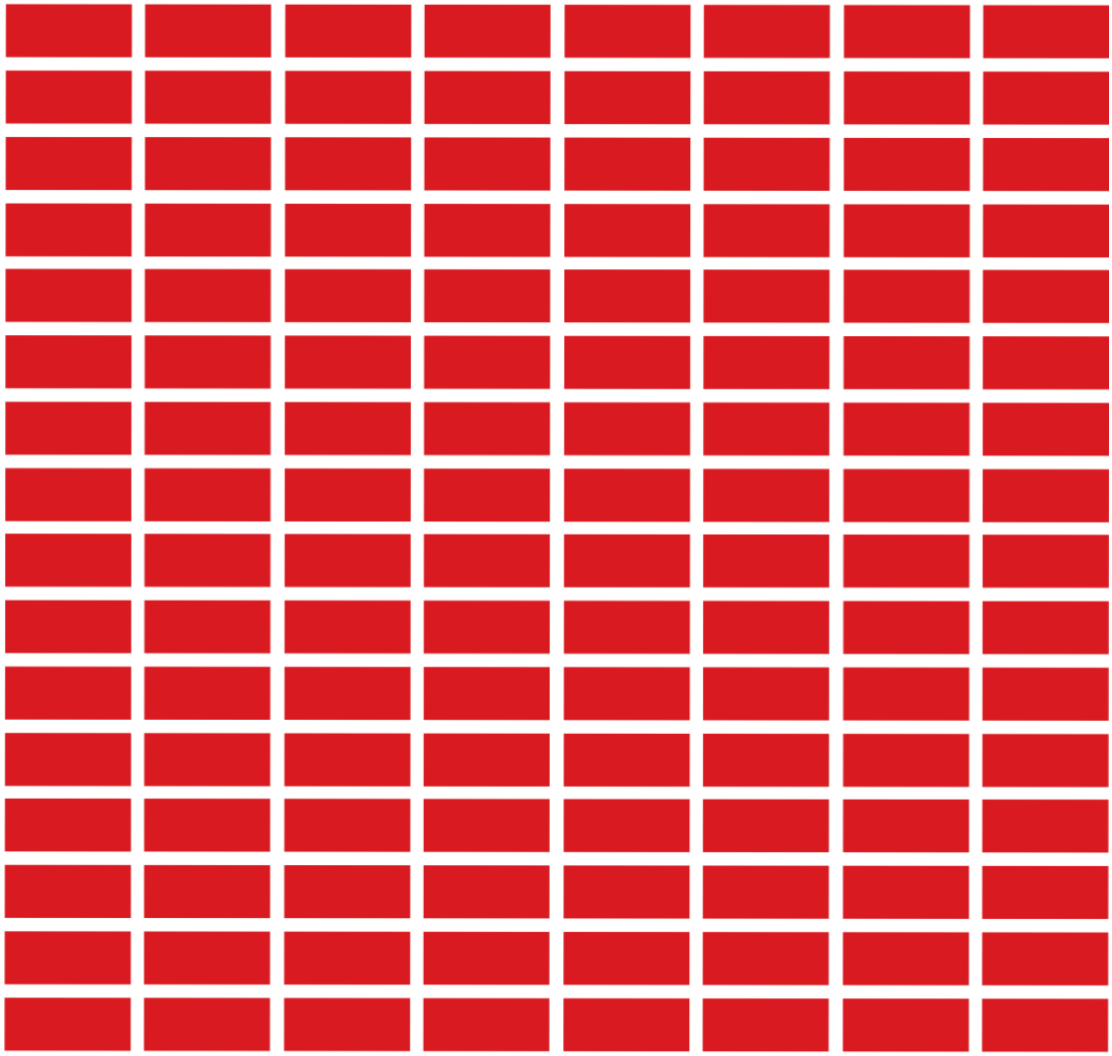

Frames:

A 3-D Image Recording and Display System

Group: 7

Daniel Hureira [CpE], Christian Vartanian [EE], Jamesley Rock [CPE]

Objective:

Our aim is to build a lightweight, relatively cheap, display system that can capture and display ultrasonic texture maps of a space or object. 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.

Function:

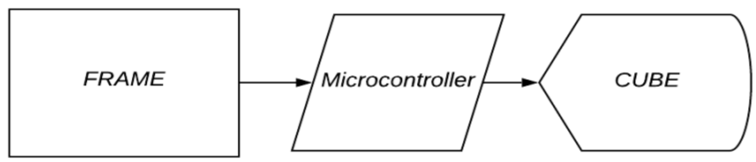

Figure 1

The system is made of three 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 are able to 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. 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.

After being captured by the Frame, it's the job of the middleware running on the microcontroller to quantize the incoming distance readings into discrete levels that can then be displayed by the Cube. The X and Y components of the output have already been discretized as they are the literal X and Y positions of each sensor on the matrix; this leaves the only remaining data that needs to be quantized to be the Z axis. This would be a relatively simple process if only being implemented on a single row of lights, but upon considering the scale of even a single Frames worth of data and its respective LED requirements (a 16x16 Frame would require 4,096 LEDs) it becomes obvious that a form of edge detection and compression is crucial to limiting the build out from becoming a mess of multiplexers and an overall wiring nightmare.

The *Cube* is a volumetric array of LEDs that are able to take serialized and quantized X,Y,Z data and build a realtime image map of said data. Each LED will be connected to each other in a way that allows each row to be selected and then specific LED’s within that row. In its current design each Z axis row of the Cube corresponds exactly to an individual sensor on the Frame. This layout, while potentially complex to scale, allows the system to have a 1:1 representation of any objects in front of the Frame. As our incoming data is serialized we will be at the whim of frame rate issues and will need to guarantee a high enough frame rate in order to fluidly display the image. In lay terms it's a three dimensional display that displays the world the way bats see it, using echolocation.

Certain core functions are still being discussed, as should be expected from any project in its early stages. While the general I/O pathway is clear, the actual mechanisms for handling the data is not. We are currently investigating computer vision programs to better understand image compression and edge detection, as the more digital compression is able to be performed the more scalable the system as a whole can become. Aside from the middleware running on the microcontroller, the question of modularity and interconnectivity has come up. In theory we should be able to create a building block like system for attaching multiple sets of frames and cubes together to create larger displays and capture devices. By packaging both the Frame and the Cube as modular blocks we should be able to rapidly scale the system to fit the needs of the space it is deployed in.

Project Constraints:

Some constraints for our project include:

- HDMI Standards
- WiFi Standards
- Resolution Density
- Voxel Density
- Power Standards

Requirements and Specifications:

↓ = Negative Correlation (i.e. as power consumption is increased, efficiency is decreased)

↑ = Positive Correlation (i.e. as the quality of image increases the cost also increases)

	Efficiency	**Frame Rate** (Goal: 30 FPS)	**Quality of Image** (Goal: Able to accurately display an image)	**Latency** (Goal: < 1 minute)
Portability (Goal: Able to be able to be transported easily)	↑	↓	↓	↓
Power Consumption (Goal: Not yet set)	↑	↑	↑	↑
Weight (Goal: < 50 lbs)	↓	↑	↑	↑
Size (Goal: Able to be transported easily)	↓	↑	↑	↑
Cost (Goal: < $1000)	↑	↑	↑	↑

Table 1

Flowchart/Work Distribution:

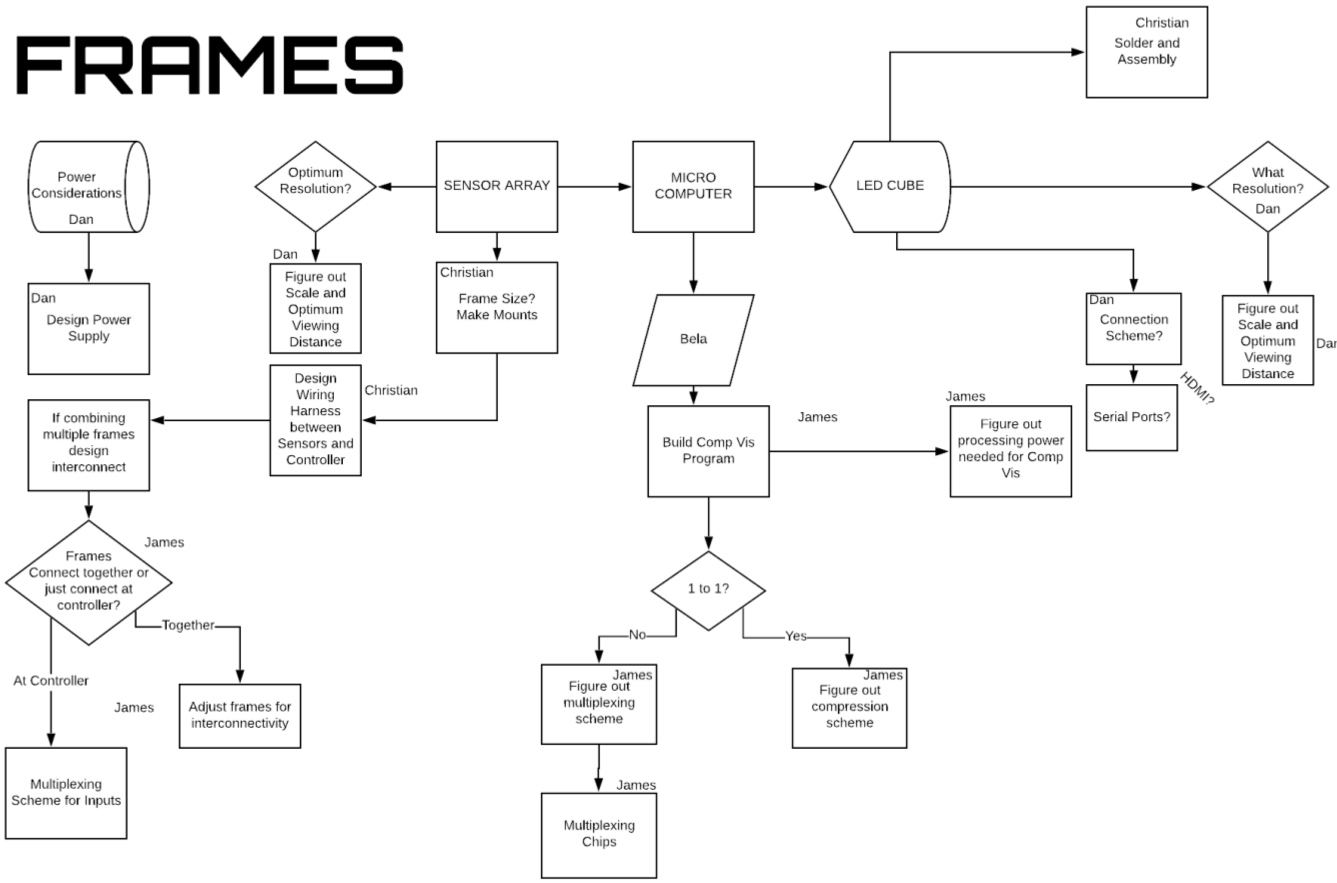

Figure 2

Estimated Project Budget:

Range Finder HC-SR04	**$.60 * 64**	**$38.40 Per Frame**
LED's	**$0.10 * 512**	**$51.20 Per Frame**
Micro-Computer Bella/BeagleBone	**$150 * 1**	**$150**
Spare Electronics for running Cube and Multiplexing the Frames		**$75**
Frame and Hardware for mounting Range Finders		**$10 Per Frame**
Per Frame Cost		**$149**
Total		**$324**

Table 2

Project Milestones:

Fall 2019:

- **September**: Determine project, write Divide and Conquer V1, meet with professors to get the OK for project design
- **October:** 60 Page Draft
- **November:** 100 Page Submission
- **December:** Final Submission

Spring 2020:

- **January**: Start working on building the project
- **February:** Testing, debugging project
- **March:** Refining, finishing touches/adjustments/tweaks
- **April:** Final Demonstration