The Feeling of Sound:
A Study of the Interaction between Audio and Haptic Technology

Matthew J. Ciarletto
and
Christian W. Ruiz
Department of Electrical & Computer Engineering and Computer Science
Tagliatela College of Engineering

Abstract:

Music visualizers are a common component of many popular media players, which adds another dimension to the listening experience. In the same way that the sense of sight can enhance sound, the sense of touch can also enhance sound. To implement this idea of integrating haptic technology with sound, a wristband was created that contains motors that vibrate in accordance with sound. A computer program was written to communicate with the wristband and control it via Bluetooth. The wristband successfully replicates the bass-range of audio, from approximately 0 to 120 Hertz, and effectively matches the beat of most songs. In addition to music, the program can be used in conjunction with a microphone, allowing the wearer to feel the sound that is happening around them in real time. This would prove helpful to the deaf, because though they cannot hear sound, with the wristband they could feel it.

Introduction:

In today's world, there are many ways of experiencing multimedia such as three-dimensional video, motion sensor technology, and audio visualization. By reproducing the essence of sound into the sense of sight, the experience is intensified. By also reproducing the essence of sound into something that can be felt, the experience would be brought into an entirely new dimension.

This research explored the effectiveness and feasibility of turning audio into something that can be felt, not just heard. The overall objective was to create a wristband containing small vibrating motors that would vibrate along with audio, thereby enhancing the listening experience. Specific goals included using Bluetooth communication with the wristband, writing a computer program to control it, and determining the most effective way to translate sound into a tactile experience.

Wrist Band Design

Before beginning construction of the wrist band, several different designs and components were considered in order to come up with the best balance of size, function and battery life.

To determine the best design, experiments were conducted on several different hardware types. Of all the motors that were tested, only one type gave sufficient vibration and was small enough. It was decided that that particular motor would be used in the prototype.

It was also decided that an Arduino microcontroller would be used for the prototype, because they are widely used and the code is very portable and can be used on a different Arduino product with the same microcontroller.

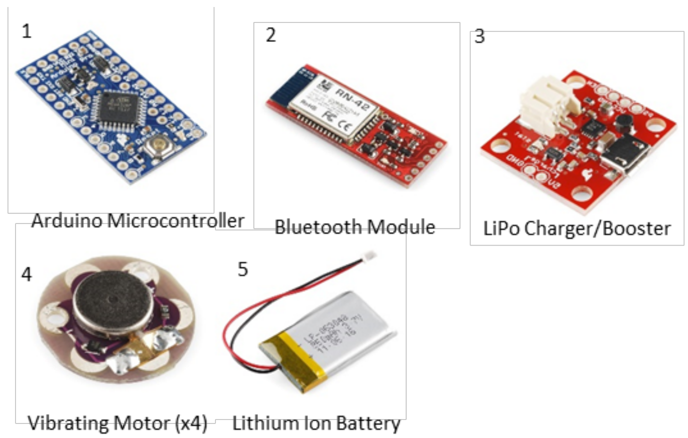

Figure 1: Components

It was necessary to pulse-width modulation (PWM) for output so that the magnitude of the vibrations in the motors could be controlled – with digital output, only whether they are on or off could be controlled. Three Arduino boards using the same microcontroller with sufficient PWM outputs were tested.

One, the LilyPad, had an onboard mp3 decoder, which would have allowed audio to be played directly from the wristband. However, that would have also made it necessary to incorporate a screen and several buttons on the wristband. The microcontroller alone would not be able to handle all the necessary functions of the wristband, and the PCB board was physically too large to fit into a reasonably-sized wristband.

Most of the testing was performed on an Arduino Pro, which does not have an mp3 decoder but is much smaller than the LilyPad. After testing proved successful with the Arduino Pro, the code was tested on an Arduino Pro Mini, which has all of the same relevant specifications but is significantly smaller than the regular Pro. As expected, there were no notable changes between both boards, therefore, the Arduino Pro Mini would be used in the prototype.

Using a computer program to send the output to the Arduino, the Arduino was set up to send the data directly to the output pins. This minimized the workload on the microcontroller, which correlates to lower power consumption, therefore, better battery life. Additionally, since it was controlled by the computer with this design, the wristband did not need a screen or buttons to control it; everything could be done via the program.

In order to communicate with the wrist band, a Bluetooth module was needed. One specifically designed for use with the Arduino microcontrollers that were tested was chosen and incorporated into the design.

Once both the computer program and the microcontroller code were working, testing was done to determine the best quantity and placement of the motors. After trying several different designs, most of which consisted of three motors, it was decided that four motors provided the strongest feedback. They were evenly spaced to stagger the locations of the motors and the hardware: motors were placed on either side of the Arduino microcontroller, the Bluetooth module, and the battery and charging module.

Once the layout of the hardware components was decided, all of the appropriate pins were connected with wires and soldered. The hardware was then placed into a fabric sleeve with an on/off switch.

The Computer Application

Early testing was performed without a graphical user interface (GUI) in order to test the serial communication between the Arduino and the computer. The prototype design required a GUI, as it was intended to be user-friendly.

Several platforms for development were considered. QT Creator and GTK+ were evaluated for developing the GUI. For ease of integration with the existing code from earlier testing, the GUI was developed using Microsoft's .NET framework. The programming language C++ was used to write all parts of the application.

The purpose of the wristband was to represent sound in a tactile format. However, one does not simply send audio signals directly to a motor. A Fast Fourier Transform (FFT) has to be applied to the audio, which splits the audio signal into an array of values that represent the magnitudes in a range of frequencies.

An audio library that contains an FFT as well as other useful functionalities such as starting/pausing audio files called BASS, by Un4seen Developments, was used for this application. The audio-playing features as well as the FFT were utilized in this application for the sake of working easily with a variety of types of audio, from mp3s to the microphone.

The magnitudes, as determined by sending audio through the FFT at a select few frequency ranges, were used to determine how powerfully the motors should vibrate at every point in time. The louder the sound in those ranges, the more intense the vibrations would be to the motors.

Much testing was performed to determine the best overall range of frequencies to utilize. Many different combinations were tried, from the lowest 40 Hertz to the entire audible range. In all cases, the value sent to the motors was proportional to the average of the intensities at each analyzed frequency range. This value was sent from the computer application to the Arduino microcontroller. From there, the Arduino sent the values directly to the motors.

The application implemented a multiplier, which the user could set via the GUI. This value would multiply the value sent to the motors, acting like a volume control, while the GUI changes the intensity of the motors. This gives more control to the user.

Another component in the GUI was the ability to connect to a serial port on the computer. This allows the user to choose where the data is being sent. When a Bluetooth device is connected to a computer for the first time, it sets up a communications (COM) port. This port will automatically appear in this application, making it easy to set up the communication.

Additionally, a music visualizer was incorporated, which not only adds the visual dimension to this application, but also makes it easy to see why the wristband vibrates as it does. The visualizer prints out one pixel for each frequency range, which is approximately 40 Hertz, as calculated by the FFT. From the bottom up, it prints the low frequencies to the high frequencies. The color of each pixel corresponds to the volume in that range: black is silent, blue is quiet, and then purple, red, orange, and yellow represent progressively louder volume levels.

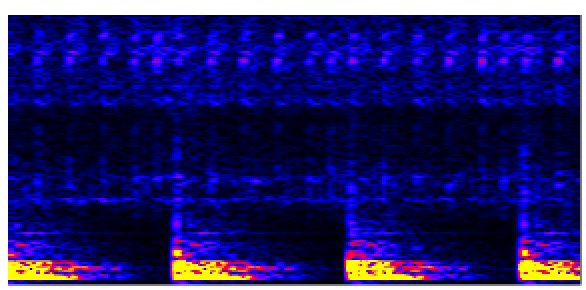

Figure 2: Audio Spectrum

Displaying this visual representation of the FFT made testing more efficient, as it was possible to see the values that corresponded to the motors' vibrations. It also allowed for analysis of the loudest frequencies of different songs, different genres of music in general, and audio as it comes in through the microphone. The overall goal was to develop a method of translating the audio into vibrations that would work across all audio, not just one or two specific genres of music.

A slider for volume, a slider for the current time of the song playing, metadata from the current song, and a button for switching between playing audio files and using the microphone are some of the other major features included in the program, both for the sake of making testing easier as well as improving the user's experience.

Results and Discussion

In the end, a prototype wristband was created with four vibrating motors. It could connect to a computer via Bluetooth, making it completely wireless. In conjunction with the wristband, a computer application was built that controls the wristband by sending data to the Bluetooth connection. The application can open, play and pause songs, and the wristband responds appropriately. The application implements a Fast Fourier Transform to divide the audio signal into separated, meaningful values, which are used to make the motors vibrate.

Figure 3: Wristband Prototype

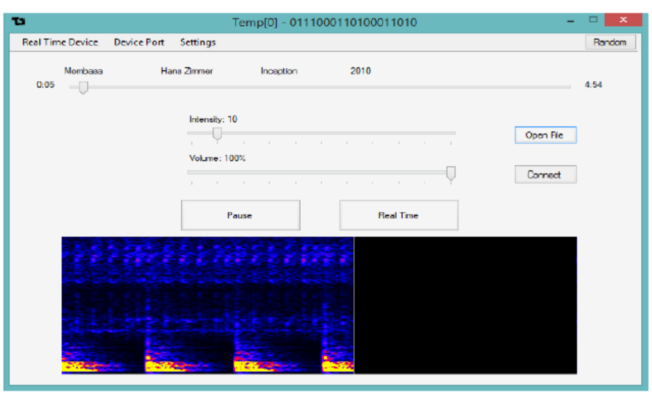

Figure 4: Computer Program

The FFT splitsthe sound into ranges of approximately 40 Hertz. For this application, the prototype takes the three lowest – from 0 to approximately 120 Hertz, which is entirely bass-range audio – frequency values, averages them, and sends that value to the Arduino microcontroller. That value is then sent to the motors.

This small frequency range was chosen because it was the most consistent among all the audio that were tested. Though it would be ideal to get the wristband to vibrate with the melody of a song and not just with the bass, which mostly keeps the beat, the melodies are so different from song to song that what would work for one song might produce next to no vibrations for a different song. In almost every song that was tested across several genres, from Hans Zimmer and Imagine Dragons to The Who, as well as the microphone, there was some level of consistent low-frequency sound.

Had higher-frequencies been used, some songs might produce virtually nothing. This design is not ideal for every song, but it is best on the average among all the tested designs.

The wristband hardware achieved all that it was set out to achieve. The motors produce significant vibrations and it is completely wireless. Among those who tested the device, there was general consensus that it adds a great deal of entertainment value to the music-listening experience. There is some agreement that it would be more satisfying if it could incorporate the melody of songs, but that it is still fun as it is. The components make this prototype bulky, but if it were to be mass-produced, smaller, more specialized components could be used that would make the wristband much more compact as well as inexpensive.

Though the wristband and computer application have succeeded in achieving their entertainment value, most people who tested it believe the device's highest potential lies in its ability to assist the deaf.

If a person is deaf s/he is not likely to be aware of the fact that anyone is trying to talk to him/her. In addition, s/he may not hear urgent sounds such as sirens or car horns. This is not just inconvenient, but may result in dangerous situations.

This wristband could make them aware of sounds around them by using the program in conjunction with a microphone instead of with music. Though the application currently uses only bass-range audio, the code could be modified to work with higher-frequency ranges, such as the range of normal speech and the range of common car horns and sirens.

In addition, it could be programmed to recognize patterns of sound: in other words, it could recognize something like a gradual rise followed by a gradual decrease in the frequency of a sound. This would help detect sirens; a special pattern of vibrations could exist, which would help clearly distinguish sirens from other, less urgent, sounds and would let the wearer know that an emergency vehicle is approaching.

Both in terms of entertainment value and helping the deaf, those who tested the device agreed that it has significant potential.

References

Arduino Pro Mini 328 - 5V/16MHz [Web Photo]. Retrieved from https://dlnmh9ip6v2uc.cloudfront.net//images/products/1/1/1/1/3/11113-01d.jpg

Bluetooth Mate Silver [Web Photo]. Retrieved from https://dlnmh9ip6v2uc.cloudfront.net//images/products/1/0/3/9/3/10393-01b.jpg

Power Cell - LiPo Charger/Booster [Web Photo]. Retrieved from https://dlnmh9ip6v2uc.cloudfront.net//images/products/1/1/2/3/1/11231-01a.jpg

LilyPad Vibe Board [Web Photo]. Retrieved from https://dlnmh9ip6v2uc.cloudfront.net//images/products/1/1/0/0/8/11008-01.jpg

Polymer Lithium Ion Battery - 850mAh [Web Photo]. Retrieved from https://dlnmh9ip6v2uc.cloudfront.net//images/products/3/4/1/00341-01b.jpg

Acknowledgements

We would like to thank our research advisor and mentor, Dr. Christopher Martinez, for his knowledge and support throughout the research, as well as Mark Morton, who helped order hardware and solder the prototype. We would also like to thank the SURF program for making our research a reality, as well as the all the donors for their contributions that made this possible, particularly the Carrubbas for getting the program started. It is only with the help of everyone who made SURF possible and helped us with our research that we were able to turn our ideas into a reality.

Biographies

Matt Ciarletto is a third-year Computer Science student at the University of New Haven. He enjoyed the opportunity to perform research as an undergraduate, and now hopes to eventually go into research and development, creating consumer software or working on the software-side of hardware products. He currently works as a cashier and as the webmaster for his father's brick-and-mortar business, the Kitchen Corner, in Fairfield, CT, as well as a computer science tutor at the Center for Learning Resources at the University of New Haven. In addition to his love for computer science, he enjoys golf, the piano, and driving his car.

Christian Ruiz is a third-year Computer Engineering student with a minor in Computer Science at the University of New Haven. With previous interest in music and participation in Marching Band, he enjoyed working on this project and hopes to continue development of this device. After receiving his bachelor's degree, he hopes to continue his studies to earn a Master's degree in Computer Science back in his home state of California. He is interested in the concept of virtual reality and thus hopes to work in that industry in the future. He is part of the Alpha Lambda Delta and the Etta Kappa Nu Honor Societies. Christian Ruiz is from Temecula, California.