Frames: A 3-D Image Recording and Display System

Christian Vartanian, Daniel Hureira, Jamesley Rock

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

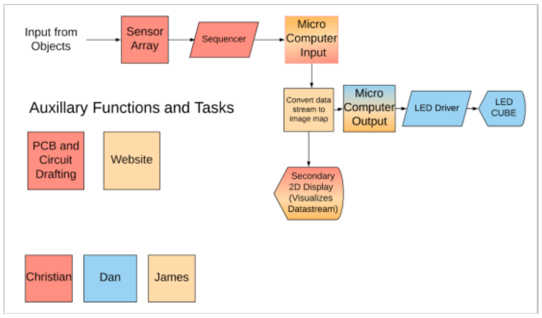
***Abstract* — Visual display systems have captured the attention of many since the first television was created in 1927. With technology rapidly advancing the same has happened to visual display. Throughout history visual projection or as some say projection mapping has changed rapidly. To modern day where projection and even holograms are common. This allows for visual equipment to be cheaper and even more environmentally friendly due to the technological advances. Our aim is to build a lightweight, relatively cheap, display system that can capture and display ultrasonic texture maps of a space or object. Which can be utilized for entertainment and even special forces purposes. By arranging several hundred LEDs in a cube, we will be able to display a 3D image using incoming data from a larger ultrasonic sensor array. Our motivation for picking this design was to not only challenge the skills we have learned throughout studying our undergrad but also to be able to reproduce a visual representation of what technology is capable of and displaying this to our fellow classmates and faculty.**

I. INTRODUCTION

The system is made of two core components, a two-dimensional array of ultrasonic sensors (henceforth known as the “Frame”), a microcomputer, and a volumetric array of LEDs (“Cube”). The Frame acts as the input of the system. As Ultrasonic Rangefinder (HC-SR04) units can generate a continuous stream of depth data, arranging these sensors in a matrix will allow for a projection map of any space to be captured. Originally we were going to send this texture map of the space can be directly serialized, passed over to a micro controller unit, compressed via a computer vision program, and then sent over a serial connection (potentially WIFI) to an LED Cube that will then utilize the X,Y,Z data generated by the Frame array to display a real time 3-D representation of whatever is in front of the sensor array. However, as time progressed, we realized that we would not be able to achieve this goal due to the low pixel density of the cube. Therefore, we decided instead, to have the Ultrasonic sensors detect if an object is near it, then proceed to make a sound the closer or farther an object gets to it, and lastly light up the cube or dim the cube based on the distance of the object.

II. OVERVIEW

Our project can be split into 3 major components. The LED Cube, the Ultrasonic sensor array and the Bela board/Beaglebone black. An image is taken in, sound is used to indicate the distance of the object, and the LED lights up to indicate the distance of the object. PCBs were used to connect the ultrasonic sensors together and be able to connect them to the Bela/Beaglebone Black. Each component of the project is essential to the next component. Every component is essential and interconnected with each other. Below is the block diagram of our project as well as the work distribution.



Distributing our project in this manner allowed us to complete our project in a timely manner and allow us to contribute to each other’s sections if need be.

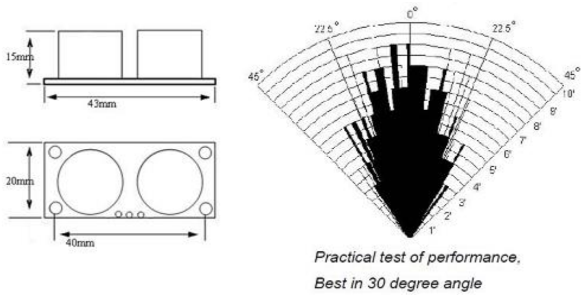
III. SYSTEM COMPONENTS

In this section, we will detail each individual component, their conception, their build process, and the testing/debugging phase.

A. Ultrasonic Sensor Array

In terms of Ultrasonic range finder devices, the HC-SR04 is probably the most ubiquitous design available. Two transducers connected to an 8-bit microcontroller allow for an easy and cheap method of detecting range. In lay terms the module sends out a high frequency “chirp” through one transducer and receives the echo generated by said chirp at the second transducer. When connected to a microcontroller/microcomputer one can write a small amount of code that takes the time between pulses sent from the output transducer (Trig) and those received at the input transducer (Echo). Using this time and multiplying by the speed of sound in air we can get the round-trip distance of the chirp (and thus the distance of the object which reflected the initial chirp).

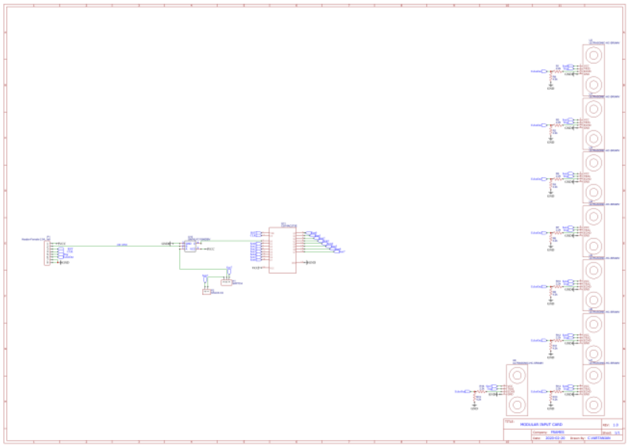
The HC-SR04 has been chosen for its well documented characteristics (operating ranges, cone of effect, ETC), wide availability, and low cost. With a +/- 3mm tolerance the device should be well suited for imaging of larger figures. Additionally, the package itself comes with four mounting holes for screws that allow them to easily be installed into the grid required by the Frames input system. These modules can be found quite cheaply from international suppliers and can also be made readily available domestically, albeit at a cost that may be prohibitive to install at a large scale.



While the HC-SR04 excels in form factor, price, and reliability (in both operation and sourcing), it does have some potential issues when being installed in the Frames unit that must be considered. As the module uses sounds to determine distance, cone propagation begins to affect the reading. A cone is projected in front of the Trig transducer and anything within that cone is subject sending an echo back to the Echo transducer. Fortunately, there is correction system for this potential data fuzzing, the HC-SR04 only records the data of the nearest (shortest time between pulses) object. While this works quite well for a single sensor set up at virtually shrinking the cone of "vision" of the unit, the effects of sound propagation begin to become a legitimate problem when one considers stacking upwards of 256 units together. As the units are placed side by side an issue quickly becomes apparent that the Echo Transducer of one module is extremely close (and potentially within the cone of effect of the transducer itself). As such it becomes important to think of a way of limiting crosstalk.

One possible method of "shrinking" the cone of effect is to erect dividers around each Trig transducer. This will effectively clip the propagation and allow more space for the echo to receive its intended chirp as opposed to that of a neighboring module. Each module has a cone of vision of about +/- 30 degrees from the center. And as such it may be possible to restrict that cone by simply erecting walls on either side of the module. These walls should be made in a way that they will not cause extraneous reflections from hitting the Echo transducer or else all of the work put into limiting crosstalk may be wasted as the dividers may generate false readings if they are more reflective than need be. Potentially placing dividers around only the transducer may remedy this as the transmission range will be shrunk and the Echo transducer will not be subject to an overage of excessive reflections, thus avoiding extraneous readings from being generated.

Below is an image of our PCB Schematic used to place our ultrasonic sensors in an array.



B. Beaglebone Black/Bela

This microcomputer is well suited to our needs, sporting a 1Ghz ARM Cortex-A8, ample GPIO pins, 512MB of RAM, and 4GB of flash memory. With this much headroom running into issues regarding resource shortages should be mitigated, thus guaranteeing (from a hardware perspective) enough computational resources to facilitate real time 3D imaging. Additionally, the BeagleBone Black's open source nature has led to a large community of developers and manufacturers creating modifications and capes for just about any foreseeable purpose.

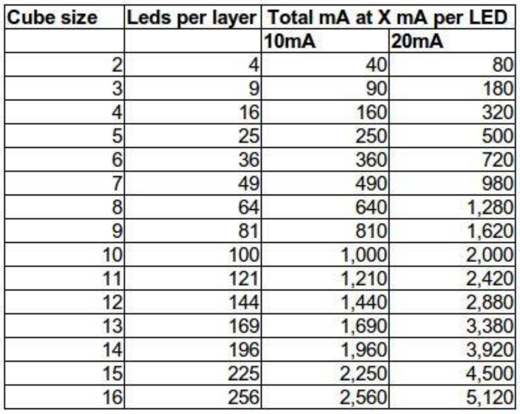
Additionally, the Beaglebone Black features a Wi-Fi chip that would allow for wireless interfacing with the data stream, and interconnect ability between microcomputers. While not directly applicable to our current system, if we are to increase the Frame's system beyond the scale of our current project, interconnectivity would become of paramount importance. With wireless communication between microcomputers, we would have the Frames input unit any distance away from the 3-Dimensional LED Cube Display. This would enable the system to be installed in almost any space that it would require, freeing it from tethers of wired communications.

One of the capes available for the BeagleBone is the Bela. Initially made for the development of audio processing tools and general experimental instrument design, the Bela works as a fantastic modification to the BeagleBone Black especially when real time I/O is of the highest priority. One of the main advantages of using this cape is its Linux shell that puts the highest priority on maintaining I/O speed and fidelity. Reportedly the cape can perform round trip I/O with times as small as 1ms, a speed ample enough to provide a responsive feel to the system. With the custom OS on board comes a portable IDE that allows for rapid prototyping and sensor data monitoring that would otherwise be cumbersome to develop on its own. The IDE has custom support for numerous languages, including C++, Super Collider, and Java.

As a bonus, Bela's IDE has fantastic support for developing GUIs. Bela’s support and graphics power should make it a breeze to develop a 2 display system, additionally this display output is easily accessible over USB or Wi-Fi via a laptop and can then be projected at any scale to best demonstrate what's happening in the system in real time.

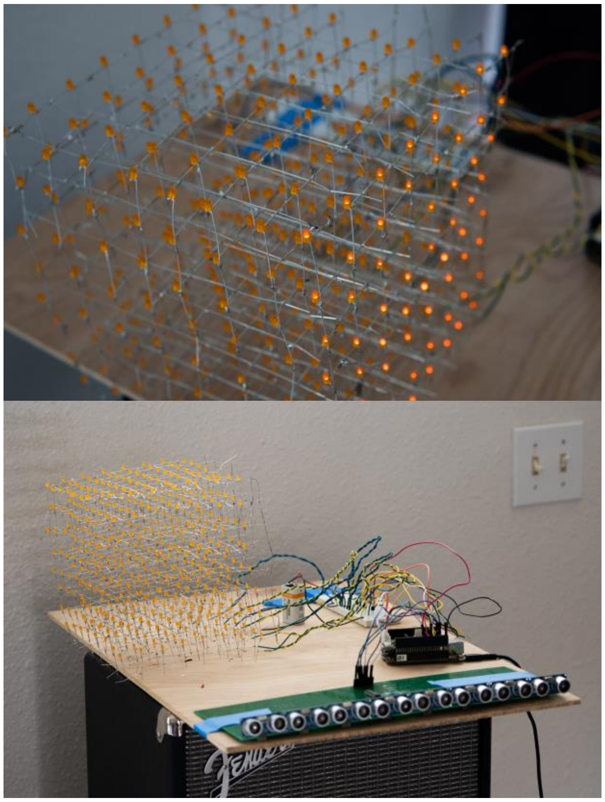
C. LED Cube

An integral part of this project is the LED Cube. When implementing the LED cube we must look at a plethora of things, building the LED cube, what LEDs to use, what resistors to use, how to integrate it with the software of Bela, how to power the cube, and much more. The following section we will begin to outline how we would go about building it and the theory behind LED Cubes. In order to construct the LED Cube, we had to purchase an appropriate number of LEDs. The following table lists the number of LEDs we needed. The Image after that shows the amount of power we will need.



Cube size	Leds per layer	Total mA at X mA per LED	
		10mA	20mA
2	4	40	80
3	9	90	180
4	16	160	320
5	25	250	500
6	36	360	720
7	49	490	980
8	64	640	1,280
9	81	810	1,620
10	100	1,000	2,000
11	121	1,210	2,420
12	144	1,440	2,880
13	169	1,690	3,380
14	196	1,960	3,920
15	225	2,250	4,500
16	256	2,560	5,120

Then, we must solder each cathode together to form the row, and then solder each anode together to form the columns. LED Cubes are essentially LED displays but with a third dimension. Because of this, you cannot stack pixels together like you normally would in an LED display, because you must see through the cube which makes it appear 3D. Therefore, you need spacing between the voxels (which are pixels in 3D). Because of the magnitude of building a 8x8x8 array, we will be manipulating an optical phenomenon called the persistence of vision effect (POV effect). This effect essentially states that an image will stay on your retina for some time after it has ceased but only if the image is flickering fast. A great example of this is a spinning torch. When you spin a torch, you’ll be able to see a full circle but, that is not what’s going on. The same applies to our LED cube. When we flash each layer of the cube really fast the image will display as if it is in 3D. The images below show the structure of our cube.



The persistence of vision effect is the main reason why we will be able to display 3D images on the LED cube. The brain can only process 10-12 separate images per second and each image can only be retained for up to a fifteenth of a second. Replacing an image with another one in this timeframe will trick the brain into thinking there is continuity between the images. This is how animation works in TV and movies. We will utilize this technique to create the illusion of 3D by flashing the LED layers in quick succession to trick your brain into assuming a 3D image is being displayed. Techniques such as this one has been implemented in many lighting shows

IV. SOFTWARE

The software we will write is an integral part of our project. Without the software, nothing will work. The basic idea is that the software should be able to process the image that the ultrasonic sensors are reading, and then perform the corresponding computations to get the desired output from the sound component and the LED Cube. Much of the software integration is already supported by Bela and Beaglebone. Therefore, the main thing we will have to do is be able to accurately translate the input to output. The GUI display support for the Bela will also come in handy when it comes to tweaking our project and making the interface look good for presentation.

As previously stated, the system employs a method of demuxing to group the sensors into sets small enough to be directly read into the microcomputer. By modulating which groups are powered on via a binary selection line, we can evade any dicey situations presented when trying to digitally demodulate a continuous analog stream. The general structure of our software suite is wholly informed by the way we have set up data collection and output as, with any embedded system, designing software for an unknown hardware system is nearly impossible.

Our software component is split into two main parts. A C++ file handling analog and digital I/O from the Sensor Array, Clocking the sequencer, and managing the audio output, and a JS file using the Process library to generate graphics for our GUI based on the data collected and generated by our C++ file. These two pieces of software, while slightly rudimentary, are robust enough to scaled to almost any size cube system and can control all the necessary parts of our system with ease.

V. POWER SUPPLY

When we began executing the project, we realized that the actual power supply requirements were not as high as we once thought. As our initial spec called for only a single LED to be lit at a given time, we would have been able to get away with a very simple and low power supply system. Our search led us to 3 options, an external Madewell type variable 12V supply, a DC Wall Wart connection for the Bela, and just a combination of 9V Batteries and Power of USB. The first of our options proved more bulky than necessary and was subsequently scrapped, along with the second option as we wanted to limit the amount of unnecessary connections as much as possible. If we opted to power the BeagleBone over USB we could save on a connection as we required a laptop to display our 2D GUI (Note: Had we opted to not use a 2D output we could have simply used a Wall Wart connection to the BeagleBone for powering). Once realizing that power over USB was the best option for simplifying our project as much as possible, we opted to include a 9V battery to power the cube. As even in the most power-hungry spec (the current one) we found that a 9V battery provided more than enough energy to illuminate the whole cube let alone a handful of LEDs. In larger scale versions of our project we may need to opt for a more robust source for our LED power supply. Preferably one with a Mains Connection such that current limitations or life span would not be an issue in a long-term performance situation.

VI. WORKING/OPERATING THE PROJECT

This project will operate by receiving an input via the ultrasonic sensors, (for example a hand) then displaying that hand in 2D on a monitor, then in 3D on an LED Cube. It is our hopes that this model/design will be scaled up in the future and used for music festivals or big events. This device is purely used as entertainment and the following section will outline troubleshooting tips, safety precautions and general information.

This device is a completely safe device. However, with any project, things can go wrong, therefore safety precautions must be taken when operating the device to ensure that all safety measures are taken.

1. Do not operate the device near any liquids to ensure the device is not damaged and no electrocution occurs.
2. Equip insulator gloves if work on the project must be done on the spot. This will prevent electrical shock.
3. Ensure that no wires are exposed, loose, or exposed to the elements. This will help prevent degradation and electrical shock.
4. If the situation arises where the device must be tampered with, please shut off the power and let it sit for a few minutes to ensure the device is not holding any electrical charge.
5. If any of the LEDs must be replaced, ensure that proper solder safety is followed.
6. Ensure that you are in a closed room so that you can properly see/use the device.

Following these safety precautions are important to ensure that no one gets hurt operating this device.

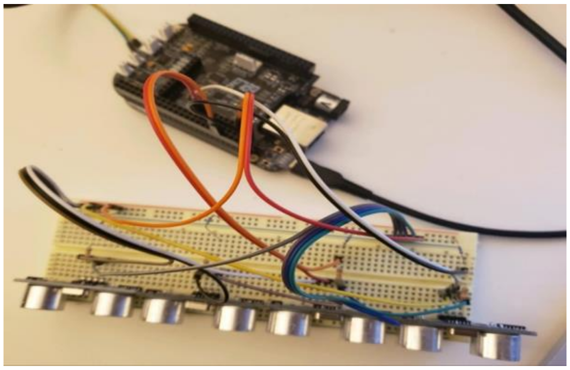
In order to use the device, ensure that all the connections are firm and properly made between the Bela, Beaglebone Black, Monitor, LED Cube, Ultrasonic Sensor Array, and any other connections that are made. Secondly, ensure that the software to properly render in 2D and 3D is up and running and working properly. Thirdly, ensure that the Ultrasonic Sensor Array is accurately receiving input. Lastly, give the Ultrasonic Sensor Array input for it to display. After the input is received, the software/Beaglebone Black will process the input and then render it in 2D then render it in 3D on the LED Cube. Before using the device please ensure that all safety precautions are taken.

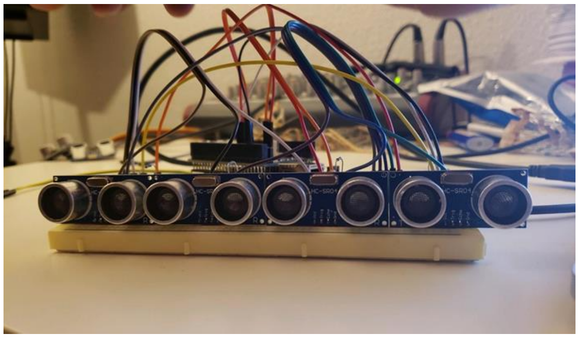
Listed below are a few tips and tricks for common troubleshooting issues you might have. This table will allow the user to fix common issues they might have when using the device. It is our hope that this section provides immense help and reduces the complication of a device that is intricate and complicated.

Problem	**Troubleshooting Tip**
Device not powering on	• Make sure the device is receiving power • Make sure there are no loose connections/wires
Device is not outputting correctly	• Make sure the LED Cube is working properly • Make sure the software is working properly • Make sure the Sensor Array is working properly
Ultrasonic Sensor Array is not picking up the correct input	• Ensure the target is at a reasonable distance from the Array. • Ensure that each sensor is working properly
LED Cube not responding to input properly	• Make sure each LED is working properly • Make sure that there are no loose connections on the LED to the Bela/Beaglebone

VI. PROTOTYPING/TESTING

It is important to ensure that each section of the project is working. Due to this we decided to begin prototyping each stage. The results below are from prototyping the main component which is the ultrasonic sensors. We did this as a proof of concept to ensure that we could stack our ultrasonic sensors together and run them, we decided to build a prototype connected to the Bela. We then tested this prototype to ensure that we could read/receive inputs and outputs. Furthermore, we used this time to test our sound component. Ensuring that the closer the object got to the sensors, the louder the sound would be and vice versa. We used a breadboard to simulate what our PCB will eventually do. After we got our PCB, we swapped it out, soldered the components and it worked. The pictures below are pictures of the prototype connected to the Bela.



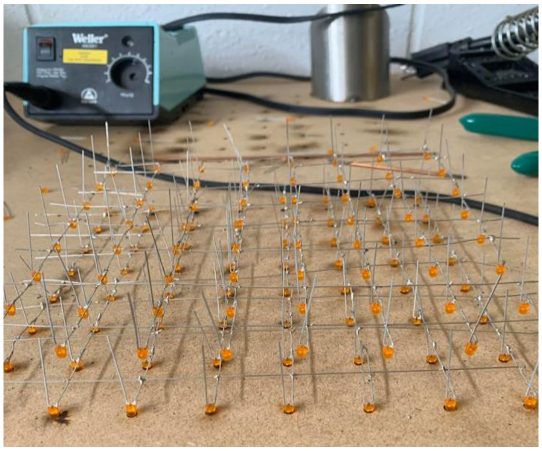


The pictures above show the ultrasonic sensors connected to a breadboard aligned. We used this prototype to develop software in order to ensure that connecting ultrasonic sensors in an array in this fashion could work. We also connected the sensors to the Bela board to ensure that the Bela would not only be compatible with the ultrasonic sensors, but we wanted to ensure the software on the Bela would be able to accept the input from the ultrasonic sensors. Furthermore, we tested the Ultrasonic sensors to ensure that they pick up objects located in front of it. Pictured below are graphs that indicate how close or far an object is from the sensor along with the pulse-widths and the actual object in front of the sensor.

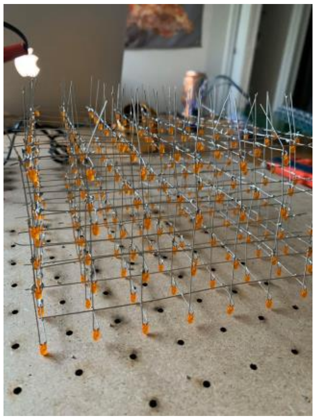
This was a proof-of-concept for the ultrasonic sensors. The objective afterwards, was to scale it up and place more ultrasonic sensors. This will allow the sensors to pick up more data points, the goal will be to take those data points, and then use the Bela to map those data points into a rendered image.

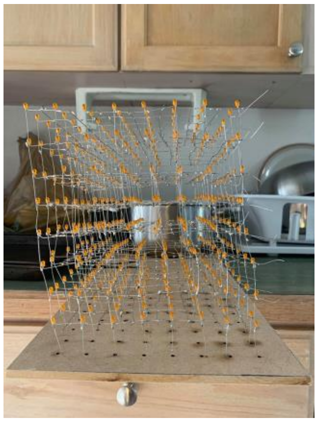
The main risk of using more ultrasonic sensors is the amount of crosstalk that will be received however, we should be able to reduce the amount by de-muxing and by carefully selecting which sensor is on or not. This is more explained in the following section.

For the LED Cube, we built an 8x8x8 cube. This means that there were 512 solder joints to solder. This means that for prototyping/testing of the Cube, we had to do it layer by layer, LED by LED. This ensures that not only each LED is correctly working, but each connection is good allowing us to multiplex each layer. This whole process is done by soldering each of the cathode together to form each layer, and then to solder each anode together to form the columns. Below are images of our Cube.

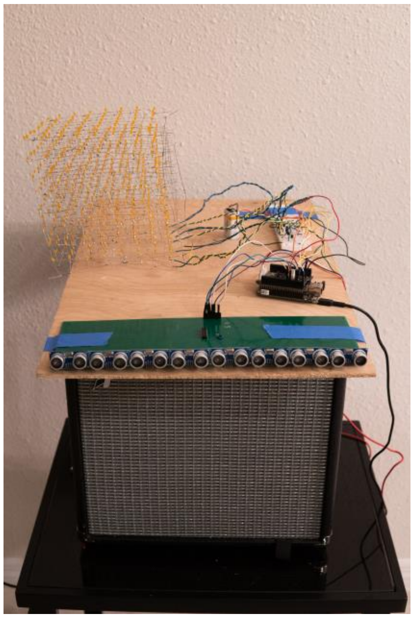


In the image above, two of the layers are connected. Following this pattern, we began to solder and connect each layer, making sure to test not only each LED but each layer and column, until the cube was successfully an 8x8x8 cube. The following pictures show the continuation of the build process, with the last picture showing the final cube.





Afterwards we began the process of soldering the protoboard that will be used to multiplex each individual row to indicate how close or far an object is.



After putting all the components, wires, and pieces in place we extensively tested our project multiple times to ensure that everything was working properly.

VII. CONCLUSION

Along our journey of designing this system a common theme cropped up regarding our project. Namely, the concept that image capture is the inversion of image reconstruction. In lay terms this seems like a painfully obvious heuristic but much like any great adage, its simplicity begets a deep understanding to fully comprehend. For every action we took to process, file, and record the data from our sensor array; the exact action in reverse must have been taken. This simple yin and yang were at first difficult for us to understand. We kept trying to design our input system and our output system differently, continually making the mistake of thinking that we had to have two vastly specifically and unique systems to handle output and input. After hours of deliberation over various I/O schema for images, it would become clear, at every level of our device, a simple realization. We found that often, the best input scheme is also the best output scheme. Once finally reaching this epiphany a lot of our design was able to be solved simply. Instead of having a task of designing two separate systems, our process became that of designing a singular system, and then inverting it for the output stage. Any realizations made while musing over either input or output would benefit the other and vice versa. This massively helped simplify our system at every level, from the main wiring and hardware layout to the software we needed to write in order to run everything. In its current state, Frames is more of a small-scale proof of concept of a general idea and technology. With ample budget from a client and time we could quickly scale the device to a room sized installation. In short, the scalability and adaptability of the Frames unit is clearly large. Initially conceived as a marketable system for entertainment, the concept as a whole has been developed enough to be easily adaptable to different technologies and the underlying concept has been realized to the point of utilization in different scenarios (such as night vision and 3D imaging through surfaces). We look forward to potentially marketing this device in its current form and believe that it would stand up too many of the major touring installations out in the industry currently. Without a doubt this project has been an extremely insightful trip through the design and fabrication process that at times taught us more about what it means to be an engineer than much of our collective university experience. We look forward to the start of our collective careers and the beginning of our path.

VII. AUTHORS

Christian Vartanian

Christian Vartanian is a senior at the University of Central Florida and will receive his Bachelor of Science degree in Electrical Engineering in May 2020. His interests are digital signal processing, audio processing, acoustical design, and music. He plans to pursue a career in this field after graduation.

Daniel Hureira

Daniel Hureira is a senior at the University of Central Florida and will receive his Bachelor of Science in Computer Engineering in May 2020. His interests include embedded systems, FPGA design, and digital systems design. He plans to pursue a career in the field of Embedded Systems after graduation.

Jamesley Rock

Jamesley Rock is a senior at the University of Central Florida and will receive his Bachelor of Science in Computer Engineering in May 2020. His interests include Object Oriented Programming, and Operating Systems. He plans to pursue a career in the field of Front-End Development after graduation.

VIII. REFERENCES

- “What Is USB (Universal Serial Bus)?” Computer Hope, 4 May 2019, https://www.computerhope.com/jargon/u/usb.htm.
- Alarco, Jose, and Peter Talbot. “The History and Development of Batteries.” Phys.org, Phys.org, 30 Apr. 2015, https://phys.org/news/2015-04-history-batteries.html.
- “USB History.” Premium USB, https://www.premiumusb.com/usb-history.
- “First US Detachable Electric Plug – Today in History: November 8: Connecticut History: a CTHumanities Project.” Connecticut History | a CTHumanities Project, https://connecticuthistory.org/first-us-detachable-electric-plug-today-in-history-november-8/.
- FutureLearn. “Persistence of Vision: How Does Animation Work?” FutureLearn, https://www.futurelearn.com/courses/explore-animation/0/steps/12222.
- Instructables. “LED Cube 8x8x8.” Instructables, Instructables, 26 May 2019, https://www.instructables.com/id/Led-Cube-8x8x8/.
- “ISO/IEC 14882:2003.” ISO, 1 Sept. 2011, https://www.iso.org/standard/38110.html.
- ISO, https://www.iso.org/obp/ui/#iso:std:iso:12224:-1:ed-1:v1:en.
- Anurag. “Raspberry Pi vs Arduino vs Beagle Board.” Cloud Computing, https://www.newgenapps.com/blog/raspberry-pi-vs-arduino-vs-beagle-board.
- *ISO*, https://www.iso.org/obp/ui/#iso:std:iso:12224:-1:ed-1:v1:en.
- “ISO/IEC 7816-12:2005.” *ISO*, 5 Dec. 2018, https://www.iso.org/standard/40604.html.
- “Multiplexed Display - the Basics.” *LEDnique*, http://lednique.com/display-technology/multiplexed-display/.
- FutureLearn. “Persistence of Vision: How Does Animation Work?” *FutureLearn*, https://www.futurelearn.com/courses/explore-animation/0/steps/12222.
- Smith, John SmithJohn, and John Smith. “3D Hologram and the Future of Technology and Entertainment.” *Business 2 Community*, https://www.business2community.com/tech-gadgets/3d-hologram-future-technology-entertainment-01214803.
- Weber, Marcus. “How Does a 3D Hologram Work? - Meee Services: Electronics Design.” *MEEE*, 4 Dec. 2018, https://www.meee-services.com/how-does-a-3d-hologram-work/.
- Klien, Asher. “Raspberry Pi vs. Arduino Vs. BeagleBone – What's The Difference?” *Vilros.com*, Vilros.com, 17 Jan. 2019, https://vilros.com/blogs/getting-started/raspberry-pi-vs-arduino-vs-beaglebone-what-s-the-difference.
- Aggarwal, Param. “Mapping A Room Wh Ultrasonic Distance Sensors.” *Hackster.io*, 7 Jan. 2018, https://www.hackster.io/paramaggarwal/mapping-a-room-wh-ultrasonic-distance-sensors-9725b7.
- *YouTube*, YouTube, https://www.youtube.com/watch?v=z8ntr5lYxGU.
- Instructables. “2D Ultrasonic Mapping.” *Instructables*, Instructables, 3 Oct. 2017, https://www.instructables.com/id/2D-Ultrasonic-mapping/.
- “First US Detachable Electric Plug – Today in History: November 8: Connecticut History: a CTHumanities Project.” *Connecticut History | a CTHumanities Project*, https://connecticuthistory.org/first-us-detachable-electric-plug-today-in-history-november-8/.
- Alarco, Jose, and Peter Talbot. “The History and Development of Batteries.” *Phys.org*, Phys.org, 30 Apr. 2015, https://phys.org/news/2015-04-history-batteries.html.
- “What Is USB (Universal Serial Bus)?” *Computer Hope*, 16 Nov. 2019, https://www.computerhope.com/jargon/u/usb.htm.
- “Political Constraints.” *Consequential LCA*, https://consequential-lca.org/clca/marginal-suppliers/political-constraints/.
- “Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements.” *The Journal of Object Technology*, http://www.jot.fm/issues/issue_2004_03/column3/.
- Google. “Google/Hilbert.” GitHub, 22 Nov. 2018, https://github.com/google/hilbert?fbclid=IwAR2j5-JaX_fMjesdWXo44TPAM4SGkEKX9SgRvTMYrnFkcAaLbYlaF2cWo0A.
- Ruohan, Gao. *2.5D Visual Sound*. CVPR, http://openaccess.thecvf.com/content_CVPR_2019/papers/Gao_2.5D_Visual_Sound_CVPR_2019_paper.pdf.