filters are fed in this version of pybinsim. Then, a listening speaker with the near-mouth microphone and the headphones constantly produces oral sounds and listens simultaneously in order to find the direction of the nearby surface. In VAEs, the head orientation data of the user through head-tracking system corresponds to the selection of appropriate OBRIR filter.

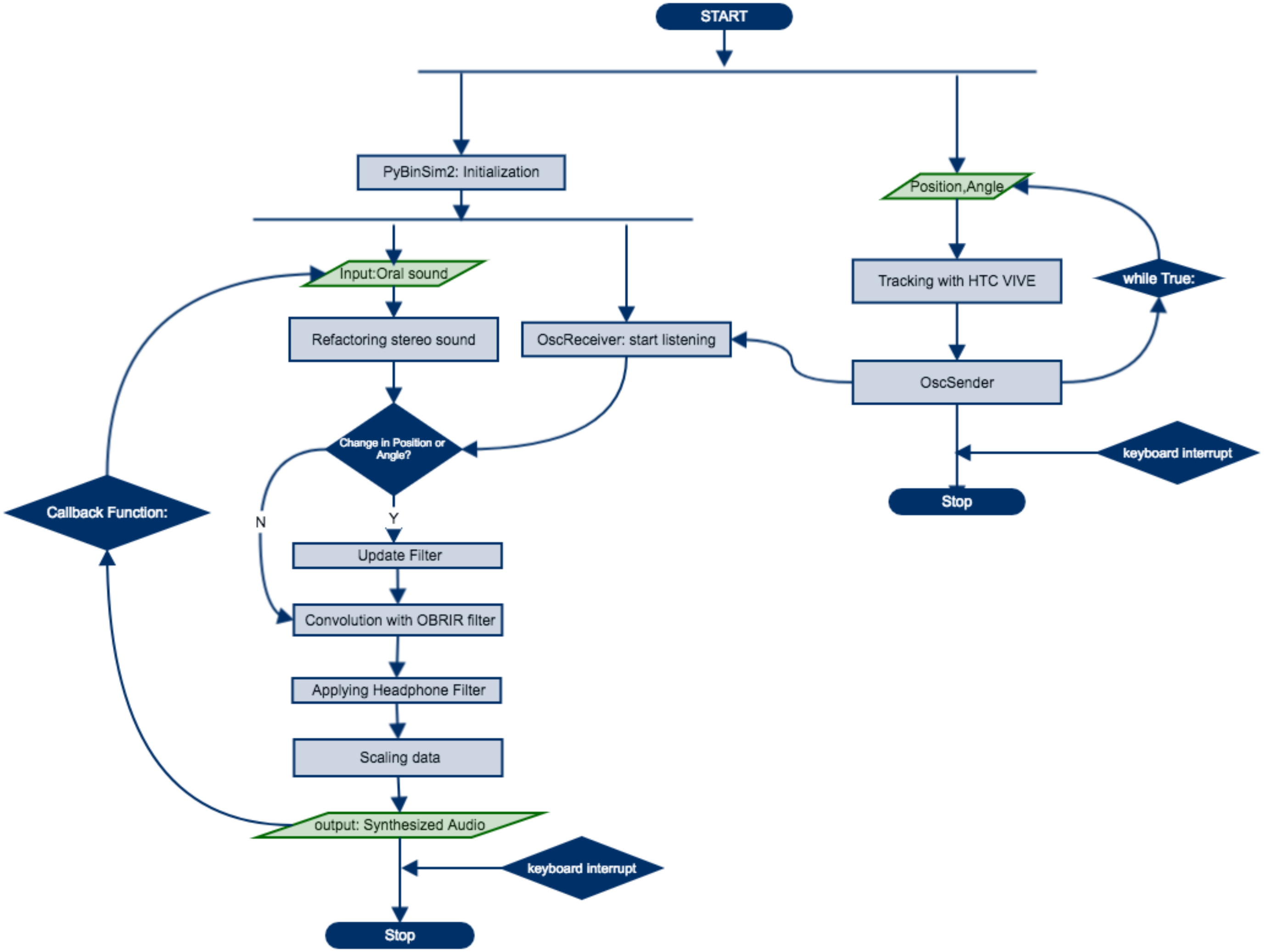


FIGURE 5.4: Pybinsim2

5.1.3 *Callback function*

Callback function in *portaudio* is the programming function which makes the treatment of live oral speech possible. Manipulation of pybinsim was mainly concentrated on callback function as many parameters of input signal as well as the convolution and filterstorage such as blocksize, filter length, number of channels and loudness can be accessed. A functional programming code, performing the desired task of processing real time audio input is available.

The program command for callback function is:

```
def callback(in_data, frame_count, time_info, status):
              in_data = np.fromstring(in_data, dtype=np.float32)
               audio_data = np.empty((2,binsim.blockSize))
              audio_data[0,:]=in_data[::2]
              audio_data[1,:]=in_data[1::2]

         # Get sound block. return buffer_content
        binsim.block= audio_data

        # Update Filters and run each convolver with the current block n=0,1
        for n in range(binsim.nChannels):

            # Get new Filter
            if binsim.oscReceiver.is_filter_update_necessary(n):
                  filterValueList = binsim.oscReceiver.get_current_values(n)
                  filter = binsim.filterStorage.get_filter
                  (Pose.from_filterValueList(filterValueList))
                  binsim.convolvers[n].setIR(filter,
                  callback.config.get(’enableCrossfading’))
                  left, right = binsim.convolvers[n].process(binsim.block[n, :])

            # Sum results from all convolvers
            if n == 0:
                binsim.result[:, 0] = left
                binsim.result[:, 1] = right
            else:
                binsim.result[:, 0] = np.add(binsim.result[:, 0],left)
                binsim.result[:, 1] = np.add(binsim.result[:, 1],right)

        # Finally apply Headphone Filter
        if callback.config.get(’useHeadphoneFilter’):
            binsim.result[:, 0], binsim.result[:, 1]
           = binsim.convolverHP.process(binsim.result)
        # Scale data
        #binsim.result = np.divide(binsim.result, float((binsim.nChannels()) * 2))
        binsim.result = np.multiply(binsim.result,callback.
        config.get('loudnessFactor'))
        return (binsim.result[:frame_count].tostring(), pyaudio.paContinue)
    callback.config = binsim.config
    return callback
```

* Variable input_data is the oral sound input what needs to be processed.

In the next section, we would talk about the different types of delays encountered during this project. The complete overview of the different delay measurements and the techniques used to do measurements is explained in the following section.

5.2 *Delay measurement*

The figure given below explains the presence of time delay, in the system.

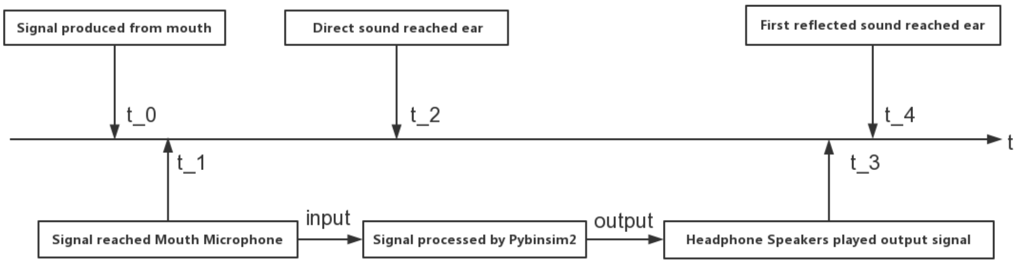

Signal produced from mouth
Direct sound reached ear
First reflected sound reached ear
t_0
t_2
t_4
t
t_1
t_3
input
output
Signal reached Mouth Microphone
Signal processed by Pybinsim2
Headphone Speakers played output signal

FIGURE 5.5: Time flow of different component

t_2 - t_0 = Direct sound delay

t_4 - t_0 = First reflected sound delay

t_2 - t_1 = Time difference between signal reached mouth microphone and ear

t_3 - t_1 = System delay

Creation of new OBRIR filters requires the delay data for the direct sound and the system delay data. The direct sound delay is the delay of sound from the mouth to ears. System delay comprises of the time taken by the sound from the near-mouth microphone to be produced in the headphone after first reflection.

5.2.1 *Direct Sound Delay Test* (t_2-t_0, t_2-t_1)

Direct sound delay is the time difference between the sound reaching the near-mouth microphone and the ears. For this purpose, the tester speaks or makes high frequency clicks from the mouth. The input signal is recorded. In total there are three input channels, channel one is the near-mouth microphone, channel two and three being the in ear microphone for left and right ears respectively.

The sampling frequency was set to 44100Hz and the block-size of 128 was considered. The distance between the mouth and the near mouth microphone was approximately 4 cm. The distance between dummy head mouth and the ears is about 17 cm. Keeping this in consideration.

According to the above settings, some modifications were made in *Audacity.* Buffer length of 100 ms was chosen and 32-bit float PCM was used, keeping the same sampling frequency i.e 44.1 KHz.

The description of the testing procedure for the delay data is explained further. The tester was asked to speak or make preferably the sharp oral clicks in the near-mouth microphone and the in ear microphones also sense the orally produced sound. All this data from production of oral sounds to being sensed by the in ear mics was recorded in computer with the help of *Audacity* software. From the recorded data the difference of samples between the direct sound and the sound to the respective ear was found.

Direct sound delay is directly proportional to the distance between mic and ears.

$$t_{directsound} = \frac{d_{mouth-to-ear}}{c} \tag{5.1}$$

where: d is the distance of near mouth microphone to ear and c is the speed of sound at room temperature.

Delay between signal from mouth microphone to ear:

Distance between mouth and ear and the distance between mouth and mouth microphone are in consideration for this delay.

$$t_{(mic\_to\_ear)} = \frac{d_{(mouth\_to\_ear)} - d_{(mouth\_to\_mic)}}{c} \tag{5.2}$$

Delay type	Delay (in samples)
Mic to ear	16
Mouth to ear	19

Table 5.1: Direct Sound Delay Data

Measurement	Unit
Mouth to ear distance	17 cm
Mouth to mic distance	4 cm approx.
Delay	22 samples

Table 5.2: Measurements for KEMAR

5.2.2 *System Delay Test* (t_3-t_1)

It is the near-mouth microphone to headphone delay. For finding out this delay, filters with zero delay were implemented. In reference to the flow chart 5.5, it is the time taken by sound from t_1 to t_4. All the settings were kept as the same as for the direct sound delay test. The first test in this process was to run the code i.e pybinsim2 without the use of tracking system. Then, consequently the code is run subsequently with the tracking system also in function. Lastly, the code is run without filtering function and also in absence of the tracking system.

The stated three tasks were performed in Windows as well as MAC operating system was also used to perform the first and the last task. For running the tracking device with collaboration with MAC operating system requires a specific high performance graphics card or eGPU. A MAC system with the desired specifications was not available for testing.

$$t_{(systemdelay)} = t_{(headphone-output)} - t_{(mic-input)} \tag{5.3}$$

Equation 5.3 represents the delay from microphone to headphone.

Sampling rate of 48000Hz was considered and the results are presented.

Delay factors	Windows 7	MacOS
Delay	8600	1200
Delay (in ms)	180 ms	25 ms
Testing software	same	same
Algorithm	same	same

TABLE 5.3: System Delay Data

MacOS	Delay(in samples)	Delay (in ms)
Input to output in Sound-card	40	0.833 ms
Delay without filtering	1200	25 ms
Delay with filtering	1215	25.33 ms

TABLE 5.4: Delay Data For Mac OS

The large difference of system delay between Mac and Windows operating systems shows that system delay is highly influenced by the host conditions. In the system, PortAudio acts as a software layer that maps between the PortAudio API and host API. As such, system delay mainly includes PortAudio buffering and host API buffering.

Figure 5.6 shows the operating systems and host API we used in the measurement PortAudio arranges for the host API to invoke a callback function to process audio in PortAudio buffers. This buffer size is initialised by fixing stream parameters and flags, such as the frames_per_buffer, paNonInterleaved flag indicates, data_format and channel in Pa_OpenStream( ) function. frames_per_buffer presents the number of frames passed to the stream callback function. It should be noticed that with some host APIs, the use of non-zero framesPerBuffer for a callback stream may

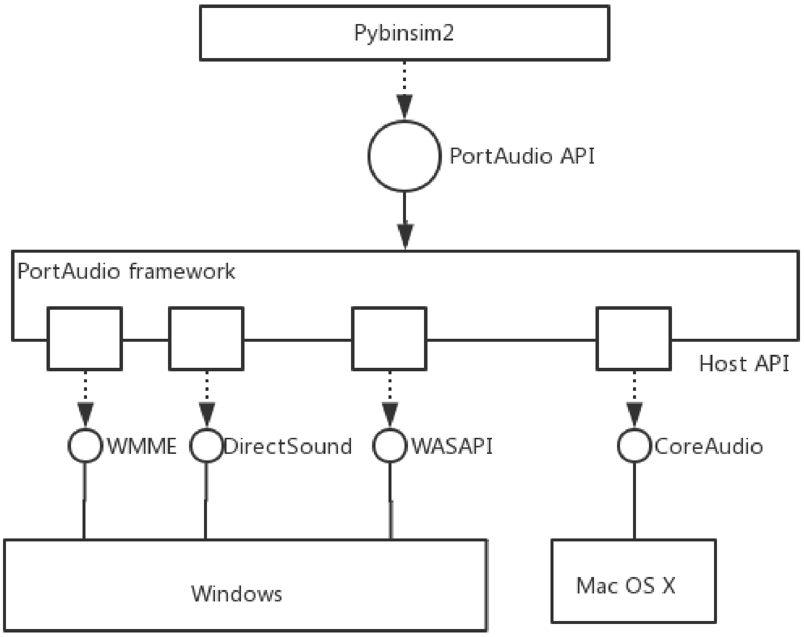
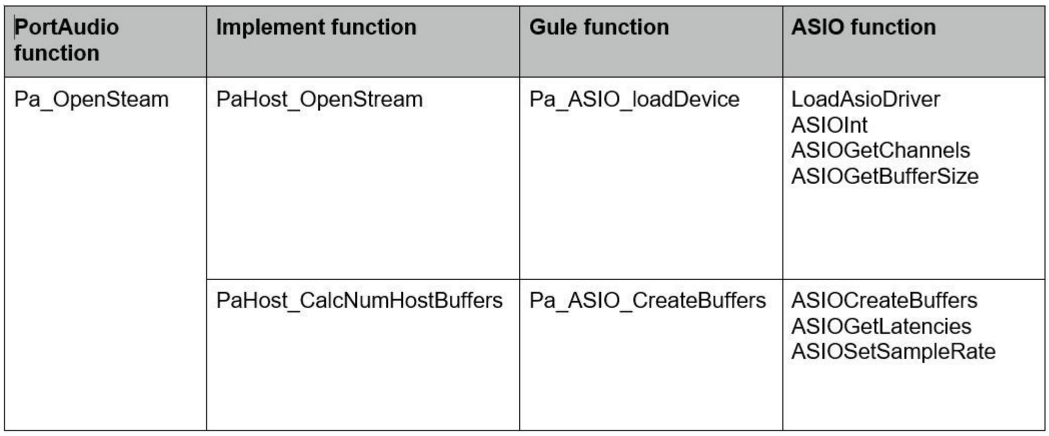


FIGURE 5.6: OS and Host APIs

introduce an additional layer of buffering which could introduce additional latency [56]. The paNonInterleaved flag indicates that audio data is passed as an array of pointers to separate buffers, one buffer for each channel. Usually, when this flag is not used, audio data is passed as a single buffer with all channels interleaved.

The initialised "mapping" between PortAudio functions and host API functions can be described by following table, using PortAudio to ASIO to describe:



PortAudio function	Implement function	Gule function	ASIO function
Pa_OpenSteam	PaHost_OpenStream	Pa_ASIO_loadDevice	LoadAsioDriver ASIOInt ASIOGetChannels ASIOGetBufferSize
	PaHost_CalcNumHostBuffers	Pa_ASIO_CreateBuffers	ASIOCreateBuffers ASIOGetLatencies ASIOSetSampleRate

FIGURE 5.7: Mapping in PortAudio and APIs

After the initialisation step, the host API will basically do the following operations in a full-duplex case:

1. Transfer samples from the host API input buffer to the PortAudio API input buffer.

2. If the PortAudio input buffer is full, call the PortAudio callback which will produce a PortAudio output buffer.

3. Transfer samples from the PortAudio output buffer into the host API output buffer.

These operations may be done several time depending of the host API and PortAudio API buffer sizes that are used. At each host API callback, all samples from the host input buffer must be consumed and a complete host output buffer has to be produced [57].

In pybinsim2,
channel= 2
format= paflot32
frames_per_buffer= 128
sampling_rate= 48000
Then, size of per sample= 32bits,
Size of per frame= channel x sample_size= 64bits,
Size of buffer= frames_per_buffer x frame_size= 8Kbits.

Then, host API function creates host API buffer size, if the global requested PortAudio buffer size is inside the minimum/maximum host size range, assuming host buffer size = requested PortAudio buffer size. Else, host buffer size = minimum host size.

Then, the minimum of total latency = host API input latency + host API output latency + PortAudio buffering latency + ADC latency + DAC latency [58].

In MAC OS 'X' system, CoreAudio is the host API for PortAudio. The audio data format is LinearPCM and in an interleaved structure for two channel signal. The minimum/maximum host size range is 0x4000 0̃x10000, which is 16Kbits 6̃4Kbits.
Then, host API input buffer size = host API output buffer size = 16Kbits
Total buffer size = 2 x 16Kbits + 8Kbits = 40Kbits
Number of frames per total buffer = total buffer size/frame size = 640
Number of samples per total buffer=1280
Plus ADC and DAC 40 samples, the system delay= 1320 samples.
This value is closed to the result from measurement in MAC system. The large latency is an issue in implementation. Other possible approves in total latency are discussed in conclusion chapter.

5.2.3 *Creation of new filters*

This section explains the creation of new OBRIR filters based on the measurement data accumulated through the previous two delay tests. The original set of OBRIRs were measured as explained by *Neidhardt et al [1].*

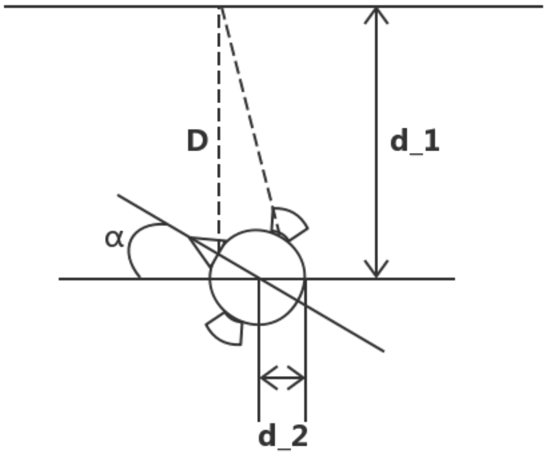


FIGURE 5.8: Distance between the center of the head and wall(d_1), Radius of the head(d_2), Orientation of the head(α), Total path of sound(D)

The fig. 5.8 shows the total distance, the sound from microphone to strike the virtual wall and the first reflection to be heard in the headphone. The formulation for computing the total distance is given in the following equation.

$$D = d_1 \,-\, d_2\, sin\,\alpha \,+\, \sqrt{[(d_2\, sin\,\alpha \,+\, cos\,\alpha)^2 \,+\, (d_1 \,-\, d_2\, cos\,\alpha)^2]} \tag{5.4}$$

where:

D is the total distance of the sound travelling from the mouth to the virtual wall and coming back to the ears after first reflection.

d_1 is the distance between the center of the head and the virtual wall.

d_2 is the radius of the head.

d_1 can be chosen as 25cm, 50cm, 75cm, 100cm, 125cm, 150cm, 175cm, 200cm.

d_2 is 8 cm.

Angle, α is 0° to 90°

Hence, when $D = D_{min}$ then $\alpha = 45^\circ$. With a resolution of 4°, the shortest distance is in 136°. With the assumption that the system delay is smaller than the first reflected sound delay, the new delay Δn is measured. It presents the number of samples set to zero in front of reflected sound impulse response. The first sample taken from the original OBRIRs is defined by equation 5.5 below.

$$n_{start} \,=\, fs\, \frac{D_{(min)} \,-\, d_{eartomouth}}{c} \tag{5.5}$$

where:

n_{start} is the start index of the samples.

$d_{eartomouth}$ is time of direct sound reaching the ear.

D_{min} is the minimum distance the sound travels from the mouth back to the ears after reflection.

c is the speed of sound at room temperature.

The new delay samples were set to zero in order to make new filters. Hence, the new oral binaural room impulse responses (OBRIRs) are created by addition of the delay samples acquired and the first reflected sound.

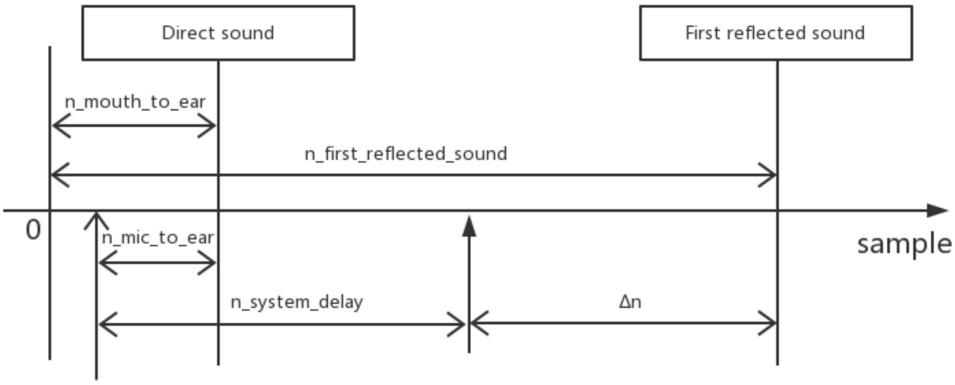


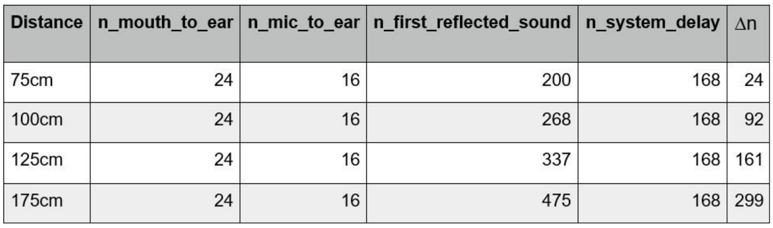
FIGURE 5.9: Scheme for creation of new filters

Fig 5.9 shows the algorithm to generate new filters. Δn is the number of samples have to be set to zero, and it's value should not smaller than 0.

$$\Delta\, n = n_{first\,reflected\,sound} - n_{mouth\,to\,ear} - n_{system\,delay} + n_{mic\,to\,ear} \tag{5.6}$$

The large system delay is an issue due to which Δn cannot be found using the available OBRIRs. An ideal system delay of 168 samples (at fs = 48000Hz) is set-up to do a pre listening test. The total buffer size is assumed to be 128 samples plus 40 samples from A/DC buffer size.

According to the algorithm, the Δn of new filters is showed in the following table:



Distance	n_mouth_to_ear	n_mic_to_ear	n_first_reflected_sound	n_system_delay	Δn
75cm	24	16	200	168	24
100cm	24	16	268	168	92
125cm	24	16	337	168	161
175cm	24	16	475	168	299

FIGURE 5.10: Measurements of delay samples for 75cm, 100cm, 125cm,175cm

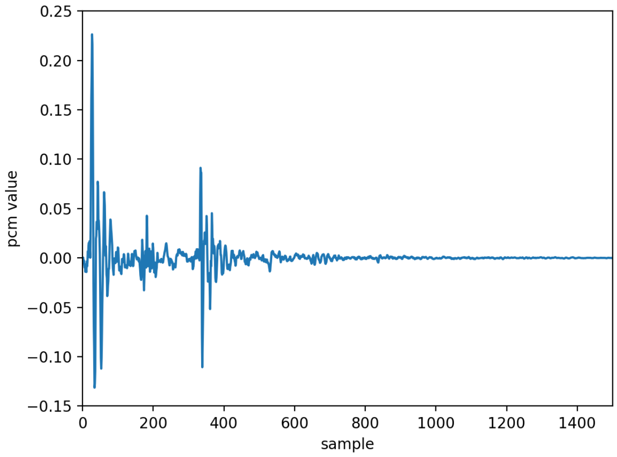


Figure 5.11: Original OBRIR for 125cm, α, right channel

Figure 5.11 shows the OBRIR for the right channel. The wall is simulated at the distance of 125cm. The angle of orientation is $\alpha = 136°$.

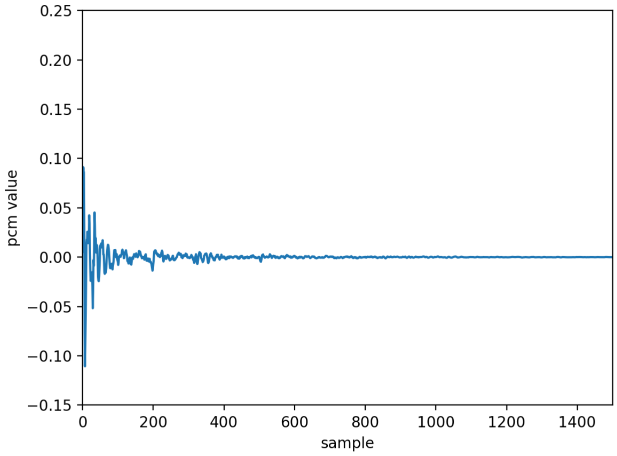


Figure 5.12: Reflected sound impulse response

The impulse response of the sound after reflection is shown in the figure 5.12

The new filters were developed using the measurements and the plot is given in figure 5.13.

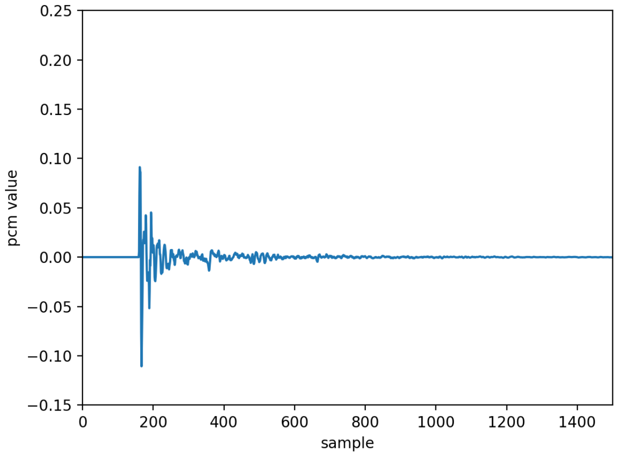


FIGURE 5.13: New OBRIR for 125cm, $\alpha = 136°$, right channel

5.3 Pre-listening test: Own sound vs. Recorded sound

Due to the limitation caused by large latency, the new filters were created with the assumption that the ideal system delay is 168 samples and sampling frequency was considered 48000Hz. This value of system delay has been assumed because the block size of the system is 128 samples and there is 0.4 ms of delay of the sound-card. So, 40 + 128 = 168 samples of system delay is chosen to be ideal.

A listening test was designed to investigate the capability of the testing listeners to perceive and localize a virtual wall by self-produced sounds. Also the system with the recorded sounds was tested and the results were compared.

5.3.1 Listening test setup

A HTC Vive scaled room setup was used for the experiment. The participant was equipped with head mounted display for tracking the position and head orientation. A blue grid marks the boundary of the virtual environment. Some cues were also provided on the floor for walking around.

For the system using recorded sounds, an 8 minutes long excerpt of dry male speech reading an audio book was chosen as the test signal. For the audio reproduction, a RME Fireface UCX sound card and AKG K1000 headphones with headphone compensation filter were used. The OBRIRs were dynamically applied to the oral sound using the original version of pybinsim.

For the system using self-produced sound, a Sennheiser mouth microphone was used for capturing oral sound. The participant had to wear the mouth microphone about 7 cm from the centre of the lips.

Two persons, both of them females, took part in this pre-test. The participants were aged 25 and 29 respectively. Both stated to have normal hearing abilities.

5.3.2 *Test procedure*

The room with 1-wall-scene, recorded in the anechoic chamber was used to test two systems. One for recorded sounds, other for self-produced sounds.

Before each pre-test, a short training test was taken.

Training part: Walking towards the virtual wall.

During the training part, the listener was told the direction of the wall. Listener could walk along a given line with a length of 2m and turn around in all directions. With the training part, listener could find differences and get familiar with acoustical cues a wall brings along.

Testing part: Estimating the direction of the virtual wall.

During the test, the listener was asked to turn around until he/she thought to be facing the wall. The participant indicated, the current direction which he/she recons is the wall. This answer marked the tracking data for a particular participant.

The distances of 75cm, 100cm, 125cm, and 150cm were chosen for test and each of those distances was asked with 4 random rotations 0°, 92°, 180°, 268°. Thus, for each system, 16 assignments had to be done.

5.3.3 *Performance*

The estimated direction of the virtual wall as well as the resulting error in azimuth was captured as a multiple of 4°, because the directional filters were only available in 4° steps.

Fig.5.14 provides an overview of the average error for all test participants. The results are for different distances and systems, considering only an azimuth angle of 0° as correct.

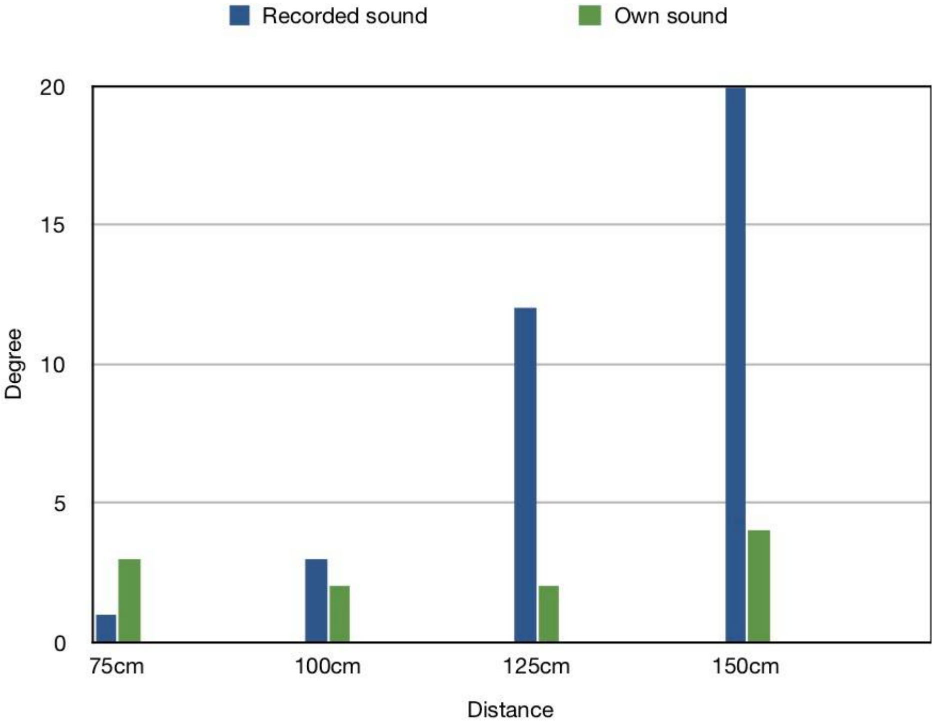


Figure 5.14: Errors: own voice vs recorded sound.

The average errors observed for the 125cm and 150cm distance in the system using recorded sound are much higher than the system using self-procedure sound, which shows the performance of using own sound had a better perception even with large latency.

The differences between the performance of two systems could be seen during the testing parts of each of them: it took the same time for listeners to find the direction of the wall in 75cm and 100cm. But in 125cm and 150cm participants spent more time using recorded sound than their own sound. And the feedback from the listeners was that they were more concentrated on their own sound.

For the actual comparison between real time self produced oral sounds and recorded voice, this experiment is not successful. For concrete results, an additional improvement of system delay is awaited.

Chapter 6

Conclusions and Future Works

In this chapter a summary about the study is presented. Different problems confronted and the solutions have been discussed. This section also gives an overview of the highlights of this project and also suggests the steps that can be pivotal to achieve very low latency.

6.1 *Conclusions*

It has been indicated by this report that the perception of virtual room might be better when a system for self produced oral sounds is employed. Though the results were not favourable to conduct an official listening test, still the data accumulated is useful for future accomplishments.

There are many factors affecting the delay in the system. These are mainly hardware related issues but changes in the software approach might also increase the success chances. On the basis of the results and experiences, the problems with the signal flow in the system can be seen. Moreover, code related latency is also not ruled out. The time delay of 1 ms for a wall simulated at a distance of 25 cm is required by the sound produced in the headphone. System delay is the restrictive factor. Increase in the delay time reduces the participants involvement and focus.

Pybinsim is manipulated to process real time audio. This achievement paves the path for future trials with various techniques. Callback function was developed and implemented in order to continuously update the user data. Headphone compensation filters were determined from the measurements to accommodate the stereo output through the headphone.

The direct sound delay test resulted in the conclusion that the bigger the distance between the mouth and the ears of the same head, the higher the delay. So, the time taken by sound from mouth reaching the ears is different for different people. In system delay test it was noted that the software influenced the delays in different computational machines. The delay difference of approximately 7400 samples between a mac and windows OS was found.

The new OBRIR filters were created in order to balance the audio input to output process. For this purpose, the delay between the direct sound and its first pure reflection after striking a surface was considered.

One point of discussion is that the distance between mouth and the near mouth microphone was not constant during the tests. The better option could be the use of hand-held microphone for increased user convenience.

Background noise is always present during the course of experiment. It is more sensible to conduct the tests in quiet surroundings with minimum human presence. Background noise can also disturb the concentration of the tester during the experiment.

It was noticed that Windows 7 operating system supports only two simultaneous channel recording. It is not possible to connect three input devices through an external sound-card and record side by side.This is because the ”Audio Host” is the interface between Audacity and the sound device. On Windows, the choice is between the following audio interfaces.

MME: This is the Audacity default and the most compatible with all audio devices. Windows DirectSound: This is more recent than MME with potentially less latency. Windows WASAPI: This host is the most recent Windows interface, that

Audacity supports, between applications (such as Audacity) and the soundcard driver. WASAPI was first officially released in 2007 in Windows Vista. WASAPI is particularly useful for "loopback" devices for recording computer playback. 24-bit recording devices are supported. Playback is emulated using this host [59]. So accordingly, the delays were measured by switching the channel inputs between the left and the right in-ear microphones with mouth microphone respectively.

The dummy head (HATS) used for the delay test purposes had the same measure of delay from mouth to the left and right ears respectively. Sharp clicking and tapping sounds were made near the mouth of KEMAR in order to get the delay measurements. One or two samples were neglected during the delay data accumulations in the case of humans but exact readings were recorded with dummy head.

6.2 *Future works*

One approach can be to reduce the system delay by using C++ programming language instead of python. According to [60], C++ is upto three times faster than program written in python. Also [61] gives the comparison of different programming languages. It shows that use of C family languages is much more feasible than python.

In Mac OS, the delay of 1200 samples is present, this delay might be reduced by use of C family language and bring it closer to acceptance. Use of Mac OS requires eGPU [62] installed in the computer in order to operate HTC vive with it. eGPU is an external graphics card which boosts the performance of the computer. So consequently it can handle the heavy application programs such as Steam VR etc.

Use of real time convolvers such as MSP430 [63] of different generations from Texas Instruments might be helpful. Real time convolvers are the external micro-controller based devices used for low powered embedded systems. These devices can be used to perform the task of convolution externally by doing some alterations to its basic mechanism to support the needed specifications. Hence, neutralizing the computational burden on the computers. The computer may be used just to record and control the tracking data, saving a lot of memory to perform the assigned task. The external convolver can perform the convolution between the input audio and the OBRIR filters and send the output to the headphones via computer. The computer can easily take charge of the tracking data. The desired results might be achieved.

A software based VST convolver [47] may also be tried. [46] also makes use of one such system for convolution of own voice with OBRIRs in the real room surroundings. An online forum [64] suggests different techniques of convolution. One such technique of dividing input data into small chunks and applying convolution on each chunk has already been used by the authors. This forum also mentions `"FFTConvolver"` for short convolutions, which incurs a latency of 0.5 times per sample. But `"ReverbConvolver"` in the same website is said to be suited for performing extremely long real-time convolutions on a single audio channel. It uses multiple `"FFTConvolver"` objects as well as an input buffer and an accumulation buffer. It has been stated there that it possible to get a multi-threaded implementation in performing convolution by utilizing the parallelism technique. Another claim to be noted is that theoraticaly it is said to be possible to get zero latency if the very first `"FFTConvolver"` is replaced with a `"DirectConvolver"` (without using a FFT). An attempt could also be made in this direction.

Bibliography

[1] Annika Neidhardt, Janine Liebal, and Juhani Paasonen. Human echolocation in virtual acoustic environments: Estimating the direction of a close wall.

[2] Co-ordinate system. URL `http://www.tonmeister.ca/main/textbook/intro_to_sound_recording487x.png`. [accessed April 3, 2018].

[3] Room impulse response. URL `http://intarch.ac.uk/journal/issue44/12/tof.html`. [accessed April 8, 2018].

[4] Drr. URL `https://m.eet.com/media/1109853/acoustics_and_psychoacoustics_applied_fig28t.jpg`. [accessed April 8, 2018].

[5] Kemar 45b.a. URL `http://www.aimil.com/products/kemar-manikins`. [accessed April 11, 2018].

[6] Manuj Yadav, Densil Cabrera, Ralph Collins, and William L Martens. Detection of headtracking in room acoustic simulations for ones own voice. In *Proceedings of the Australian Acoustical Society Conference*, 2011.

[7] Online. URL `https://www.roadtovr.com/wp-content/uploads/2016/01/htc-vive-pre-system.jpg`. [accessed May 13, 2018].

[8] Online. URL `https://support.steampowered.com/steamvr/HTC_Vive/images/vive_installer_step10.png`. [accessed May 16, 2018].

[9] Grigore C Burdea and Philippe Coiffet. *Virtual reality technology*, volume 1. John Wiley & Sons, 2003.

[10] Jonas Brunskog, Anders Christian Gade, Gaspar Payá Bellester, and Lilian Reig Calbo. Increase in voice level and speaker comfort in lecture rooms. *The Journal of the Acoustical Society of America*, 125(4):2072–2082, 2009.

[11] Manuj Yadav, Luis Miranda, Densil A Cabrera, and William L Martens. Simulating autophony with auralized oral-binaural room impulse responses. In *Audio Engineering Society Convention 133*. Audio Engineering Society, 2012.

[12] Annika Neidhardt, Florian Klein, Niklas Knoop, and Thomas Köllmer. Flexible python tool for dynamic binaural synthesis applications. In *Audio Engineering Society Convention 142*. Audio Engineering Society, 2017.

[13] Jyri Huopaniemi, Lauri Savioja, Tapio Lokki, and Riitta Vaananen. Virtual acoustics applications and technology trends. In *Signal Processing Conference, 2000 10th European*, pages 1–8. IEEE, 2000.

[14] Christoph Pörschmann. One's own voice in auditory virtual environments. *Acta Acustica united with Acustica*, 87(3):378–388, 2001.

[15] Lore Thaler. Echolocation may have real-life advantages for blind people: an analysis of survey data. *Frontiers in physiology*, 4:98, 2013.

[16] Leo H Riley, David M Luterman, and Marion F Cohen. Relationship between hearing ability and mobility in a blinded adult-population. *New Outlook for the Blind*, 58(5):139–141, 1964.

[17] Ira Kohler. Orientation by aural clues. *Res. Bull. Am. Found. Blind No. 4*, pages 14–53, 1964.

[18] Thomas A Stroffregen and John B Pittenger. Human echolocation as a basic form of perception and action. *Ecological psychology*, 7(3):181–216, 1995.

[19] H Fastl and E Zwicker. Psychoacoustics: Facts and models. springer series in information sciences. *Springer*, 2007.

[20] Alexander Lindau, Torben Hohn, and Stefan Weinzierl. Binaural resynthesis for comparative studies of acoustical environments. In *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.

[21] Jie Huang, N Ohnishi, and Noboru Sugie. Spatial localization of sound sources: azimuth and elevation estimation. In *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, volume 1, pages 330–333. IEEE, 1998.

[22] Jens Blauert. *Spatial hearing: the psychophysics of human sound localization*. MIT press, 1997.

[23] Mark B Gardner and Robert S Gardner. Problem of localization in the median plane: effect of pinnae cavity occlusion. *The Journal of the Acoustical Society of America*, 53(2):400–408, 1973.

[24] Alan D Musicant and Robert A Butler. The influence of pinnae-based spectral cues on sound localization. *The Journal of the Acoustical Society of America*, 75(4):1195–1200, 1984.

[25] William A Yost. Pitch strength of iterated rippled noise. *The Journal of the Acoustical Society of America*, 100(5):3329–3335, 1996.

[26] Bo N Schenkman and Mats E Nilsson. Human echolocation- blind and sighted persons ability to detect sounds recorded in the presence of a reflecting object. 39(4):483–501, 2010.

[27] Daniel H Ashmead and Robert S Wall. Auditory perception of walls via spectral variations in the ambient sound field. *Journal of Rehabilitation Research and Development*, 36(4):313–322, 1999.

[28] J Blauert, D Kolossa, K Obermayer, and K Adiloğlu. Further challenges and the road ahead. In *The technology of binaural listening*, pages 477–501. Springer, 2013.

[29] Stephan Werner and Simone Füg. Controlled auditory distance perception using binaural headphone reproduction–evaluation via listening tests. In *in*

Proceedings of the 27 th Tonmeistertagung, VDT International Convention, Cologne, 2012.

[30] Eleftheria Georganti, John Mourjopoulos, and Steven van de Par. Room statistics and direct-to-reverberant ratio estimation from dual-channel signals. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4713–4717. IEEE, 2014.

[31] Jamie Angus and David Howard. *Acoustics and psychoacoustics*. Focal press, 2013.

[32] Tilen Potisk. Head-related transfer function. In *Seminar Ia, Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia*, 2015.

[33] Benedikt Grothe, Michael Pecka, and David McAlpine. Mechanisms of sound localization in mammals. *Physiological reviews*, 90(3):983–1012, 2010.

[34] Andrew Wabnitz, Nicolas Epain, Craig Jin, and André Van Schaik. Room acoustics simulation for multichannel microphone arrays. In *Proceedings of the International Symposium on Room Acoustics*, pages 1–6. Citeseer, 2010.

[35] Graham M Naylor. Odeonanother hybrid room acoustical model. *Applied Acoustics*, 38(2-4):131–143, 1993.

[36] Densil Cabrera, Hayato Sato, William L Martens, and Doheon Lee. Binaural measurement and simulation of the room acoustical response from a person's mouth to their ears. *Acoustics Australia*, 37(3), 2009.

[37] Pauli Minnaar, Soren Krarup Olesen, Flemming Christensen, and Henrik Moller. The importance of head movements for binaural room synthesis. Georgia Institute of Technology, 2001.

[38] Philip Mackensen, Markus Fruhmann, Mathias Thanner, Günther Theile, Ulrich Horbach, and Attila Karamustafaoglu. Head tracker-based auralization systems: Additional consideration of vertical head movements. In *Audio Engineering Society Convention 108*. Audio Engineering Society, 2000.

[39] Pavel Zahorik and Frederic L Wightman. Loudness constancy with varying sound source distance. *Nature neuroscience*, 4(1):78, 2001.

[40] Andrew J Kolarik, Silvia Cirstea, Shahina Pardhan, and Brian CJ Moore. A summary of research investigating echolocation abilities of blind and sighted humans. *Hearing research*, 310:60–68, 2014.

[41] Andrew J Kolarik, Amy C Scarfe, Brian CJ Moore, and Shahina Pardhan. Blindness enhances auditory obstacle circumvention: Assessing echolocation, sensory substitution, and visual-based navigation. *PloS one*, 12(4):e0175750, 2017.

[42] Nadia Lessard, Michael Paré, Franco Lepore, and Maryse Lassonde. Early-blind human subjects localize sound sources better than sighted subjects. *Nature*, 395(6699):278, 1998.

[43] Ludwig Wallmeier, Nikodemus Geßele, and Lutz Wiegrebe. Echolocation versus echo suppression in humans. *Proceedings of the Royal Society of London B: Biological Sciences*, 280(1769):20131428, 2013.

[44] Bo N Schenkman and Mats E Nilsson. Human echolocation: Blind and sighted persons' ability to detect sounds recorded in the presence of a reflecting object. *Perception*, 39(4):483–501, 2010.

[45] Daniel Rowan, Timos Papadopoulos, David Edwards, Hannah Holmes, Anna Hollingdale, Leah Evans, and Robert Allen. Identification of the lateral position of a virtual object based on echoes by humans. *Hearing research*, 300: 56–65, 2013.

[46] Manuj Yadav, Densil Cabrera, and William L Martens. A system for simulating room acoustical environments for ones own voice. *Applied Acoustics*, 73(4):409–414, 2012.

[47] VST plugin SIR2. URL `http://www.knufinke.de/sir/sir2.html`.

[48] Max/MSP. URL `http://cycling74.com/`.

[49] David Pelegrin Garcia, Monika Rychtáriková, Christ Glorieux, and Brian FG Katz. Interactive auralization of self-generated oral sounds in virtual acoustic environments for research in human echolocation. In *Proceedings of Forum Acusticum 2014*, 2014.

[50] Fireface ucx. URL `http://www.rme-audio.de/en/products/fireface_ucx.php`. [accessed May 22, 2018].

[51] Online. URL `https://www.wareable.com/vr/htc-vive-review`. [accessed May 10, 2018].

[52] Nicolas La Rocco. Virtual reality: Vr-brille von htc und valve aus der nhe betrachtet. URL `https://www.computerbase.de/2015-03/htc-vive-vr-brille-hands-on/`. [accessed May 10, 2018].

[53] Steam support. Htc vive installation guide. URL `https://support.steampowered.com/steamvr/HTC_Vive/`. [accessed May 12, 2018].

[54] Audacity. About audacity, . URL `https://www.audacityteam.org/about/`. [accessed April 17, 2018].

[55] Adobe audition cc. URL `https://www.adobe.com/products/audition.html`. [accessed May 2, 2018].

[56] Portaudio. URL `http://www.portaudio.com/`.

[57] Stephane Letz. Porting portaudio api on asio. *GRAME-Computer Music Research Lab. Technical Note-01-11-06*, 2001.

[58] Pulse code modulator. URL `http://dictionnaire.sensagent.leparisien.fr/Linear%20Pulse%20Code%20Modulation/en-en/`.

[59] Audacity. Audacity manual, . URL `https://manual.audacityteam.org/man/tutorial_selecting_your_recording_device.html`. [accessed April 17, 2018].

[60] C++. URL `http://www.cplusplus.com/forum/general/159583/`. [accessed June 4, 2018].

[61] Comparison between different programming languages. URL https://thenewstack.io/which-programming-languages-use-the-least-electricity/.

[62] egpu: Apple support. URL https://support.apple.com/en-us/HT208544.

[63] Texas instruments: Msp340- ultra low power mcus. URL http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/applications.html.

[64] Web reverb. URL https://www.w3.org/TR/2013/WD-webaudio-20131010/convolution.html.

Contributions and Responsibilities

This project is a collective effort of all the group members. Each member provided assistance in every aspect of this work.

- Shuang Wang, Matr. No: 54764, was responsible for programming in python to modify *Pybinsim*. She also offered her assistance in implementation.
- Chenyao Diao, Matr. No: 58484, was mainly assigned the implementation task. She provided valuable inputs in the programming and writing as well.
- Manan Lamba, Matr. No: 56658, was responsible for the documentation and writing the report. He assisted in the implementation tasks as well as testing.

Signed:

Shuang Wang: ____________________

Chenyao Diao: ____________________

Manan Lamba: ____________________