Journal Report 18
02/16/20-02/23/20
Aimee Feng
Computer Systems Research Lab
Period 1, White

Daily Log

Monday, February 17

Implemented a shape identifier from OpenCV and labeled the circles in an image to try and bypass the problem of HarrisCorner not detecting sufficient points for spherical objects.

Tuesday, February 18

Paired circles and ovals together and tried to find ways to identify corresponding points on objects and shadows. Objects that have a roughly spherical shape however were hard to locate their shadows, since their shadows did not always appear close enough to an oval or circle to be identified by the shape classifier.

Thursday, February 20

Tried to speed up my current code by using different data structures.

Timeline

Date	Goal	Met
February 9	Associate at least 1 key point on the shadow with its respective point on the object	No, I'm still working towards finding better points.
February 16	Isolate the critical points on the object from all critical points located on the image.	Yes, the critical points closer to the shadow region in XY/LAB distance are the points located on the shadow or object.
February 23	Find critical points for round objects	No, I used the shape as a whole to try and find its corresponding shadow, and I could not figure out to handle objects that were round enough to not have any corners detected but cast a shadow not round enough to be considered a circle or oval.
March 1	Associate at least 1 key point on the shadow with its respective point on the object	
March 8	Decrease run time to within one minute.	

Reflection

A huge problem that I'm seeing is that my program has a terrible run time. Looking through my code, there are pieces that are

$$O(N^2)$$

depending on the size of the image I'm inputting. On smaller images, like the image of the white flower, the program runs really fast and is able to find the points within roughly 15 seconds. However, on larger images like the sign on the beach, the program takes much longer to run, roughly 90 seconds or so. Resizing the image would decrease run time, however doing so would also decrease the quality of the image, possibly making it harder for the program to find critical points.