To: Dr. Satici, Nardos Ashenafi, Brian Higgins

From: Robotic Vision Team: Daniel Pullicar, Mason Cannon, Roscoe Ambrose, Haston LaGrone

Subject: Biweekly Update, 10/30 – 11/13

Date: November 13, 2022

Happy November professors. This memo is going to serve as a short update for the Robotic Vision Team’s progress over the weeks of 10/30 – 11/13

# Work Completed

The team has shifted their focus from detecting the puck to individually assigned goals pertaining to either detection method, user interface, or light filtering tasks. The initial test of the program’s accuracy while tracking the puck was promising, but there is a significant issue with glare on the table from one of the ceiling lights.

Daniel Pullicar:

* Got blob detection up and running
* Optimized blob detection parameters / investigated pros and cons of this detection method
* Did not address locating puck from off-angle using image transformation. Was advised by Dr. S to try OpenCV’s calibration function first.
* Began investigating diffused lighting solutions (for hard light reflections problem)
* Set up GitHub branch

Mason Cannon:

* Practiced more with OpenCV and learning all the commands that are offered.
* Learned more about how the HoughCircles function actually works when looking for a circle.
* Learned about different filters that can be applied to help detect a puck. This includes filters that detect edges (aka curves on the puck) and ones that help reduce noise in the frame.
* Setting up a GitHub account and getting access to the group work.

Roscoe Ambrose:

* Built a GUI with several sliders controlling various settings in real time
  + Circle detection algorithm settings
    - Blurring, ratio of resolution, and Minimum distance between circles (more to come)
  + Camera settings
    - Brightness, contrast, saturation, hue, gain, exposure, frame width, and frame height
* Set up GitHub branch

Haston LaGrone:

* Has program that runs both HoughCircles and Blob detection concurrently.
* Parameter tweaking on HoughCircles has resulted in impressive detection accuracy for the current fixed camera. Also, able to interface to multiple cameras at once and display them as needed.
* Identified the general method in which OpenCV numbers the port inputs.
* Further investigated the effects of grayscale and blur on the target.
* Resolved method for targeting only one circular object on screen with HoughCircles. Program can still be confused by very similar sized object in close proximity.
* Set up GitHub branch

# Areas Needing Attention

## Problems Encountered

* Hard light reflections on the surface of the table are still a problem. Discussed solutions include:
  + Custom lighting rig
    - Possibly a diffuser
  + Running two detection programs simultaneously – one for in harsh light and one for outside harsh light
    - Only use data from one at a time, but switch if the primary detection method can’t find a circle
  + Leave it up to the user to have multiple cameras cover the same area from different angles.

## Resources Needed

The team spoke with Nardos on 11/8 about potentially using a diffuser over the table as a solution for the glare on the table and to have consistent lighting across the whole playing field.

# Work Pending

The team’s main goals are to continue their research and practice on OpenCV puck detection methods, and to begin organizing the problem into sections for efficient teamwork. First, OpenCV has several different tools that could be used to detect the puck. The team is exploring several of these options to find which one will be the most efficient and easy to implement. Second, the project is beginning to split into two priorities: transforming the image (or recognized shape) based on the camera’s angle and position, and filtering the image based on lighting conditions. The team is reorganizing roles based on these two needs.

Daniel Pullicar:

* Explore OpenCV’s calibration function for off-angle puck detection.
* Find best strategy for dealing with hard light reflections.
* Benchmark the runtime of known detection methods with a variety of filters etc.

Mason Cannon:

* What makes the program slower and faster?
* Start applying different filters to the frames to see how they will help.
* Help Haston out with figuring out which method of detecting the puck will be best.
* Use GitHub for coding

Roscoe Ambrose:

* Improve GUI
  + Make quit button on the GUI only close the GUI, not the entire program
  + Make more visually appealing to the user using styles
  + Add more setting controls from the circle detecting algorithm
* Investigate Mutex in Python
  + Using threading function to run two while loops simultaneously
  + Mutex will prevent a core dump error from a thread pulling a variable while another thread is writing the same variable
* Investigate elliptical transform (cv2.ellipse()) to compare with HoughCircles

Haston LaGrone:

* Shift focus to testing whether blob detection, HoughCircles, or Ellipse detection is the most effective method of tracking puck with table glare.
* Identify an approximate maximum velocity that the program can track the puck across a space.
* Assist Roscoe with parameters for GUI
* Assist Daniel with calibration methods.
* Begin pushing code to GitHub repository

# Conclusion:

The team has been able to explore HoughCircles to an extent where we are seeing positive results already. In the teams’ meeting with Nardos on 11/8, Haston was able to demonstrate his code in which he made tweaks to the settings in a way that tracked the puck well even at high speeds. Secondly, we are also becoming familiar with GitHub and plan to begin committing our code. Lastly, we are beginning to see and plan ways to divide work amongst the team to efficiently to explore the best options for the project.