Visualize the Invisible

College of Engineering
University of Utah
Salt Lake City, Utah
United States of America

Alex Gray
agray0659@gmail.com

Beverly Yee
bevyujw@gmail.com

Website: https://my.eng.utah.edu/~byee/index.html

Repository: https://github.com/BYee7127/public_html

February 2023

***Abstract*—The modern world offers a variety of high-tech devices on which we can listen to music. Most of those devices, however, cannot provide visual stimulation to go alongside the audio stimulus. The first solution to this was a visualizer that reacted to the music's intensity/volume followed by one that is simply a single column of lights. Neither of these solutions was stimulating enough, leading to the development of a new solution–the music frequency visualizer. In this method, small sections of frequencies are analyzed to create multiple columns of visualizations.**

This project will create an indoor water show like that at the Bellagio Hotel to enhance the user's music-listening experience. But instead of pre-set patterns scheduled to run at certain times, like the show at the hotel, real-time calculations will be performed to generate a unique visual response based on an analysis of the music being played. This report will describe the materials and knowledge required to realize the concept of a wired or wireless indoor music-reactive water system.

I. Introduction & Motivation

A. *Mission Statement*

The Bellagio's mini-me: a dance of water fountains to the beat of any chosen musical piece and is highlighted by a spectrum of LEDs. The height of the water and the intensity or color of the LEDs is determined by the frequency/pitch of the music. Visualize the Invisible.

B. *Background*

A music visualizer is software that "captures data from a music audio file" and creates shapes and images based on that data. This, in a way, allows listeners to "see" the songs they are listening to [6]. Most of the time, these visualizers can be found on personal computers. An older example would be the Windows Media Player 7, rolled out in mid-2000 [4].

The first visualizers were created by Robert Brown in 1976 [5], the same person who created the game Pong. In the beginning, music visualization was primarily used in video games. As time progressed, these music visualizers transitioned from video games to personal computers. Nowadays, music visualizers appear on the internet as freeware, that is free software, for the public to make as their own and exist in a vast variety. On the computer side, this software is used mainly to create music videos for platforms like YouTube, Facebook, or Twitter.

There are also hardware-based visualizers–DJs use them in clubs to create lighting effects on the dance floor. On the smaller side, there are commercially-available speakers that have attachments for such visualizers, such as lava lamps and pumps. The latter is the inspiration of this project, as it also uses water pumps and LED lights. However, these speakers only 4 pumps that shoot water at the same height, based on the total intensity of the music playing through them.

The goal of this project is to not only recreate that speaker, but double or triple the pumps. Additionally, vary when the pumps are turned on and the height based on the music to create a unique pattern. That pattern is retrieved by putting the music through a Fast Fourier Transform in the micro-controller used to control the pumps.

II. Project Tasks

A. *Specific Task Interfaces*

*1) **PCB design**:* Ideally, a student-designed PCB (printed circuit board) will be the main controller of the project to **1)** control the height and release of the water from the pumps and **2)** control how the LED light strip(s) appear on said water. Should the ideal situation fall through, other options will be considered, as discussed in Section V.

[Schematic for pump driver placeholder]

[Schematic for LED driver placeholder]

[PCB design image for Pump driver placeholder]

[PCB design image for LED driver placeholder]

*2) **Fast Fourier Transform Analysis**:* Fast Fourier Transform (henceforth referred to as FFT) is an algorithm for analyzing signals to convert them into sections that describes the frequencies present. The frequency resolution of each spectral line is equal to the Sampling Rate divided by the number of points in the FFT. Therefore, a higher spectral resolution can be achieved by increasing the number of points to sample [1]. The downside to this, however, is that the more points given to the FFT, the more time it will take to calculate.

FFT works by taking a signal-in this case, the music piece-and sampling it over a period of time. That sample is then divided into its frequency components. Each of these components is a discrete, sinusoidal oscillation with a unique frequency, amplitude, and phase [2]. Fig. 1 represents the frequency graph of a sample signal in which 3 distinct frequencies were found.

Whenever there is a spike in the frequency of the original signal, that spike is reflected as the amplitude of the individual frequencies. That amplitude can then be converted into a usable, numerical value to represent the height of each bar of the visualizer; in this case, the height of the water column.

There are multiple versions of the FFT algorithm. The one used for this project is called the Cooley-Tukey FFT algorithm. Compared to the others, this algorithm has an improved Big O time complexity (Eq. 1) than the standard discrete Fourier Transform algorithm (Eq. 2) [3].

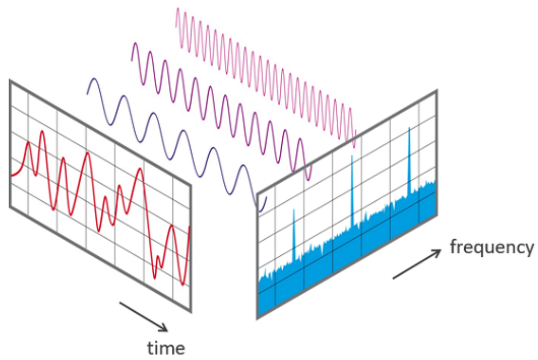

Fig. 1. A signal broken up into the 3 distinct frequencies that it is made of [2].

$$O(N * logN) \tag{1}$$

$$O(N^2) \tag{2}$$

An improved Big O time complexity is important for calculations as it allows close to real-time computation of signals such that there is not a lag between the audio and the visualization of said audio.

The method for computing the FFT recursively re-expresses the discrete Fourier transform (DFT) as a number of smaller DFTs recursively. The algorithm formula for DFTs can be seen below in (Eq. 3).

$$x_j = \sum_{n=0}^{N-1} x_n \exp{(-\frac{2\pi i}{N} * nj)} \tag{3}$$

*3) **Pump Control**:* The pumps used are pumps sold for small aquarium tanks and run between 3 to 5 volts. Controlling the voltage will in turn control the intensity/height of the water, which will be used to represent the intensity of each frequency.

As mentioned previously, a PCB will be used as the pump control, and it will need to be configured to connect with the micro-controller. There, it will receive and use the numerical data given by the FFT to determine exact voltage needed for each individual pump.

*4) **LED Control**:* Depending on the LEDs found, there are a few ways the lights can be controlled.

First, if individual LED lights are used, such as the ones found in Arduino or Discovery starter kits, a PCB is needed. This also applies to the SMD (Surface Mount Diode) LEDs. The advantage given by this way is the lower costs for materials and freedom to determine where the LEDs go. The disadvantages, however, are the possibility more calculations are needed for power; more work is required to get the hardware pieced together; and it may not look as nice as manufactured products

The second option is pre-made LED ring lights that can fit perfectly around the tubing of the pumps. The biggest advantage with these lights is the appeal–since there is a set of LEDs surrounding the outlet of the water pump, the columns of water would have more emphasis to them as they shoot out. The disadvantage is the cost. The cheapest option found on Amazon were roughly $15 each. Collecting enough for 8-16 pumps would result in one or two hundred dollars, just for lighting alone.

The final option is to use LED light strips that will be adhered along the inside of the container. This is by far the easiest option as only 1 LED PCB driver is needed to control the entire strip, and the lights will still be able to provide plenty of emphasis on the water in a way that it will still look appealing. The trick will be to angle the strip such that the water columns get the max effect. Additionally, the LED strip needs to either be already proofed to be used under water and kept out of the water somehow.

*5) **Final Assembly**:* The final product, roughly sketched in Fig. 2, will be an acrylic encapsulation filled partially with water. The pumps will be lined on the bottom, as they are required to be to work properly, with tubing that either stops just at the water line or protrudes above it. These tubes will control the movement of the water and make it so it shoots out in a column as intended.

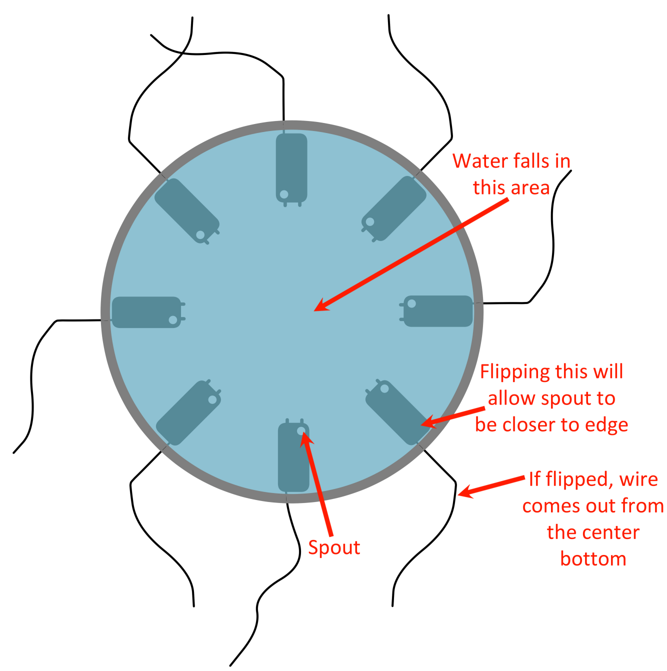

Fig. 2. Top-down view of what the final product looks like without the speaker or circuit boards attached.

The wires for the pumps will be snaked through holes drilled into the encapsulation and into a separate area. This separate area, henceforth called the service box, will house all the electrical components such as the micro-controller and PCBs for the pump and LED lights. The service box also serves as the connection point for a set of speakers, positioned one the sides of the container (Fig. 3) and providing the source of audio for the user.

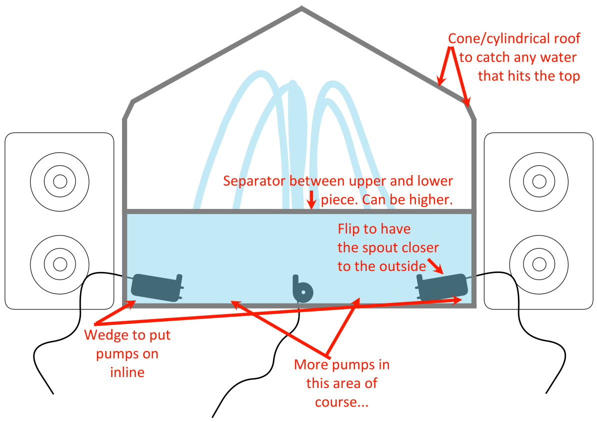

Fig. 3. Side view of what the final product looks like.

It should be noted that both Fig. 2 and Fig. 3 are initial sketches of what the final product should look like. The actual, physical construction may change as the project progresses to account for any obstacles that may appear.

B. *Stretch Goals*

There are a couple of ideas the collaborators would like to see added to the project, should time permit following the completion of the main goals. Each stretch goal is listed below, beginning at the highest priority.

*1) **Beat Sync**:* This particular goal is the most highly anticipated one as the desired output will step up the visualization of the input music. While the exact point is still in discussion, each time there is a drop in beat or bass is particularly strong, both the LED and water streams will respond with increased intensity equivalent to that beat.

The method of recognizing the beat for this goal will be a little more complicated. In addition to computing the FFT of the signal, an analysis of the amplitude data is needed to find the moment when one amplitude (on the lower end of the spectrum) is significantly louder than the neighboring amplitudes and/or a set threshold. This is because the beat is usually the loudest part of the entire spectrum when it hits.

The drawback lies in the accuracy of the detection at the cost of the speed of the calculation. As such, this goal remains a stretch goal as of the moment such that the processing speed of the signal remains in real-time.

*2) **User Interface**:* For this goal, the idea is to set up an LCD screen with buttons so a user can adjust the visualizer to their preferences, such as changing the song or switching up the settings of the LEDs to different options (eg. rainbow cycle, solid color, beat sync, etc). Having a user interface is not crucially necessary for the project to work, so it is considered a stretch goal.

*3) **Bluetooth**:* As the title suggests, instead of using an audio jack to receive audio, attach a module to the micro-controller so the visualizer can connect to Bluetooth-enabled devices, such as a smartphone. With little experience in dealing with Bluetooth, however, this stretch goal will be the last one to be considered.

III. Testing & Integration Strategy

*1) **Pumps**:* During the early stages of the project, each pump received must be tested individually to verify that is not a defective product. Once verified, the pumps will need to be tested at differing voltages to see how it operates and to give the range of values that can be used with the values given by the FFT. Ideally, each stream of water is to land/meet at the very center of the fountain when their pump is operating at max intensity.

*2) **Acrylic Casing**:* There is not much to test for on the enclosure. One of the requirements is for it to be able to be cut by one of the machines provided. Additionally, as it is difficult to find acrylic cylinders tall enough for this project's purposes, several shallow trays were bought. The bottoms of one or two of these trays may need to be cut out and glued onto another to create a taller enclosure. For this case, a water test is needed to ensure there is no leakage from the seams.

Water tests will need to be performed periodically throughout. The most import thing is for water not to get to the service box else the destruction of all electrical components are imminent.

*3) **FFT**:* TBA

*4) **Pump PCB**:* TBA

*5) **LED PCB**:* TBA

A. *Integration Process*

TBA

IV. Group Logistics

There are only two collaborators in this project, so rather than a group or team dynamic, there is more of a partnership. The work is to be equally divided, although further discussion on future tasks may/will be necessary.

A. Communication Plan

Current communication is mostly through Discord and brief after class discussions. The set schedule as things progress will be Mondays and Wednesdays and will take place either virtually for programming components or at the senior lab for hardware components.

B. Schedule & Milestones

The gantt chart (Fig. 4) is a tentative schedule as how the project will proceed. Please contact one of the collaborators if a larger figure is required.

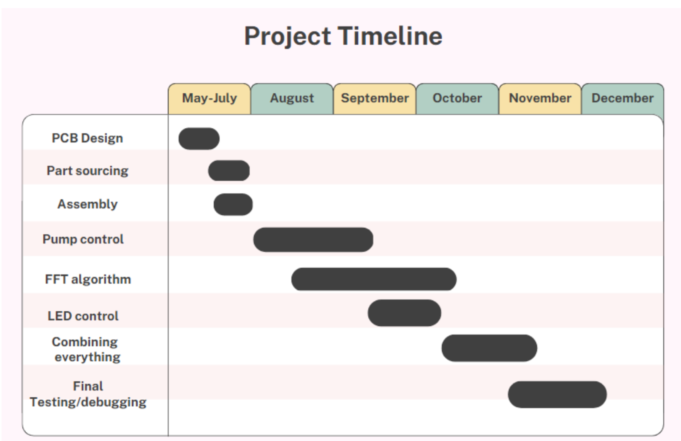

Fig. 4. Tentative project schedule.

The hope is to have at least one pump working as intended by the end of April/beginning of May to demonstrate a prototype. Once one pump can be controlled by the varying voltages, replicating the process for the others should not be difficult.

By the end of November/beginning of December, there should be a set of eight pumps minimum working. Ideally, upwards to 16 will be displayed for demonstration.

V. Risk Assessment

*1) **Audio Delay**:* Despite using an improved FFT algorithm to improve time, there is still the chance that real-time computation cannot be achievable. In that case, a small delay introduced at the beginning of the audio input may need to be introduced.

The idea is to jump start the calculations such that the algorithm is looking ahead. What the user is seeing and hearing is upwards to several seconds behind what the algorithm is computing, thus creating the illusion that the audio is being processed at real-time.

*2) **Variable Voltage**:* The voltages between the PCB and the micro-controller are incompatible or the difference is too much to buffer. Or there was a miscalculation on the power needed for all the pumps. Adding too much power to the component would only result in magic smokethat is, the component would be rendered useless. At that point, a new component is required. In case time is not friendly, spares will be set aside for this event.

Too little power, on the other hand, will result in the components not working as expected. For this scenario, heading back to the calculations and conversation with fellow Computer Engineers will, hopefully, solve that issue.

*3) **Lack of Sealant**:* Insufficient sealant was added around the wires where they exit the enclosure. Water leakage from those points can get to the circuit boards and fry the components. Just like in the previous point, new components would be needed should this happen. The hope is to apply enough sealant such that such a scenario doesn't come to pass. At the same time, however, too much sealant cannot be applied else risk the enclosure looking ugly.

*4) **Visual Delay**:* Much like the risk discussed in V-1, one of the many challenges faced in this project is getting the computations done quickly enough such the effects of the visualizer displays in real-time. As with the Audio Delay, a small delay may need to be introduced in the beginning such that the algorithm is ahead of what is being seen and heard.

*5) **Synchronization**:* Alongside computational delay, both the audio and visual effects need to be efficiently synchronized such that one is not faster or slower than the other. While it is not on par with the frustration found with un-synchronized audio/video in a movie, an un-synchronized visualizer will likely kill the simulation it's supposed to bring.

VI. Bill of Materials

The following list contains the materials required for this project:

1) Water pumps + tubing - specifically ones used in aquariums such that they are smaller and more wallet-friendly.
2) Speakers - one pair to start with; more will be added later.

3) Water pump driver PCB - either one created by the students or found online.
4) LED driver PCB - either one created by the students or found online.
5) LED strip
6) Microelectronic components - for PCBs
7) Container - preferably clear acrylic for components and water.
8) Sealant - to ensure water doesn't leak to where electrical components are stored.
9) Microcontroller - currently determining which micro-controller will best fit our needs.
10) Aux port part - for input
11) Water jug (1 gallon) - to be used as the source of water during testing and final product

This list may not reflect everything that will be used in the final assembly of the project.

A. *Vendor List*

1) Amazon
   - 4 water pumps + tubing - \$10.59 x4
     - ASIN #: B097F4576N
     - ordered 3/17/2023 - \$45.44
   - Smart LED strip lights - \$23.60
     - ASIN #: B07N1CMGQQ
     - ordered 2022-Jun-15
2) Lowe's
   - Sealant - \$10.15 Includes 10% employee savings
     - item #: 455441
3) LO Florist Supplies
   - Acrylic Trays 10" x 2.5" - \$7.99 x4
     - SKU #: ACYL102
     - ordered 2023-Mar-17 - \$45.02
4) Digikey/Mouser
   - Audio jack port -
     - model #: STX-3120-3B
   - Microelectronic parts for PCBs
5) OshPark
   - PCB manufacturing

VII. Conclusion

All-in-all, this capstone project is like a combination of the Bellagio Hotel water show and commerically-sold speakers with the four water pumps. The main difference is it shrinks the Bellagio's fountain to a more personable size and is can be operated at any time of day. It also doubles, triples, or even quadruples the number of pumps used to create more of an effect that is in tune with the music.

To create such an effect, the Cooley-Tukey FFT algorithm will be used to quickly create digital data from the audio signal, from where the varying heights of the water columns are computed. LEDs are added to the mix to further enhance the experience. The calculations for how those will appear in terms of color and maybe even pulse is also determined by the data given by the FFT. The algorithm will be found on the micro-controller used to mastermind the entire water show. Here is also where the speakers and controller PCBs for the LED and pumps will be connected.

The main jewel of the project is a clear, acrylic cylinder that 10" in diameter and several inches tall. It contains the water, LEDs, pumps, and tubing to create the water works. A set of speakers accompany the container on the sides to output the audio originally given. Finally, on either the back or bottom of the container is the "service box" in which the electrical components are stored.

References

[1] Anon. *FFT Size*. [Online]. URL: https://www.spectraplus.com/DT_help/fft_size.htm. [Accessed: 2023-Mar-22].

[2] NTi Audio. *Fast Fourier Transformation FFT - Basics*. [Online]. 2019. URL: https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft. [Accessed: 2023-Mar-22].

[3] Ben Levinson. *The Fast-Fourier Transform and Spectrograms for Audio Visualization*. [Online]. Aug. 27, 2017. URL: https://benlevinson.com/projects/spectrograms. Accessed: 2023-Mar-22.

[4] Microsoft. *Microsoft Windows Media Player 7 Brings Click and Play Digital Media To Millions Around the Globe*. [Online]. July 17, 2000. URL: https://news.microsoft.com/2000/07/17/microsoft-windows-media-player-7-brings-click-and-play-digital-media-to-millions-around-the-globe/. [Accessed: 2023-Mar-16].

[5] A. Rothstein. *The Basics of Music Visualizers*. [Online]. Dec. 6, 2019. URL: https://www.ipr.edu/blogs/audio-production/the-basics-of-music-visualizers/. [Accessed: 2023-Mar-16].

[6] Wanda Marie Thibodeaux. *What is Music Visulaization?* [Online]. Feb. 7, 2023. URL: https://www.musicalexpert.org/what-is-music-visualization.htm. [Accessed: 2023-Mar-16].