#### Roger Williams University

#### ENGR 430 Computer Vision

#### Homework 4 Feature Mapping

#### 25 Points

Please name each file according to the homework and problem number, for example H4P1.py, H4P2.py, H4P1.JPG, etc. Bridges automatically places the submissions of each student in a separate folder, so please do not submit as a compressed folder. Any files your scripts need to run, for example the original image files, must also be uploaded to Bridges. Put a header like the example shown on each script file submitted:

############################

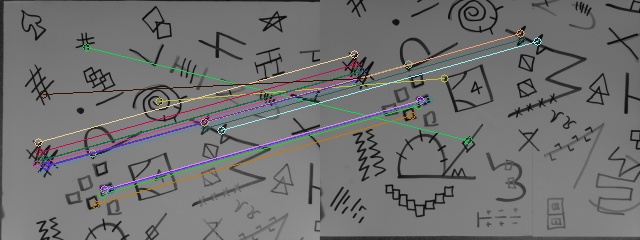
# Ash Ketchum

# ENGR430 Computer Vision H2P1.py

# 9/15/2017

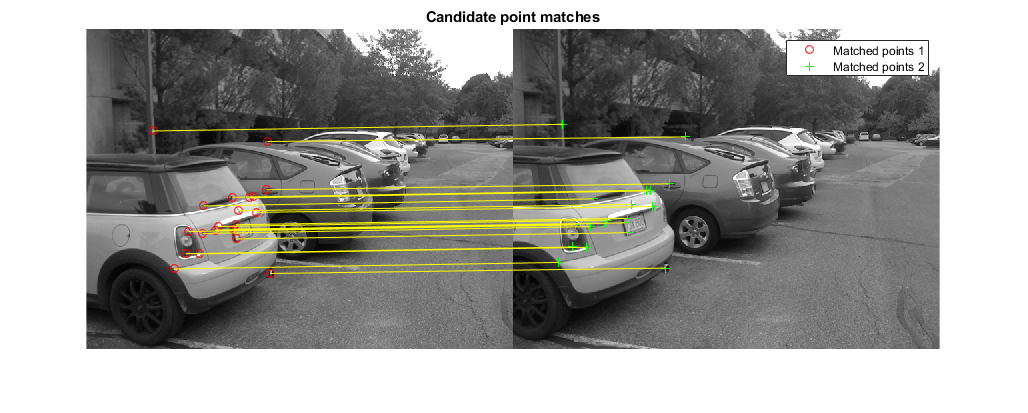
#############################

4.1 **Map Matching**. Take two images of your map that contain some evidently duplicate features displaced, these may be two images you used to stitch together your map, or you may take new pictures. Using the built-in ORB detector, knnMatcher and drawMatches developed in class, display a window with the matches drawn between the two images, as shown below. Images must be sufficiently feature rich for **estimateRigidTransform** return a transform. (Add features to your map and take new images until this is possible, if there are insufficient matches **estimateRigidTransform** will return **None**). Print out the returned transform to the command window.



Submit python script that accepts two filenames as arguments and the two image files named Map01.jpg and Map02.jpg. The script must produce acceptable results even when the instructor swaps file name order. Reverse order should also display, more-or-less, the negative of the original transform. You may also submit JPEG images of the results for consideration for partial credit if the script does not execute.

4.2 **Campus Matching**. Take two images of some prominent campus feature or scene. Pictures must be your own and must clearly be taken on the RWU campus. Display a window with 20 matches drawn between the two images, with the **majority** of the matches apparently correct. Images must be sufficiently feature rich for **estimateRigidTransform** return a transform. Script must accept two image file names and these must be named Campus01.jpg and Campus02.jpg. Reversing file names must also produce acceptable results and more-or-less the negative of the original transform.



Submit python script and the image files. You may also submit JPEG images of the results for consideration for partial credit if the script does not execute.

4.3 **Terribly wrong.** To explore the limitations of this detector/matcher, take two images of some prominent campus feature or scene that confuse it. The pictures must be such that the window displaying the **best 20 matches** clearly shows the **majority** of the matches are evidently wrong. However, it must be discernable from the human eye that a correct transform between the images exists. For example, pictures of two completely different buildings will confuse the detector/matcher but do not meet the requirements of this assignment because a true transform between the images does not exist. Do not compute or display the meaningless average transform. Script must accept two image file names and these must be named Campus03.jpg and Campus04.jpg. Reversing file names must also produce acceptable results.