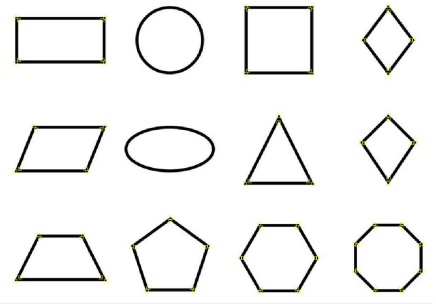
Daniel Bogusz
#301237992
CMPT 361 - Assignment 2

In this project I used algorithms found in “Computer Vision: Algorithms and Applications, 2nd Edition”, by Richard Szeliski, to write matlab code which detects harris corners, defines sift descriptors, matches features, detects objects, and matches objects.

The images used in my tests can be found in the \images folder.

Harris corner detection

After writing my initial code for Harris corner detection, I ensured it worked on simple shapes before attempting to tune my variables for the pictures I had taken.



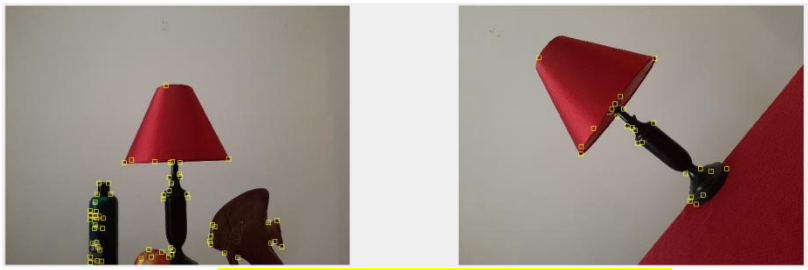
My results were largely successful, however, even with tuning the more complex images have many corners.



main("images\11.jpg", "images\11.jpg",1)
main("images\21.jpg", "images\11.jpg",1)
main("images\35.jpg", "images\11.jpg",1)

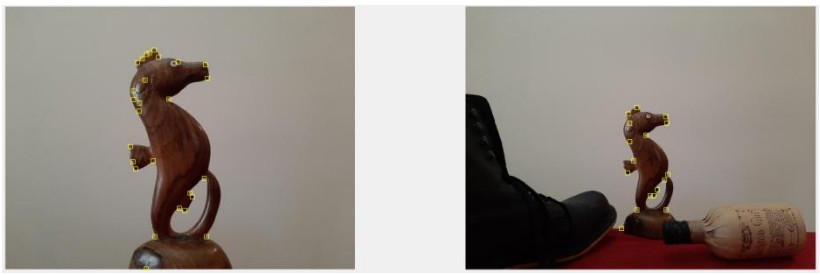
SIFT Feature Matching

On my first set of pictures, my algorithm worked very well, owing to the simplicity of the subject, a lamp. I calculated an average TPR of **0.9048**, an average PPV **0.6666**, and an F score of **0.7676**.



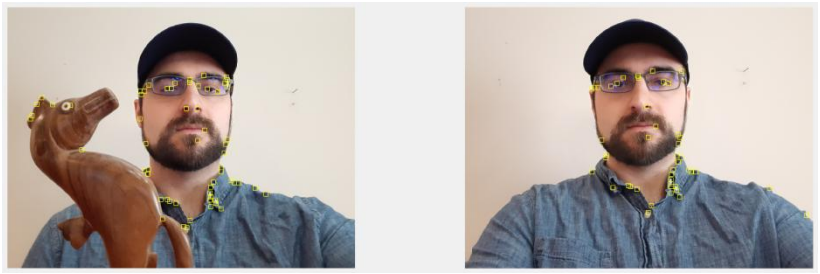
matlab command = main("images\12.jpg", "images\13.jpg",4)

My second set of pictures had overall lower scores, in part due to lighting changes caused by shadows created by nearby objects, but some images were very good. I calculated an average TPR of **0.8571**, an average PPV of **0.4557**, and an F score of **0.5950**.



matlab command = main("images\21.jpg", "images\22.jpg",4)

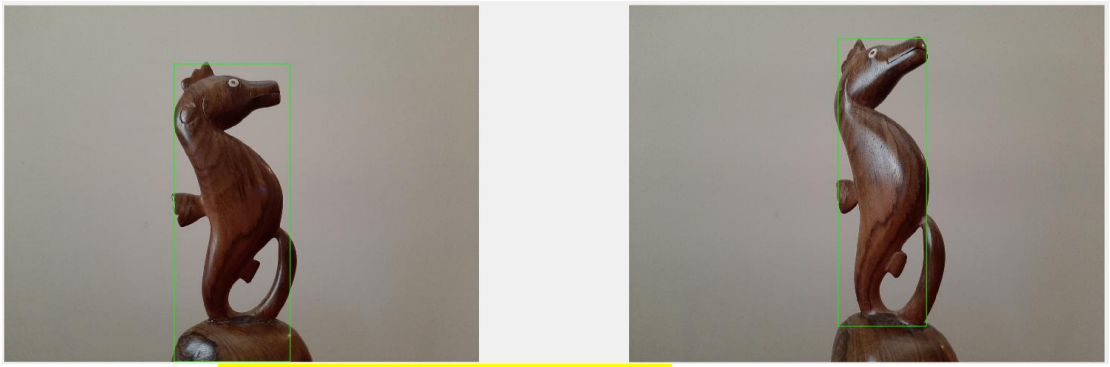
Some of the pictures in my third set created problems I did not anticipate when I took them, these photos were not tested. I calculate and average TPR of **0.8661**, an average PPV of **0.6250**, and an F score of **0.7261**.



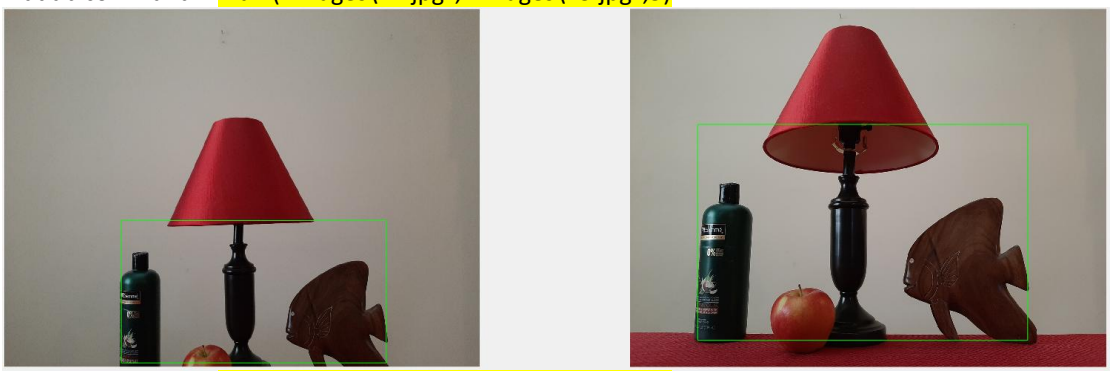
matlab command = main("images\32.jpg", "images\31.jpg",4)

Object detection

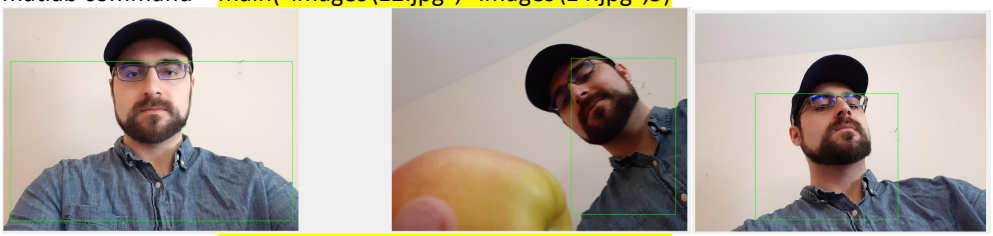
My algorithm is a direct extension and combination of the feature detection and corner matching algorithms. I use the first picture (the first argument) to make a second description pass, using groups of detected features in place of groups of corners. This becomes the template for what is searched for in the second picture.



matlab command = main("images\12.jpg", "images\13.jpg",5)

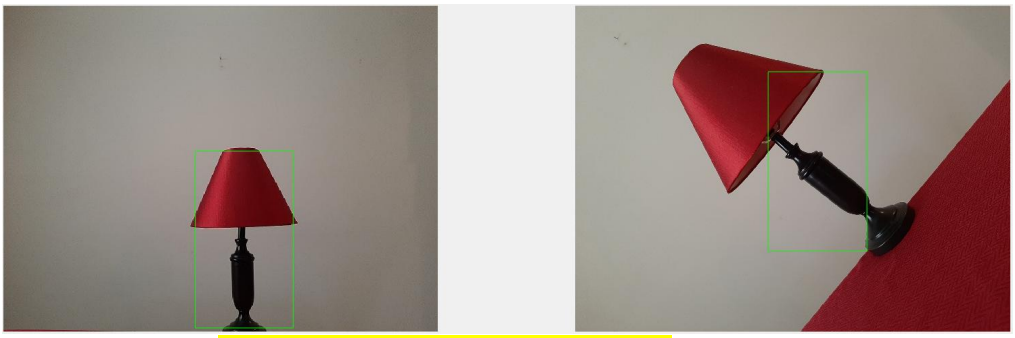


matlab command = main("images\12.jpg", "images\14.jpg",5)



matlab command = main("images\31.jpg", "images\34.jpg",5)
matlab command = main("images\31.jpg", "images\33.jpg",5)

This algorithm has mixed results because it can detect parts of the object, not necessarily the entire object.



matlab command = main("images\11.jpg", "images\13.jpg",5)

Using tunable variables I attempted to limit detections to only return one object.
I encountered a problem in which multiple parts of the same object were detected without the entire object being detected. I solved this problem by combining overlapping detected areas.

matlab command = main("images\11.jpg", "images\52.jpg",5)