Casey Celestin

Advisors:

Scott Petersen

Konrad Kaczmarek

Report on Tools and Technologies

When diving into the creation of my senior project, there were topics that were essential to its success that had only been touched on in curriculum. Therefore, before beginning the process of creating it, I had to dive into the technology and tools available that would allow the project to work seamlessly with the different forms of media that I was intending on using. Careful thought and consideration had to put into choosing the medium that would join audio processing and virtual reality graphics processing. Furthermore, I had to decide on which tools that would be used to create the spatialized sound that was needed, while still keeping flexibility and virtual reality integration a possibility. This report will be broken down into the following technological sections: coding medium, ambisonics, binaural audio, and convolution matrices. It will detail the technologies used, explain them in brief detail, and describe my intentions behind choosing a given tool.

Coding Medium

The beginning of the project was focused on taking stock of which media needed to be processed for and how they could be joined. Right away, the glaring issue was how the audio and virtual reality processing would go together. One of the big hurdles to clear was that commercial virtual reality headsets are difficult to work with on a low level. Many of the popular virtual

reality engines, such as Unity and Unreal Engine, are polished pieces of software, where the low level guts are masked. They have code that will automatically handle spatialized sound and are primarily for game development, not niche audio projects. For these reasons, these programs were avoided as to look for a more flexible, low level approach that would let me create a more specialized version of the project as well as a gain a better understanding of how the audio and visual will meld. Looking for low level solutions, I turned to the program SuperCollider, a programming language for algorithmic composition and real time audio synthesis and processing. The problem encountered was that it did not support graphics or virtual reality processing on its own and would have to be used in tandem with software like Unity. It would be possible to create this project using SuperCollider and Unity, having SuperCollider send OSC messages, a message based protocol for multimedia devices, to Unity, but the initial complexity of that solution pushed me to look for other paths.

Around this time, Professor Kaczmarek demonstrated a Max MSP patch he had been working on that used basic virtual reality graphics processing. I had already been familiar with Max through Professor Kaczmarek's class and knew it was a visual programming language that showed its power in multimedia projects. The audio processing is not as low level as SuperCollider, but still allowed for quite a bit of audio manipulation. The main advantage of Max is its integration with OpenGl, a cross platform API for rendering 2D and 3D graphics. Once the graphics are rendered, they are interfaced for virtual reality headsets using the vr object created by Graham Wakefield. This allowed for me to create low level graphics and be able to place them in a virtual reality space, which would prove to be an advantage as I had no prior experience in graphics processing. The only remaining questions were if Max's audio processing

was robust enough to create the spatialized sound that was needed and if the audio could be mapped in real time to different graphics parameters. Although this questions persisted, I decided that Max would be the best coding medium for me as it combined the low level data manipulation needed as well as being a strong interface between different forms of media, especially audio and visual.

Ambisonics

The leading form of sound spatialization today is ambisonics. Ambisonics is a full sphere surround sound format, meaning along with horizontal spatialization, it is capable producing sound above and below the listener. Developed in the 1970's, the technology has laid dormant but is now seeing a renaissance because of virtual reality technology. Although confused with surround sound, they act in a very different manor. Sound gets encoded into a form called B-format, which splits the audio into spherical harmonic decompositions of the sound field. The B-format consists of four channels. The first, named W, gives overall pressure of a sound and is equivalent as a omnidirectional microphone. The following three are named X, Y, and Z, with X representing front and back, Y representing left and right, and Z representing top and bottom. An encoder will distribute an incoming signal over these components given the position the audio source is set at. This basic form of ambisonics is called first order ambisonics. To increase fidelity and spatialization, higher order ambisonics add more components to the B-format that fill in directional gaps that exists at lower orders. Because the sound is processed in this way, it can be decoded down into many formats, from stereo to a multichannel outputs. For this project, I was concerned with decoding the B-format down to a binaural audio format.

Implementing this in Max meant searching for any ambisonics toolkits that had been created by other Max users, as Max did not have one built in. In talking with Professor Kaczmarek, I learned about a few packages that I will detail here. The first, most complete package is called Spat, a spatialization software suite for Max created by IRCAM. I found the suite to be overwhelming with all of its features. There seemed to be many things I could create with Spat but I decided that for this semester, the simpler package would suit me better, as I was constantly trying not to get lost in the detail. The next toolkit I explored was the HoaLibrary by Pierre Guillot, Eliott Paris, and Julien Colafrancesco from the University of Paris. This was a less involved toolkit, but one I was able to understand quickly and use on a lower level. The project continued using this kit without major flaw, though I will probably switch to a more robust ambisonics implementation as the project progresses. Additionally I looked in to the ICST ambisonics toolkit by Jan Schacher. This did a similar job to the HoaLibrary but as I was already up and running with the HoaLibrary, I saw no reason to switch. Future tasks include trying all of the ambisonics toolkits in my current implementation to see which one performs the best.

Binaural Audio

Beyond ambisonics, in the effort of increasing spatialization accuracy, one would be remiss not to include how humans physically interpret sound. Since headphones are the simplest medium for playback, as well as their use in virtual reality headsets, the final spatialized audio output will be piped straight into the listeners ears. This is a problem because humans use their ears, head, and face to localize sound around them. Humans use three evolutionary tools to aid sound localization: interaural time difference, interaural level difference, and head-related

transfer functions. Interaural time difference is the time difference of a signal hitting both ears. For example, a sound from our right side will hit the right ear before the second ear. Interaural level difference is the sound pressure level difference between both of your ears. This is the most effective singular way to create sound spatialization, though only effective in front of the listener on the horizontal plane. This is the process behind equal power panning. The last and most complex are head-related transfer functions, in which your head, face, and ears filter signal coming from different areas in different ways. So as sound from your right travels through your face, it is low pass filtered and what the left ear hears is not the same as the right. Also, since high frequencies are more directional than lower once and your ears face towards the front, sounds from your back sound filtered when compared to the same sound coming from in front of you.

Because the final output of this project is binaural, all of these spatialization tools that our body uses will need to be implemented synthetically in the Max patch. Through research on this topic, I found a head-related transfer function database of impulse responses by Olivier Warusfel at IRCAM. Each subject folder includes wav files of impulse responses as recorded by two microphones in the ear canals. The impulses were repeated all over the head at fifteen degree increments at a radius of 195 centimeters to give a complete spatial impulse response for each subject. The ambisonics toolkit helps with interaural time and level difference but these impulse response will help filter the incoming signal into something closer to how we hear spatialized sound.

Convolution Matrices

Implementation of these spatialized impulse responses will have to be used with a special type of convolution reverb that can handle multiple spatialized signals and apply different impulse responses to each. Convolution reverb is a reverberation process in which an impulse response of a space is taken, then a input signal is convolved with the impulse response to give the impression that the signal is being played in the space represented. This is a fairly basic concept and easily implemented in Max on a single sound source and impulse response but gets quite complex when working with multiple spatial channels, each needing to be convolved with a specific impulse response. Through my research, I have found the Max object multiconvolve~, which is part of the HISSTools Impulse Response Toolbox by Alex Harker and Pierre Alexandre Tremblay. This allows for a convolution matrix, where each input-output path can have a specific impulse response set to it by using buffers. Since the head-related impulse responses were taken in fifteen degree increments, I set the ambisonics decoder to output to 24 evenly spaced outputs then put them into a multiconvolve~ where each input was mapped to be convolved with the appropriate impulse response according to its angle. For each input, it is put through a left and right impulse response, then these signals are routed to the two outputs corresponding to left and right. Through this process, the program is able to apply the head-related impulse responses in a streamlined way, add to the spatial accuracy, and create a binaural output to go to the virtual reality headset.

Summary

The piecing together of this project brought together many forms of digital audio and video processing and learning about these subjects above were essential in tackling the problems

that arose through this process. Each of these technologies came with their learning curves and implementation issues but all of them were instrumental in creating the spatialized virtual reality space that was the inspiration and purpose for this project. Most importantly, these topics and tools helped me complete my project and I am very thankful to all the people named above who's code I have used as well as Professor Kaczmarek and Petersen who have helped me learn and interpret these topics. Overall, the information gained from this project will carry over into my other work, as I now have a firm grap over Max, ambisonics, binaural audio, and convolution matrices.