Design Review 4

Project report

Flagstaff, AZ
April 12, 2019

Edited by

Hangdi Hu, Anjun Zhang, Maximillian Jones, Stuart J.
Northern Arizona University

Contents

1 Introduction

We are team Virtual 3D Audio, and our client is Jennifer Johnson at COBHAM. Our primary goal is to emulate 3D audio using stereo headphones, for use in helicopter headsets.
Basically, by emulating 3D audio, it can be made possible to easily tell where the person who is speaking to you is, in reference to your own position. This is something we will hopefully be using with people within the same cabin, and possibly even with other helicopters in the same area.
Our team consists of four electrical engineering seniors. Two of us focus on hardware oriented design, while the other two focus primarily on software design.

1.1 Objectives - Maximillian

1. Create a program that allows for the emulation of 3D audio using stereo headphones

2. This program should allow us to choose a location, and play a sound that comes from that point relative to the listener

3. If the other objectives are completed, implement a multi-user option

4. If time allows, implement this design on standalone hardware

2 Problem Description

2.1 Motivation - Maximillian

The motivation for the project is as follows:
Create an algorithm that can emulate 3D audio for use in helicopters. This will be used in cabin, as well as for helicopters in the same general area.

2.2 Constraints - Hangdi

HRTF constraints: HRTF can not work well for everyone since the physical hearing systems are different among persons, and some listeners may hear the positional sound more clearly than others. Therefore, lots of research and tests are needed to do to improve our HRTF code, but we do not have that much time due to time constraints.
Time constraints: Researching background knowledge and methods to realize HRTF took us much time, and the integration of HRTF and GUI was completed around April 11th. The time to deliver project is before April 26th, so there is little time left for us to test and improve the whole system.
Hardware constraints: Purchasing hardware devices would cost at least one week, thus unfortunately, hardware implementation will have to be cancelled due to time constraints. Additionally, hardware choosing could be a big problem since this is a real-time audio processing system, we need to consider how much power and speed the processor requires, but all these factors depend on the completion of PC based system.

3 Subsystems - Stuart

While we had initially planned to have 3 subsystems, we unfortunately did not have enough time to meet our stretch goal of implementing our design on hardware. Due to this, we have only two subsystems - Our Head Related Transfer Function software, and our GUI.

3.1 HRTF Algorithm

The head related transfer function (HRTF) algorithm works via capturing Head Related Impulse Responses (HRIR), which are measured from inside a listeners ear at different points. These are used to generate HRTFs, which can then be used in tandem with an audio signal to emulate audio from any given point. This is done via convolving the given audio signal with the HRTF data for that specific point in space. This means that using a database of HRTFs, we are able to emulate 3d audio from any point, for any given audio. In our case, this was used for audio based on a real time microphone input, in order to properly capture the scope of the project. Basically, this system is what accounts for the processing done to the audio, and is what actually does the emulating of the 3D audio.

3.2 GUI

The GUI was made using MATLABs built in GUI creation. This GUI is what allows for the user to interface with our HRTF algorithm, and was designed to be as user friendly as possible. We designed it with the thought in mind that it could possibly be used by people with no technical background. Basically, this is what connects the end-user to the actual audio processing algorithm. See Figure 1 for example.

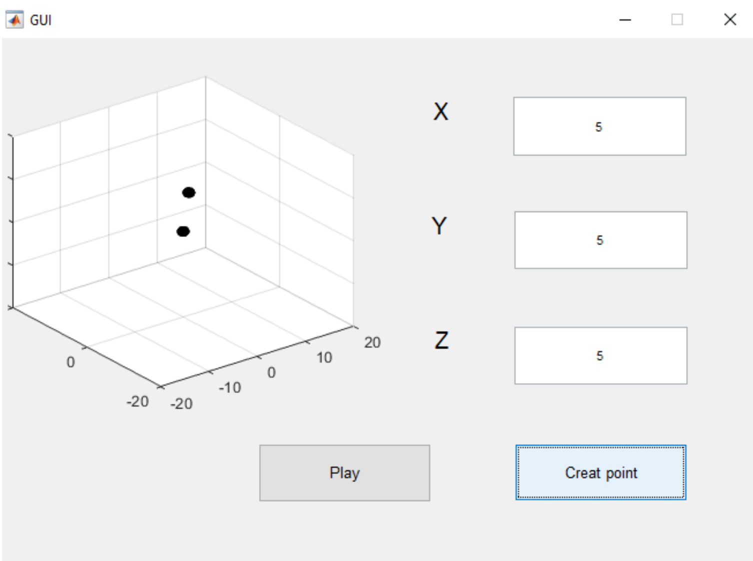

Figure 1: Sample Image of GUI. Two points graphed represent head, and point from which sound will come.

4 Work Breakdown Structures

4.1 Audio Processing - Stuart

The audio processing WBS refers to the entirety of the algorithm used to emulate the 3D audio. This was eventually split into two WBS's, once the first was completed. The first WBS (Figure 2) begins with activity 1.1, which was completed via a design decision based on conversation with our client. Our solution was to simply decide between three different sample data sets, which should allow for anybody to find one that more or less fits their physical specifics.
The next task was to add this new method to the HRTF design, which ended up being a non factor. This was actually done in the GUI, as it must allow for selection between different data sets. This data set is

then passed into the HRTF algorithm.
Task 2.1 also became a non factor, due to the fact that we unfortunately ran out of time to port our design onto hardware. Regardless, it was designed with this in mind, by being designed to work as quickly as possible on our given hardware, so that when implemented in a standalone device the delays would still be reasonable.

The second WBS (Figure 3) describes the next steps which were taken to complete the audio processing block of our design. The first task was to enable these filters to work in real time, which ended up becoming the most difficult task. This required the majority of the time spent on the tasks included in this WBS, and was solved primarily via research into different MATLAB programs which analyze real time audio.
Task 1.2 describes implementing our solution to 1.1 from the previous WBS, basically this just meant that the audio processing block had to be able to take a select input which would determine which sample set was used. This ended up being simple.
Task 2.1 ended up causing trouble, as implementing the audio processing with the GUI meant that the outputs of the GUI had to be edited to match those of the processing block. The audio processor initially took in azimuth and elevation inputs, but in order to increase ease of use of the GUI, we decided to instead use X, Y, and Z coordinates, and then convert them to azimuth and elevation coordinates inside the processing block.

Person Primarily Responsible: Stuart Jackson				
ID	**ACTIVITY/TASK**	**DESCRIPTION**	**DELIVERABLE(S)**	**OTHER PEOPLE**
1	**HRTF IMPROVEMENTS**			
1.1	Improve 3d emulation in elevation plane	Improve accuracy in elevation plane of HRTF implementation	Working method to accurately simulate audio in the elevation plane	Research help – all
1.2	Add new HRTF methods to current design	Synthesize MATLAB code based on improvements discovered in 1.1	Working MATLAB code that can generate accurate sounds in both azimuth and elevation planes	Ensure ability to integrate with GUI – Anjun and Hangdi
2	**Portability**			
2.1	Ensure code can be easily ported to other environments	The code may have to be remade in a new environment / language, ensure it does not rely heavily on MATLAB specific plugins	Code that can be easily transferred between environments, ideally using as few MATLAB toolboxes as possible	Requires knowledge of what hardware will be used - Max

Figure 2: WBS 1 for Audio Processing

4.2 GUI - Hangdi

The GUI allows users to input three dimensional data (x, y, z) to select desired directions, and by clicking the create button, the data can be passed to a plot function to plot the input and original point in the three dimensional coordinates system. When clicking the play button, the program will record sound in surrounding environment for 5 seconds and then play the recorded sound in the selected direction. See figure 4 for referenced WBS.

Person Primarily Responsible: Stuart Jackson				
ID	**ACTIVITY/TASK**	**DESCRIPTION**	**DELIVERABLE(S)**	**OTHER PEOPLE**
1	**Audio Processing Improvements**			
1.1	Enable HRTF filters in real time	Implement filter in real time upon microphone input	Working function which records, filters, and replays audio	None
1.2	Create ability to pick HRIR sample set	Ensure that sample set picked in GUI is selected and used for HRIR set	MATLAB code that can select between different sets of data for use in filtering	Ensure ability to integrate with GUI – Anjun and Hangdi
2	**Integration**			
2.1	Ensure code is written with later integration with GUI in mind	Allow for selection of different locations, as well as different HRIR sets	MATLAB code that allows for changes in azimuth, elevation, HRIR data set	Ensure ability to integrate with GUI – Anjun and Hangdi

Figure 3: Audio Processing WBS 2

4.3 Integration - Anjun

After we finished the GUI design and HTRF audio processing job, we needed to integrate those two parts together into an MATLAB based application. What we needed was to bring the data form the GUI to the HTRF database and run the audio processing. And the problem we met was how to bring the data from GUI to HTRF database. The solution we found to solve this problem was save the data, which was gathering by GUI, into local file, and read it using the database. It is a bit slow, but it worked out well. We have the stretch goal to integrate the MATLAB app into a hardware and this part of research should been done by Max. But the stretch goal would be canceled as we ran out of time. See figure 5 for referenced WBS.

4.4 Hardware Implementation - Maximillian

Choosing our hardware will have to be cancelled for this project due to time constraints. If we submit a list of parts for our project now, we will not get them until past UGrads. This is due to the fact that our program needed to be done before deciding how much processing power was needed to run our program on said hardware.
Coding in C will be attempted. We hope we can get a working prototype made that works in C, so that if we did have hardware, we could easily integrate it into our controller. We would do this by using MATLAB Coder to translate our MATLAB code to C code. The translation may not be perfect to translate to hardware though. Some further manual translation may be done to optimize the transition to hardware code. See figure 6 for referenced WBS.

4.5 Status Update

As of this moment, we have completed the tasks involved in three of our four work breakdown structures. As previously mentioned, we unfortunately did not have time to implement our design onto hardware, and as a result were unable to complete our hardware implementation WBS.
Our goals for the remainder of the projects lifespan are as follows:
1. Polish GUI and improve ease of use
2. Increase quality of replayed audio
These two goals are not required for the project to work, but are goals we would like to reach so as to improve overall quality of our solution.

WBS of Hangdi Hu				
	Task	Description	Deliverables	Other People
1	Build a PC based GUI			
1.1	Build a standalone GUI	Build a GUI in MATLAB that has no connection of program	• Fundamental buttons on the interface: coordinates, volume, run function • Can let users input three dimensional coordinates (x,y,z)	
1.2	Improve the GUI	Improve and test		
1.2.1	Connect with code or program	When Stuart finishes his HTRF, I will edit code to let GUI connect with program	• This part can't delivery individually and needs to test in 1.2.3 to see whether it works well	Need code to be completed by Stuart Jackson
1.2.2	Interactive	Make program run the users' input data	• When inputting coordinates, these data can be seen from the variables window in MATLAB • The data equals to input coordinates	Help from Anjun Zhang
1.2.3	Test	Test the GUI, at the same time test the HTRF	• Can hear the positional sound • Change the location, can feel that the direction of sound changes • The volume of sound can be changed	Testing help from Stuart Jackson, Anjun Zhang, and Max Jones

Figure 4: GUI WBS

5 Metrics for Success - Anjun

Our primary goal has been completed. We have integrated the GUI and HRTF database to a MATLAB based application. Our application can work in real time. The app could record the sound from microphone, and make it seem as though the sound is coming from another direction via the audio processing. There is about a 0.2 sec delay. We have three inputs from the GUI, which are (x, y, z), to select the point. The point may then be displayed on the attached graph. Once we click the play button, the app would start recording and run the audio processing. We can easily tell that the sound that the app played was from the direction we've entered from the GUI.

6 Conclusion

While our design process has included a few bumps in the road, overall we have more or less met the goals we set out to achieve. This has been done in the form of creating a working algorithm that generates 3D emulated audio, as well as a connected GUI. The GUI has also been designed to be as user friendly as possible.

While we unfortunately could not complete our stretch goal of implementing this design on a standalone device, we have laid the groundwork for this to be easily implemented by any reasonably motivated individual. We have plans in motion to continue work on our design in the coming weeks before the final due date, with primarily quality of life updates to the design. These updates will include polishing the GUI to improve ease of use, as well as attempting to increase the quality of audio outputted via our processing block.

Person Primarily Responsible: Anjun Zhang				
ID	**Activity / Task**	**Description**	**Deliverable(s)**	**Other People**
1	**Integration** (step 1)			
1.1	Integration of GUI and HRTF database	After we have a nice-looking GUI and a functional HRTF database, we need integrate those together in to a user friendly and functional MATLAB based application.		
1.1.1	Integration	Actual work to integrate GUI and HRTF database.	1.GUI 2.HRTF database 3.MATLAB application	
1.1.2	Simulation and Test	Build up the application and run test.	1.Test data; 2.Simulation result 3.Application	All team members would take part in the testing.
1.2	**Integration** (step 2)	Stretch goal. We need put our app into a standalone device.		
1.2.1	Translate	Translate MATLAB language into hardware language	1.MATLAB application 2.Translated hardware language 3.Hardware standby.	Lots of work to do, could use some help from team.
1.2.2	Integration	Integrate translated language into hardware like VHDL (haven't deiced yet)	1.Hardware working with positional sound. 2.Working with different type of headset 3.Working with different sound source.	

Figure 5: Integration WBS

Overall, we are currently on track to have our project complete by the final date, and have achieved our primary metrics for success.

Person primaraly Responsible: Maximillian Jones				
ID	Activity/Task	Description	Deliverables	Other People
1	**Choosing hardware**			**All team members should be present for the choosing**
1.1	Determining the necessary peripherals	Things like usb ports or audio jacks	1. Amount of people using the device at one time	
1.2	Determine the correct processor for the job	Use an arduino or a FPGA?	1. Amount of people using the device at one time 2. Processing strain from our program	
1.3	Putting together our hardware implementation	The assembly of our end product	1. Durability of Hardware 2. sufficient power needed to run hardware properly	
2	**Coding in our harware**			**Needs the MATLAB program to be completed by Stuart Jackson**
2.1	Using a program to Translate to C code	MATLAB Coder or Simulink Coder would be used. Whether or not a Dspace board is needed is yet to be determined	1. MATLAB program 2. C Program	
2.2	Use the C code to Program our hardware	Translate our C code to VHDL if needed	1. C Program 2. VHDL Program	

Figure 6: Standalone Implementation WBS