HoloFan

Alex Charters, Brandon Chen, Emmanuel Lotubai

Bios

Brandon Chen

Brandon Chen is in his third year studying Computer Engineering at the University of Utah. Having the ability to adapt quickly while being resilient has helped him achieve success through college. He is also a member of the University of Utah Men’s Club Soccer team for three years. Balancing academics with soccer, Brandon likes to spend the rest of his free time enjoying nature with different types of outdoor activities.

Emmanuel Lotubai

Emmanuel Lotubai is in his fourth year studying Computer Engineering at the University of Utah. He is expecting to graduate with a BS degree in fall of 2021. In his career, Emmanuel has focused on software development. In this project, he hopes to improve his hardware understanding by working closely with his team. Emmanuel's life long goal is to use the skills he acquires to help people.

Alex Charters

Alexander Charters is in his fourth year studying Computer engineering at the University of Utah. He has been a member of the Utah Student Robotics team for three years and has been the electrical lead for two of them. He is actively researching Spread Spectrum Time Domain Reflectometry (SSTDR) with the U's own Professor Cynthia Furse.

Presentation Outline

- Project Objective
- Project Challenges
- List of Monetary Items
- Execution Plan for Next Semester
- Current Prototype
- Mechanical Overview
- Electrical Overview
- Software Overview
- Graphics Overview

Project Objective

- What are we trying to accomplish?
  - Using the concept of persistence of vision, we will create a 3D image with a high speed fan
  - Will track the main user and update the angle of the fan to display the image towards one's direction
- Why should people care?
  - Fun gadget that can be scaled up/down for advertising, decoration, or alternative for VR
- Where is our project website?
  - https://alexcharters.github.io/Senior-Design-Website/

Project Challenges

Software

- Must develop scanline algorithm for rotary display
- Must determine how to interface with game engine or computer graphics software
- Must implement computer vision

Hardware

- Must precisely control speed and position for both motors controlling fan direction and spin
- Need to source Slip Ring for providing electrical connections to spinning parts
- Must use a sufficiently fast controller for writing to LED's

Monetary Items

Items for Prototype

Item	Cost
Raspberry Pi 4	$35
Webcam	$9
Aluminum Sheet	$36.96
Wall Adapter	Sourced free from professor
Extraneous Mechanical Components	Sourced free from professor
8020 extruded aluminum	Sourced free from professor
Monitor	Sourced free from professor
Stepper motor	Sourced free from professor
Stepper Motor Controller	Sourced free from professor
Total	**≈ $81**

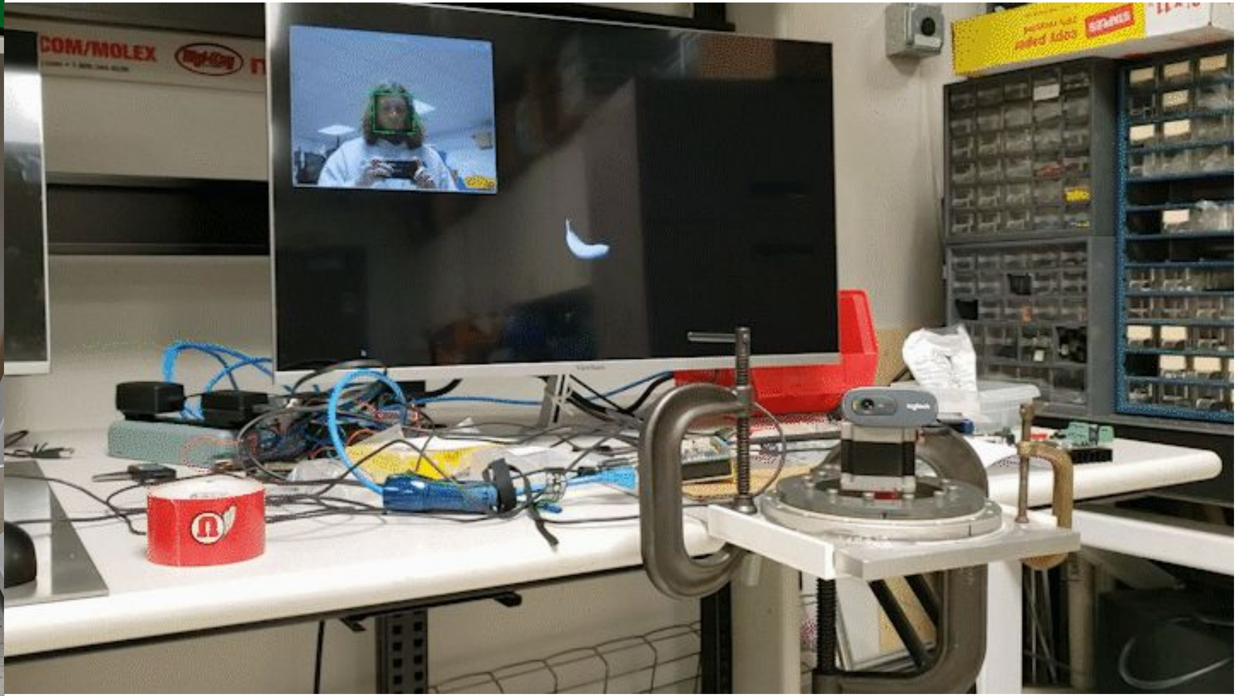
Items for Final Design

Item	Cost
LED Bar	$75
Brushless Motor w/ encoder	Sourced free from professor
Stepper motor	Sourced free from professor
Slip Ring	$122
Stepper Motor Controller	Sourced free from professor
Brushed Motor Controller	Sourced free from professor
Raspberry Pi 4	Sourced free from professor
Webcam	$9
Wire	Sourced free from professor
Connectors	$15
Housing/aluminum	$50
Wall Adapter	Sourced free from professor
Total	**≈ $271**

Execution Plan for Next Semester

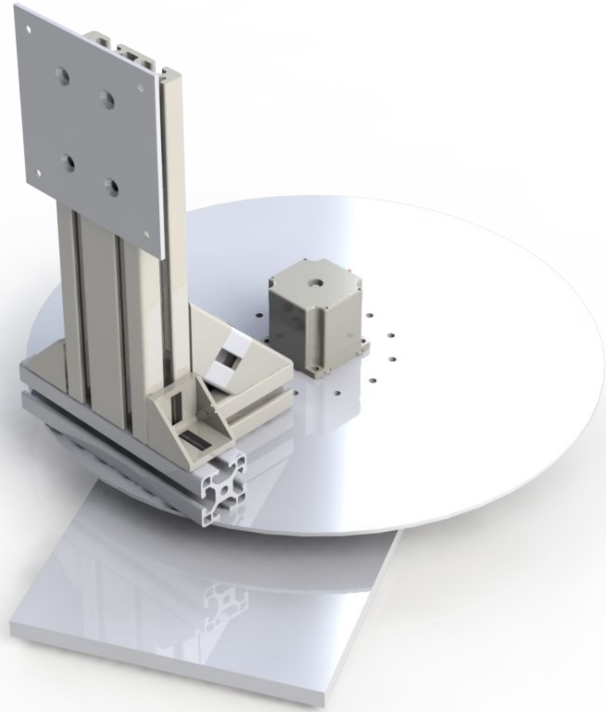
- Displaying graphics using the high speed fan
- Redesign the base to be able to attach the fan instead of prototypes monitor
- Implement slip ring for the high speed fan
- Implement a fiducial for determining who is the “main user”
- Build housing for raspberry pi and all connections
- Smooth stepper movement

Prototype



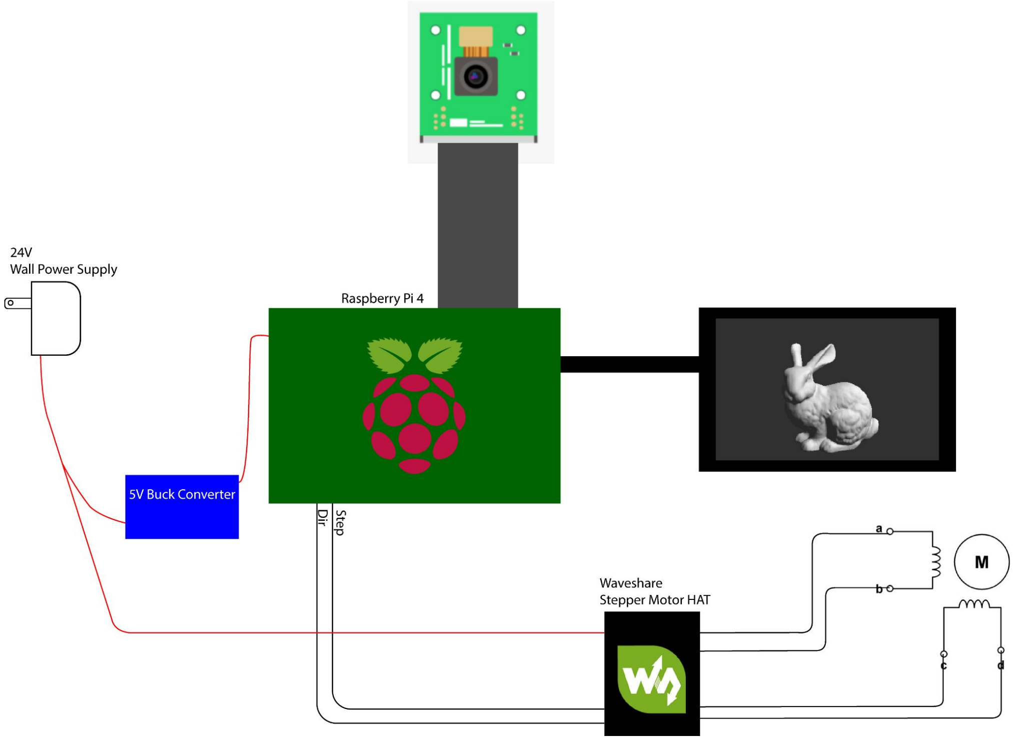
Mechanical

- 80/20 aluminum extrusions
- ⅛” aluminum sheet
- NEMA23 Stepper Motor
    - 7.2 kg-cm holding torque



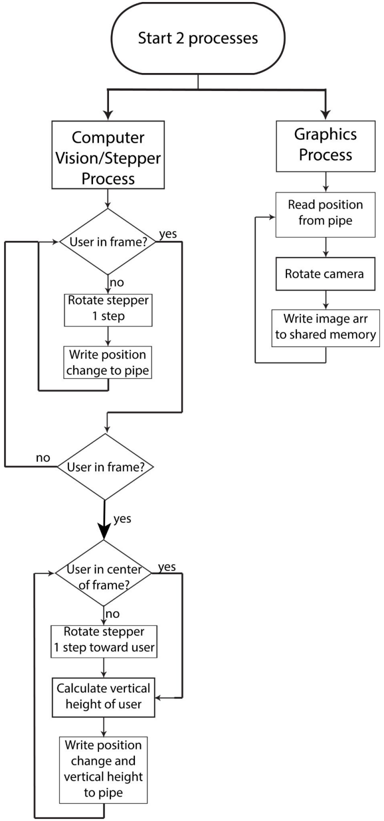
Electrical

- Computation: Raspberry Pi 4
  - More readily handles computer Graphics
  - Very fast SPI communication for future use with addressable LED light bar.
- Waveshare Stepper Motor HAT
- SM57HT51-3006A Stepper
  - Wired bipolar (full coil configuration)
- Webcam
- HD Monitor



Software

- Computer Vision/Stepper Process
  - Python
  - Locate the face, rotate the stepper motor towards the face.
- Graphics Process
  - Python 3D Graphics
  - Display the specified figure

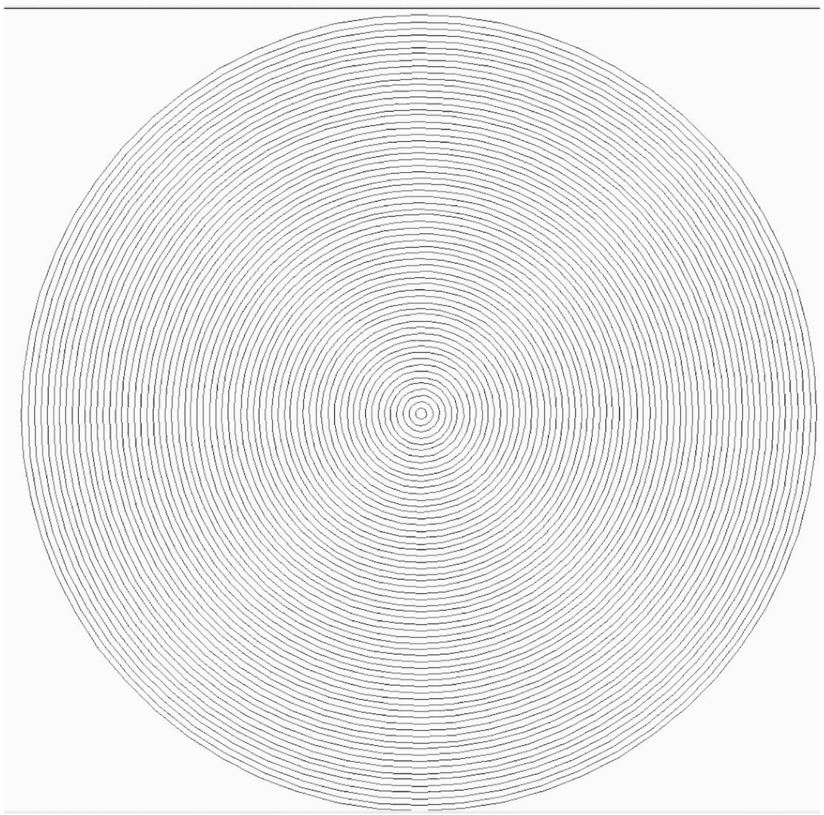


Graphics (For Next Semester)

- Fan scanline algorithm already completed
- Not used in current prototype
- Based off a modified version of the Bresenham line algorithm

pi@raspberrypi: ~/Desktop/holofan

```
from numpy import asarray
from PIL import Image
import numpy as np
import math
from Profiler import profile

def get_pixels(deg, arr, POINTS):
    h, w, channels = arr.shape
    midw = w/2
    midh = h/2
    mindim = min(w, h)
    midmindim = mindim/2

    rad = deg*math.pi/180
    m = math.tan(rad)
    b = midh-m*midw
    # print("y="+str(m)+"x+"+str(b))
    pixels = np.zeros((POINTS, 3), dtype=np.uint8)

    a = midmindim * math.cos(rad)
    o = midmindim * math.sin(rad)

    #rounding
#    wpoints = enumerate(np.rint(np.linspace(midw-a, midw+a-1, POINTS)).astype(int))
#    hpoints = enumerate(np.rint(np.linspace(midh-o, midh+o-1, POINTS)).astype(int))

    #truncating
    wpoints = enumerate(np.linspace(midw-a, midw+a-1, POINTS).astype(int))
    hpoints = enumerate(np.linspace(midh-o, midh+o-1, POINTS).astype(int))

    if abs(m) < 1:
        for idx, _x in wpoints:
            x = _x
            y = int(m*x + b) #truncating
#            y = round(m*x + b) #rounding
            pixels[idx] = arr[y][x]
    else:
        for idx, _y in hpoints:
            y = _y
            x = int((y-b)/m) #truncating
#            x = round((y-b)/m) #rounding
            pixels[idx] = arr[y][x]
    return pixels

@profile
def main():
    image = Image.open('/home/pi/Desktop/Project/yeet.jpg')
    arr = asarray(image)
    POINTS = 128
    for deg in range(180):
        line = get_pixels(deg, arr, POINTS).reshape((1, POINTS, 3))
        #thickline = np.tile(line, (30,1,1))
#        img = Image.fromarray(line, 'RGB')
#        img.save('/home/pi/Pictures/slice'+str(deg)+'.jpg')

if __name__ == "__main__":
    main()
```



Questions?