Devon Peet

Dpeet3, dpeet@gatech.edu

902557439

HW5

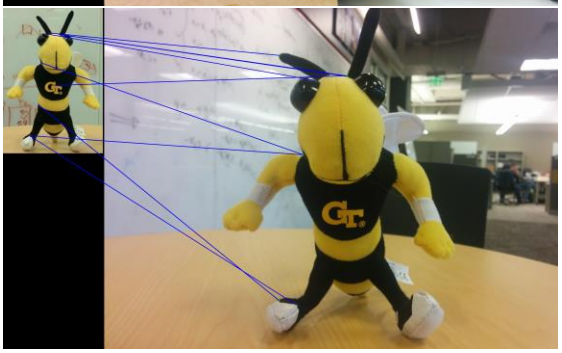
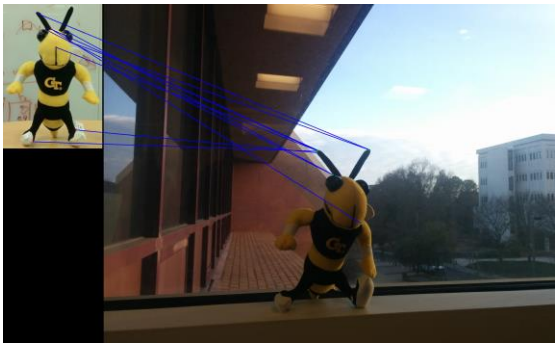
Source Images





Output Images



Implementation:

Using the documentation I used the sift detect and compute function which outputs a tuple of the keypoints and the descriptions. Then I created a BFMatcher using the cv2.Norm_Hammming constant and setting crossCheck to true. Then I matched the two images, sorted the resulting list, and returned the top 10 matches. I also had to import cv2.ORB() manually due to a memory error.

Sample: All points matched

This is the simplest task for SIFT because it is scale invariant. This is practically its entire purpose.

Lighting: 0 points matched

This foiled the SIFT Matcher quite handily, resulting in no points matching. This is mostly due to the reliance on the background color of the original image. The whiteboard color looks most like the sky, and the algorithm had lots of difficulty separating the two.

Scale: 3 points matched

While you would think scaling would be a good match for this algorithm, it again had issues with the background separation. Additionally buzz has many places where black and yellow alternate which the algorithm often made mistakes on.

Rotation: 6 points matched

The most successful of the partial matches, the key to success was similar to the sample which matched all points: look for the GT logo which has plenty of corners to latch onto, and in all our pictures is visible to the camera.