Gaze Calibration Tool

Background: When left unattended, robot abuse is inevitable. In 2018, Marynel Vazquez and her lab conducted a study on bystander interventions during robot abuse. They found that humans were more likely to intervene and stop robot abuse, when the robot responded to the abuse with human-like “emotions”. Bystander intervention was less likely when the robot did not react to the abuse. This means that the more “emotional” a robot reacts, the more likely it is to be treated humanely. A robot’s “emotional” connection with a human can also lead to a variety of other long-term benefits. Upon further research we’ve learned that a lot of human emotions are conveyed through the eyes (ie; eye contact, eye expressions, etc.). Thus, this project is focused on creating a package that will universally make robot gaze appealing to humans.

Abstract: We hypothesize that the way gaze is rendered by a robot with a screen face needs to change as a function of the expressions that it renders to be considered appropriate by users. We also hypothesize that when multiple people...
interact with a robot with a screen face, they may perceive different gaze directions due to the mona lisa effect. In order to combat this effect, I built a package that allows people to more easily calibrate the representation of a robot's gaze. This package consists of several Ros Nodes that render a gaze based off a user’s distance from the camera, face position and face size. The package then allows users to adjust gaze until they deem it appropriate from where they are standing. Thus, this package allows users to calibrate a robot’s gaze to render appropriately as per feedback received from human-robot-interactions.

This package was developed to create a process through which gaze can be easily calibrated for a robot or virtual character with a face rendered on a screen. This will allow any party interested in studying human-robot interaction to eliminate the Mona Lisa Effect from their test results. It will also allow people to more easily render appropriate gaze based off of a person's face size and distance. The gaze-rendering package combines the use of a depth perception camera, gamepad controller and a robot face to calculate an appropriate gaze-position based off of user input and face location. By requesting user-input, the gaze-rendering package can be used to learn what gazes seem the most appropriate for specific face shapes, sizes and depths.
**Project Origins:** This package is built off of a set of processing eyes developed by Marynel Vazquez and Neha Govil. Originally the eyes were hard-coded to work accurately only with a laptop computer. They were also hard-coded to move based off the coordinates produced by a face-detection package. The original code, also had a hardcoded face distance and size.

**Procedure:** In order to create a package that could be universal across robot set-ups, I worked on a series of camera calibration documentation and scripts. First, we defined the camera needed to run this project as a stereo camera. In our lab, I use a ZED stereo camera, however the tools included in the gaze-rendering package will work for any stereo camera. Building off of the work I had done last semester, I created a gaze_master script with the help of Professor Vasquez. This script receives position info of the users face from the ZED camera. It also receives input from the PS3 controller. In the gaze_master script, I process the given data to create left_eye and right_eye vectors that are then passed to the processing script. Within the processing script, the vectors are transformed into the left and right eyes. The user can then move the left joy controller in order to adjust the eye position until they deem the rendered gaze appropriate. A visual representation of the gathered inputs and nodes is below.
**Usability Tests:** I conducted a usability test using gaze-master only listening to joy. Users were confused because if they moved a little, after adjusting the gaze with joy, the eyes would also shift and they’d have to readjust. They said it wasn’t clear how they could express being done, adjusting the gaze.
**Next Steps:** I intend to conduct usability tests with more participants next year, that incorporate the face poses subscriber too. I will also add a feature that allows users to press a confirmation button when they deem the robot gaze appropriate so that we can eventually build a package that learns from user input. This new feature will also allow users to receive a sense of confirmation that their gaze feedback was received.

**Acknowledgements:** Special thanks to Neha Govil and Marynel Vazquez for writing the original processing and gaze_master scripts. Thanks to Tim for creating the original Face Poses node.