Smartphone-Based Video Conferencing System

Joseph Lagasse, Travis Canfield, Patrick Scott, Department of Electrical and Computer Engineering, University of New Hampshire

University of New Hampshire

Abstract

The purpose of this project was to bring the atmosphere of a face-to-face smartphone Zoom call to a conference setting. To achieve this, our proposed directional dock on which the smartphone is placed will automatically optimize the orientation of the phone during a video conference meeting or a video call meeting. Our proposed direction dock will contain four microphones that will detect the location of the speaker and turn the dock accordingly such that the phone will face the general direction of the subject. A mounted camera will then use facial detection to center the subject in the frame and make minor adjustments to the phone's orientation. Our design involves two stepper motors, one for rotating the platform and the other for adjusting the pitch of the phone and the external camera.

Objectives

- Achieve low-cost functional alternative to similar, but more expensive commercial conferencing products.
- Optimize and refine microphone based triangulation.
- Implement real time facial detection software to further improve camera centering.
- Achieve desired combined functionality.

Contacts

Patrick Scott - pfs1005@wildcats.unh.edu
Travis Canfield - twc1004@wildcats.unh.edu
Joseph Lagasse - jl1231@wildcats.unh.edu
Faculty Advisor:
Edward Song - Edward.Song@unh.edu

Methods

- **Audio Detection:**
  - Utilized four microphones in a 360° array, each positioned at 90°
  - With the microphones in this position we are able to detect sound from any angle
  - We were able to configure the noise threshold so background noise did not trip the sensors
- **Facial Detection:**
  - Facial detection was performed on Raspberry Pi 4 to minimize delay in the detection process
  - The OpenCV computer vision library utilizing Haar-like features within the Viola-Jones Framework was used to accurately detect any humanoid face within a picture frame
  - Post analysis coordinates then transferred to arduino via serial to provide positioning adjustments
  - The facial detection works by first evaluating a grayscale image and evaluating whether Haar-like facial features are present. Haar-like features can be described by the gradient patterns found in all human faces. For example the sides of someone's nose are generally a darker shade than the front part of the nose. After confirming the existence of various Haar-like features and passing them through the Viola-Jones framework it can be determined whether a face is indeed present in the frame.

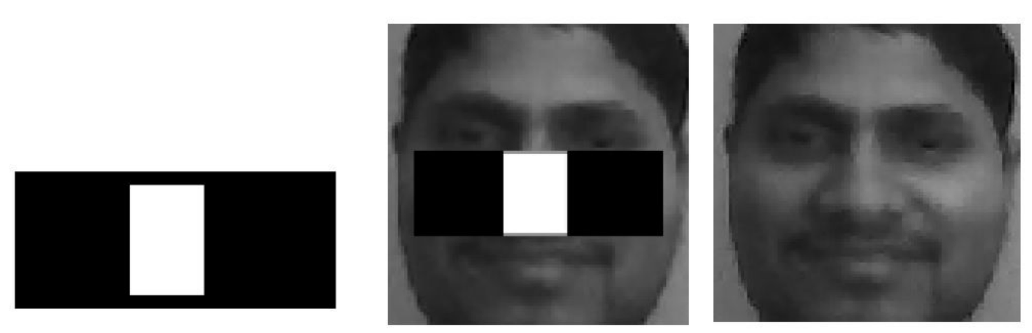

Fig 1. Example of Haar-like features

- **Dock Assembly:**
  - Microphone mounts and cones for triangulation.
  - Stepper motor mounts and attachments for turning base/phone pitch.
  - Raspberry Pi and camera mount on back of phone mount.
  - Design in SolidWorks and 3D print using PLA.

Results

- **Dock Assembly:**

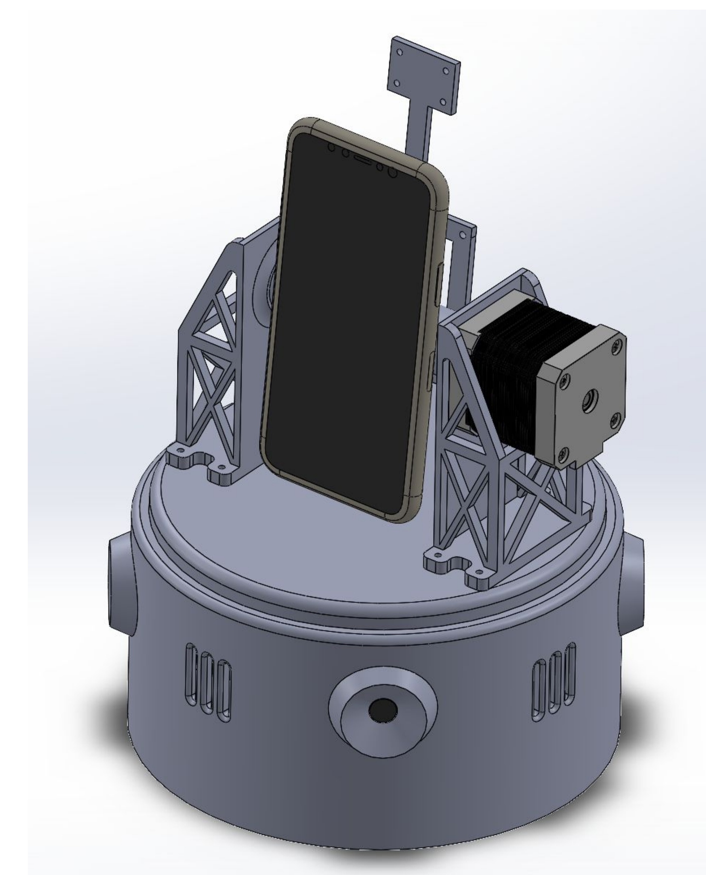

Fig 2. Dock Assembly in SolidWorks

- **Final Dock Prototype Internal:**

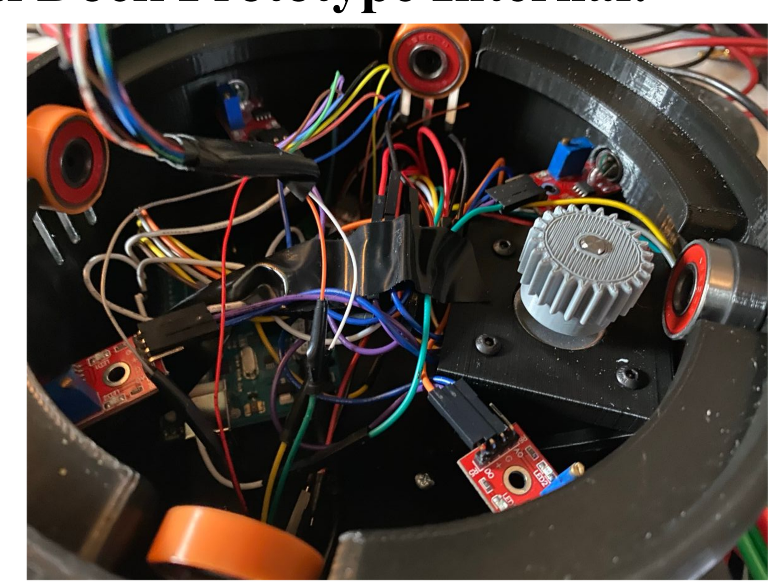

Fig 3. Final Dock Assembly

- **Final Dock Prototype External:**

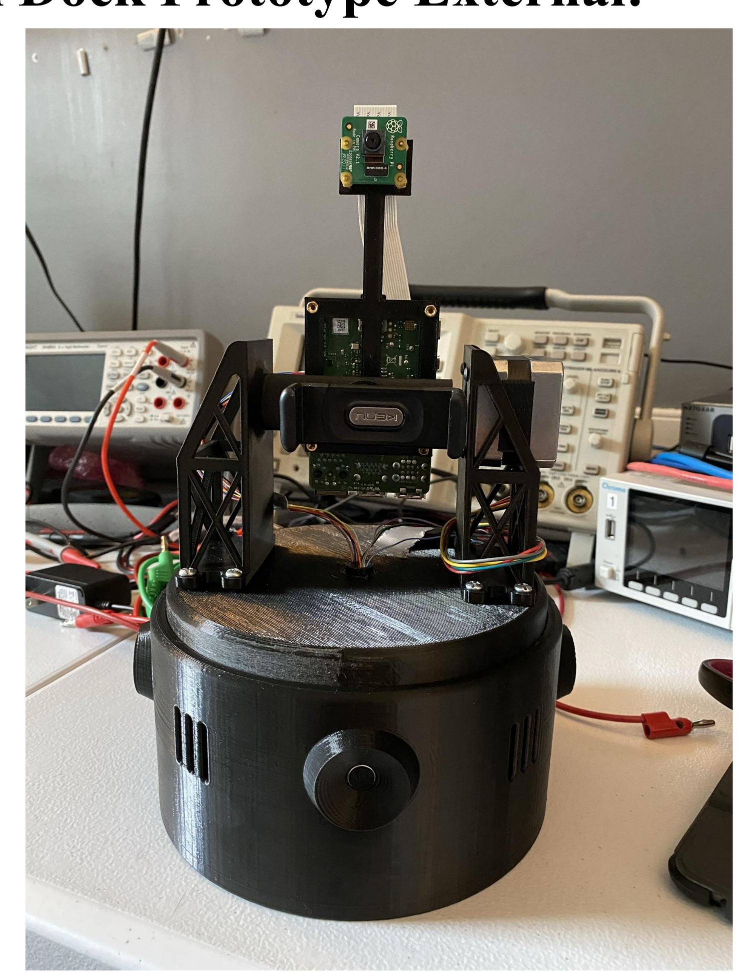

Fig 4. Final Dock Assembly

Conclusions

- Our prototype is functional and is a promising start towards a final product
- Audio detection correctly locates audio sources
- Facial detection correctly locks onto the speakers face and sends location data
- The location data gets interpreted and the dock follows the speakers face
- Our prototype is up to 76% cheaper than current market devices

Next Steps

- Add more sound damping features to reduce operating noise.
- Integrate all processing and I/O onto one integrated circuit.
- Reduce size/price of system.
- Use higher quality hardware for true audio triangulation, not just detection.
- Improve facial detection to reduce false positives
- Utilize a multi-core processor to simultaneously read all microphones at the same time, instead of one by one

References

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Ivancic, Kristijan. "Traditional Face Detection With Python." *Real Python*, Real Python, 7 Nov. 2020, realpython.com/traditional-face-detection-python/.