Progress Report: Team 29

Gaze and Speech Tracking with Haptic Feedback for the Visually Impaired

Hannah Burke – Zeen Luan – Nicolas Chapman

Working Principle and Technical Limitations

Our product aims to track people in the camera field of view, compute if these people are looking at the camera and if they are talking, and communicate this information to the user via a vibrotactile device. The camera, attached to a servo-controlled head mount with pan and tilt capability, will scan the environment surrounding the user and track people who are talking and looking at the camera. The location of these people with respect to the user will be communicated via five vibrating motor disks arranged evenly around the head.

The control of the camera orientation using servo motors and the Raspberry Pi has been implemented. When combined with basic image processing tools and an optimised PD controller, the camera successfully tracks the position of an object accurately. Intensity control of the vibrating motor disks via a PWM signal has also been successfully tested. The system and circuit diagrams of the device are seen in Figure 1 and Figure 2.

The challenge moving forward lies in the implementation of the computer vision algorithms for recognising gaze direction and speech. Our product requires multiple machine learning algorithms to be run (face recognition, gaze recognition, mouth feature detection etc.) on a single face. Repeating this process on multiple faces in a crowded environment results in a large computational load. Initial tests of numerous open source gaze and facial feature recognition algorithms inform that the Raspberry Pi could not execute these algorithms fast enough to maintain real time camera tracking of people. To solve this, the team has implemented a communication protocol (see Figure 1) between the Raspberry Pi and a computer connected to the same LAN. Sending images to the computer incurs a fixed upload and download time, but ultimately enables complex algorithms to be run far more efficiently. Initial tests showed a vast improvement in the number of frames per second that the proposed algorithms could be applied to.

The second challenge faced has been the distance and lighting limitations of gaze recognition algorithms. Due to camera resolution, the algorithms tested appear only to be accurate to about 1.5 metres and in good lighting. More complex algorithms which incorporate deep neural networks have been shown to address these limitations, but they may further inhibit computing power and be difficult to integrate. Another solution is to use head orientation in conjunction with gaze direction to infer speaker attention. Even simple head orientation algorithms were tested to have a range of up to 3m while working in more varied lighting conditions. However, head orientation is a less accurate indicator of attention. More details of the proposed algorithms to be used can be found in the Machine Vision Algorithms section.

Lastly, the device faces a physical limitation in that it must be powered from a wall socket. Battery solutions can be simply implemented on the Raspberry Pi, but doing so will exceed the budget limitations. Thus, in this iteration of the device the user must stay within range of a wall socket, limiting it to indoor use.

Machine Learning Algorithm

Our camera tracking product is intended to be an aid for guiding blind people to the most likely speaker in the field of view. Thus, the device prioritises tracking of ‘speakers’ over ‘potential speakers,’ and ‘potential speakers’ over ‘non speakers’. Thus, we will implement the following priority algorithm:

- A person looking at the camera and speaking (speaker) will be tracked over those only looking at the camera (potential speaker) and those doing neither (non-speaker).
- A person looking at the camera (potential speaker) will be tracked over those not looking at the camera (non-speaker).If there are two people in a frame of the priority to be tracked, the closest person to the user will be selected for tracking based on the size of their face

We therefore require a series of face, gaze and speech recognition algorithms to identify speakers. One resource for face recognition, eye tracking and mouth movements that we plan to use is https://github.com/vardanagarwal/Proctoring-AI. This uses OpenCV and the Python package dlib to recognise faces and identify key landmarks on the face (including mouth and eyes). Eye tracking is performed by thresholding the eye area and locating the centre of the pupils. Mouth movements are detected by looking for change in the height of the lips.

This resource also has the option for detecting the direction that a face is turned, which we will use in conjunction with the eye tracking to better identify potential speakers. This algorithm works by again finding the landmarks on the face and comparing the location of the nose to the rest of the facial features. Due to the resolution of the camera, we might require face direction for people who are further away from the user, or if their eyes are covered by glasses.

However, our testing has not confirmed that the above eye tracking algorithm will be accurate enough for our requirements. For gaze detection, we instead might use https://github.com/antoinelame/GazeTracking, which also uses OpenCV and dlib. This algorithm is more accurate as it analyses the eye region of the face in more detail, obtaining information about both the iris and the pupils. This allows it to specify a more precise angle of gaze direction, enabling a more accurate prediction of speaker attention.

System Diagram

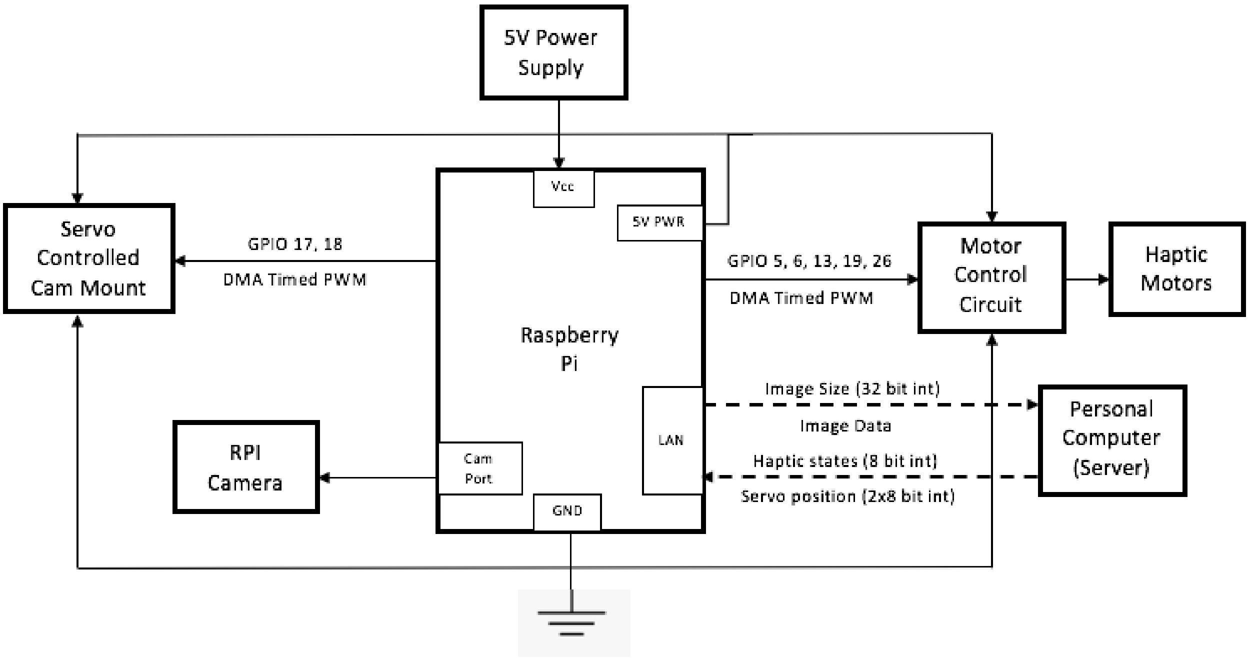

Figure 1: system diagram of device. Note that the data transfer between the Raspberry Pi and Personal Computer occurs over LAN using the Python socket networking interface. The image size followed by image data is uploaded to the PC server from the Raspberry Pi, and if an image size of 0 is received the link is terminated. The PC applies the computer vision and PD controller algorithms to the received image data, and sends back the change in servo position and haptic states required.

Circuit Diagram

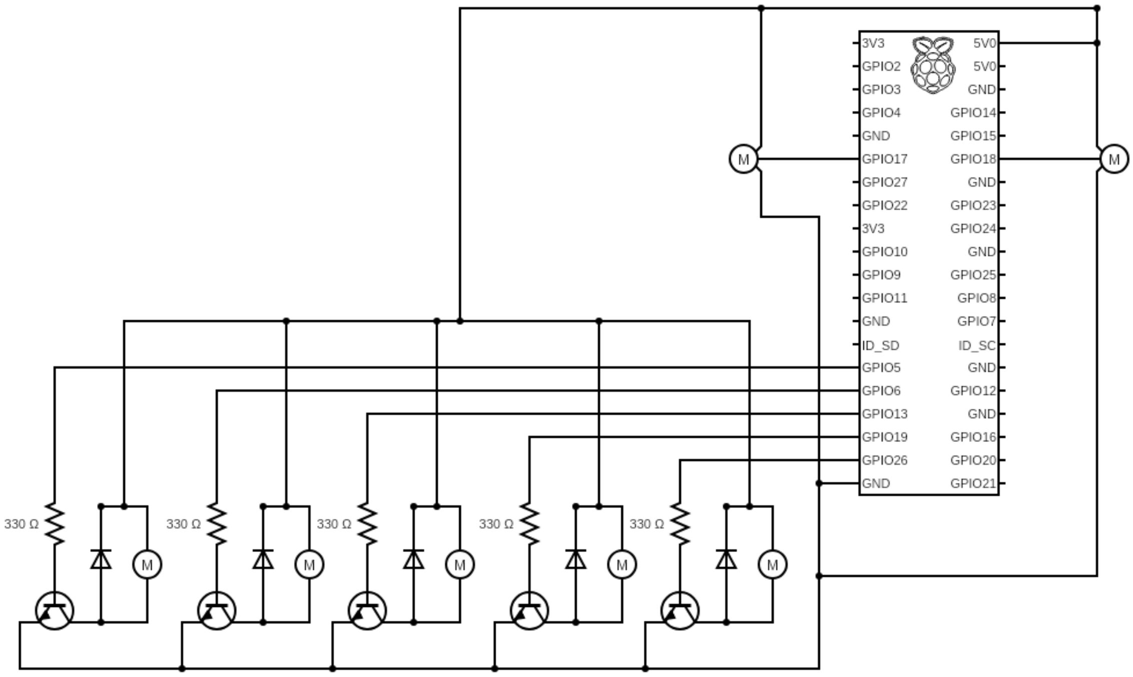

Figure 2: circuit diagram of device, consisting of PWM controlled motor driver circuits and the servo-motor configuration required for camera pan-tilt control.

CAD Models of Mechanical Design

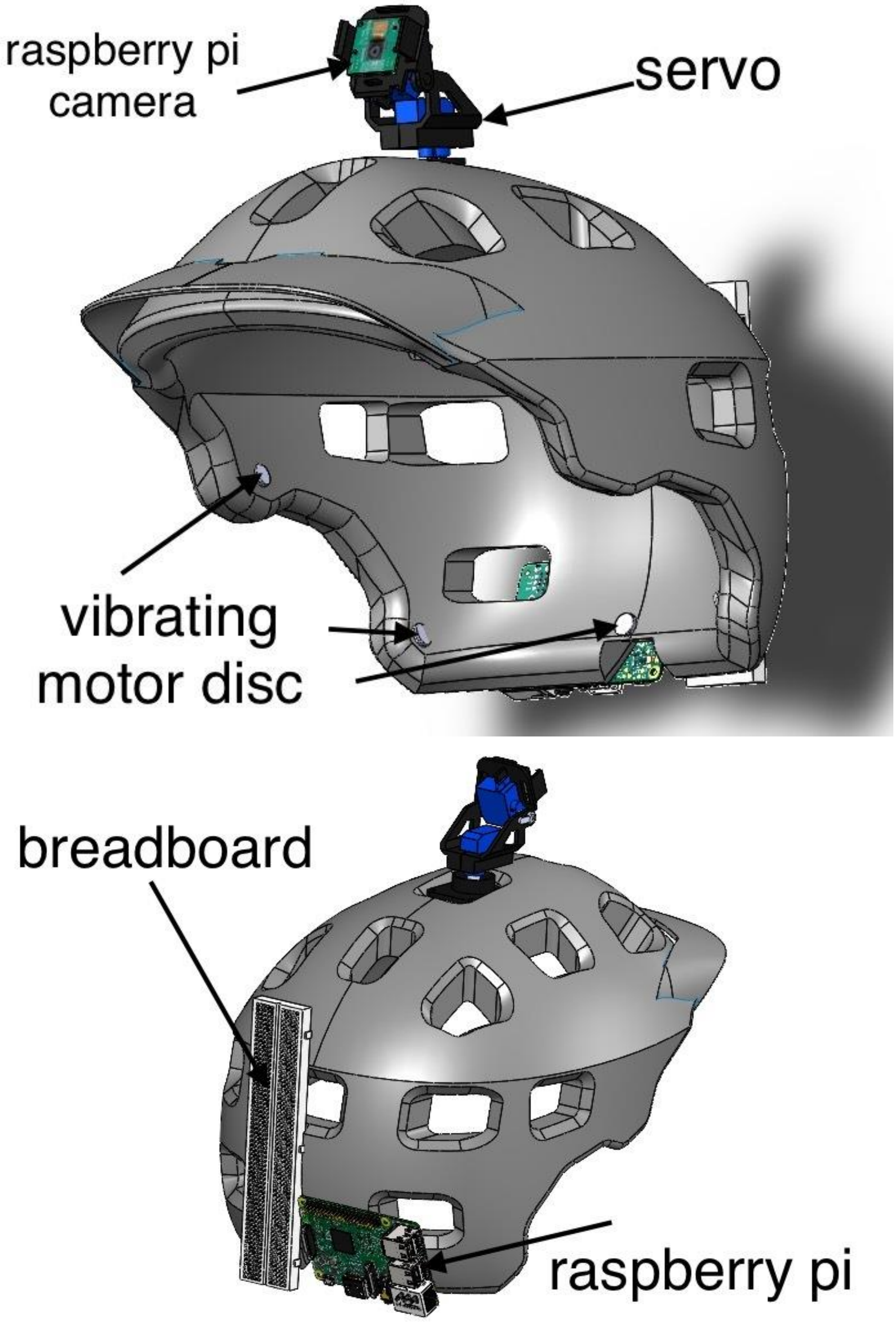

Figure 3: CAD design of the device. Due to lack of access to 3D printing services, a standard bike helmet is to be used instead of a 3D printed head mount. The camera, breadboard and Raspberry Pi will be mounted on thin pine using M3 screws and then attached to the helmet using a hot glue gun. The vibrating motor disks will be attached to the inside of the helmet using Velcro, making their exact position adjustable to the needs of different users.