Problem Set 5: Harris, SIFT, RANSAC

Jake Ritter
902036712
gth675r

Questions

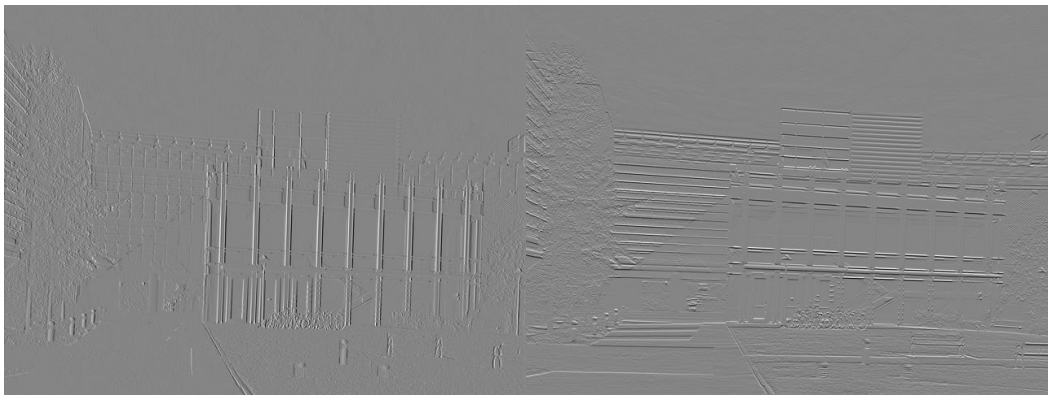
1. Harris corners

In class and in the text, we have developed the *Harris* operator. To find the Harris points, you need to compute the gradients in both the X and Y directions. These will probably have to be lightly filtered using a Gaussian to be well behaved. You can do this either the "naive" way - filter the image and then do simple difference between left and right (X gradient) or up and down (Y gradient) - or you can take an analytic derivative of a Gaussian in X or Y and use that filter. The scale of the filtering is up to you. You may play with the size of the Gaussian as it will interact with the window size of the corner detection.

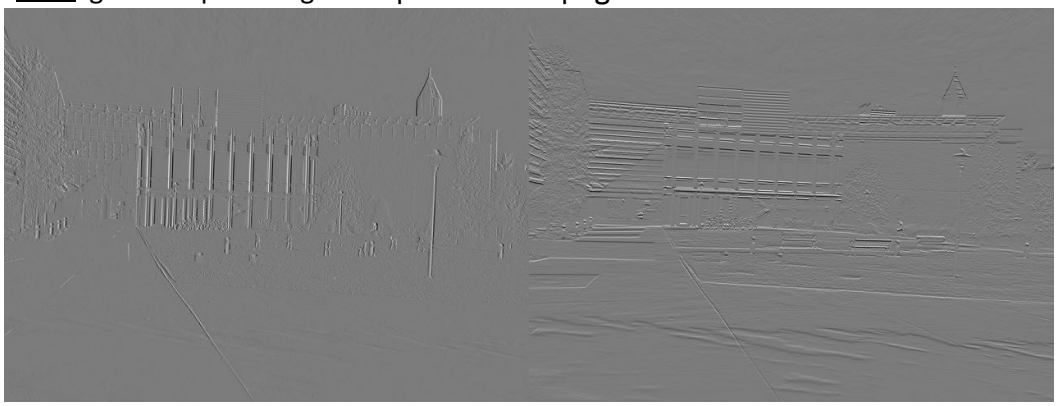
a.

Output: The gradient-pair image for both transA and simA:

- transA gradient-pair image as `ps5-1-a-1.png`



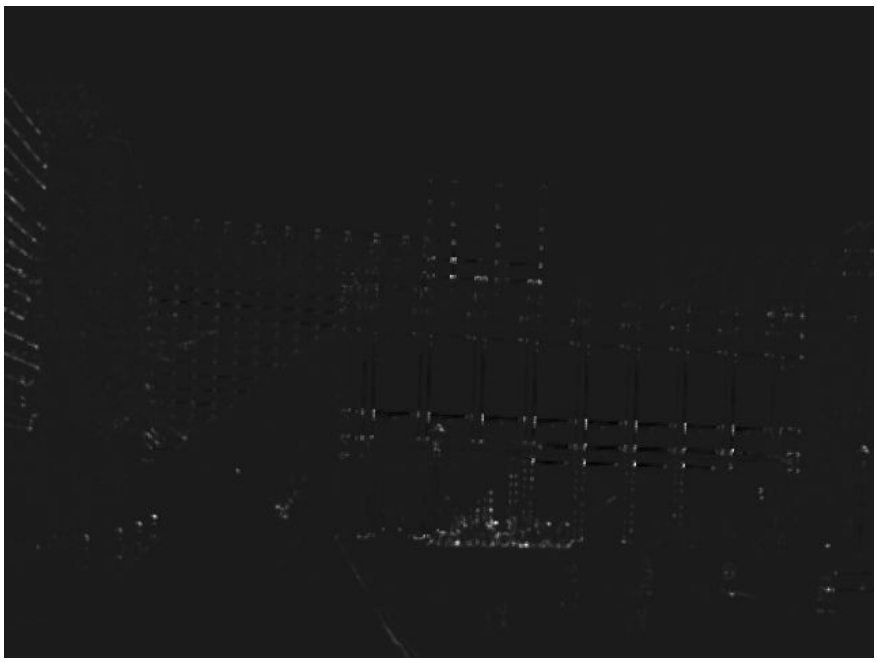
- simA gradient-pair image as `ps5-1-a-2.png`



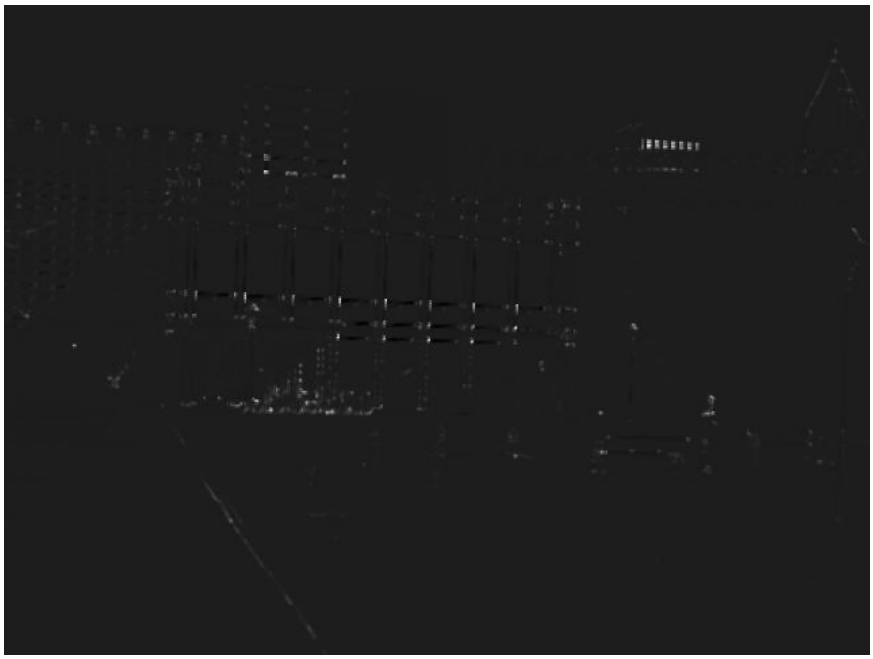
b. Now, you can compute the Harris response for the image, defined as:

Function: `harris_response(Ix, Iy, kernel, alpha) -> R`

Output: The Harris response image for:
- transA image as `ps5-1-b-1.png`

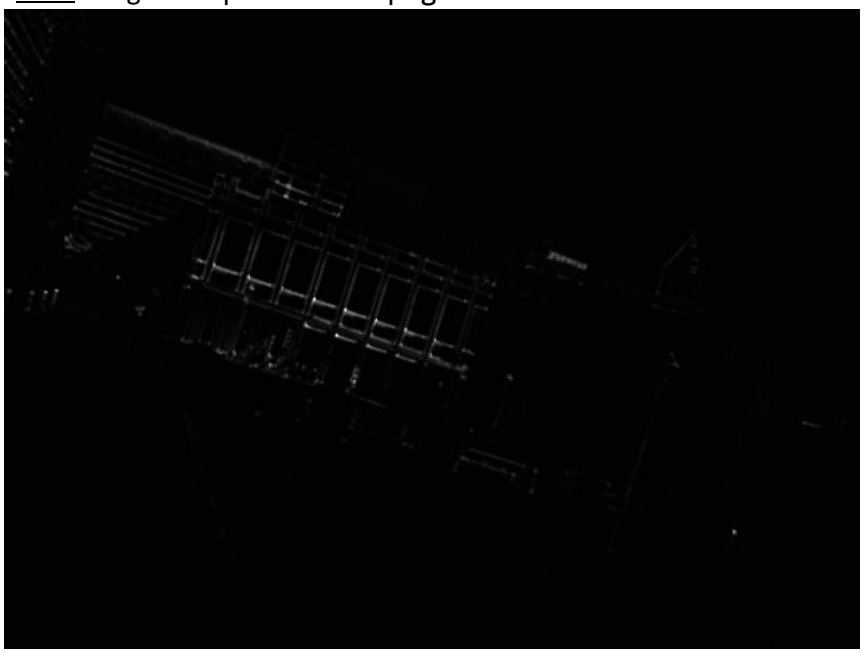


- transB image as `ps5-1-b-2.png`



- simA image as `ps5-1-b-3.png`

- simB image as `ps5-1-b-4.png`



c.

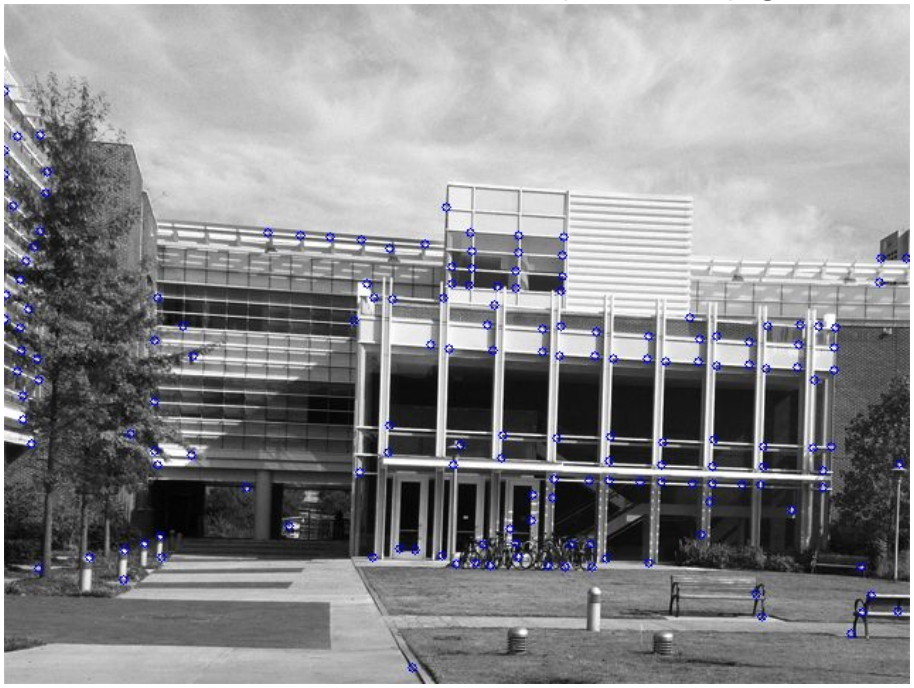
Functions:

- `find_corners(R, threshold, radius) -> corners`
- `draw_corners(image, corners) -> image_out`

d.

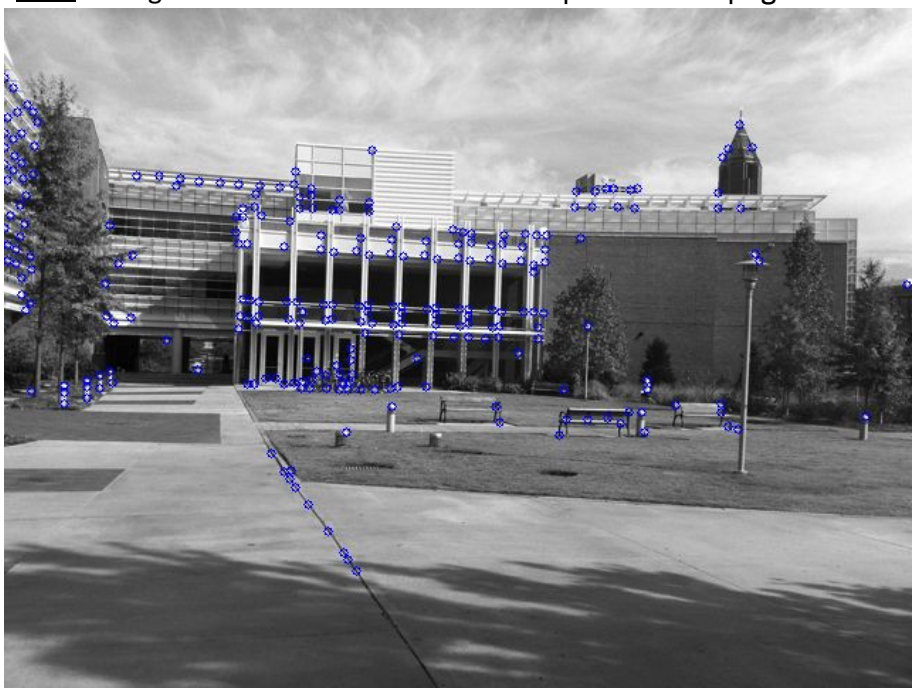
Output: Apply your function to both image pairs: (transA, transB) and (simA, simB). Draw the corners visibly (in color) on each of the four result images and provide those images.

- transA image with Harris corners marked as `ps5-1-c-1.png`

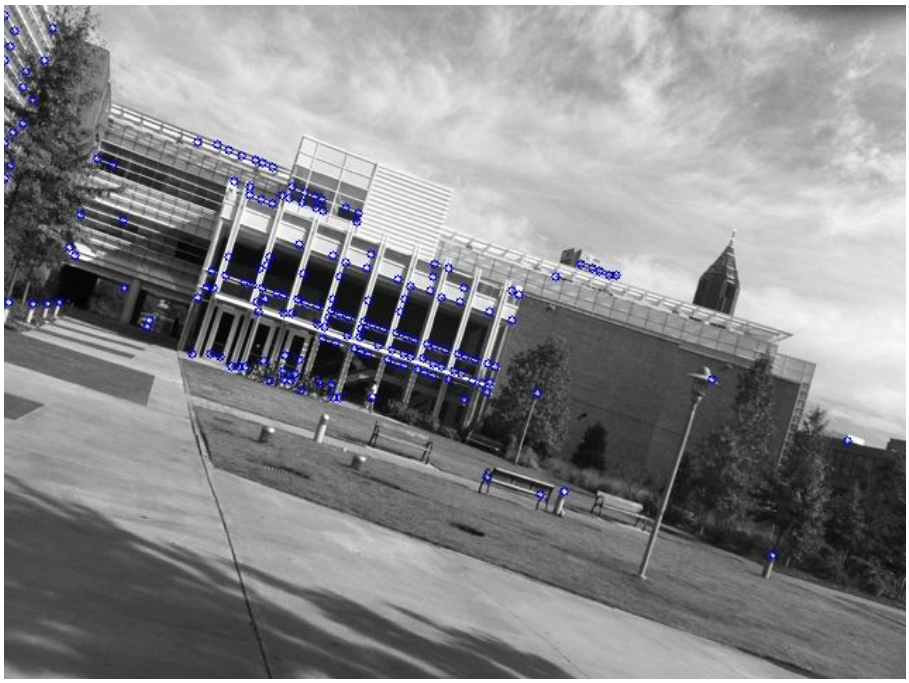


- transB image with Harris corners marked as `ps5-1-c-2.png`

- simA image with Harris corners marked as `ps5-1-c-3.png`



- simB image with Harris corners marked as `ps5-1-c-4.png`



Output (Textual Response): Describe the behavior of your corner detector including anything surprising, such as points not found in both images of a pair.

These were the corner results after a lot of playing with the different parameters. Ended up reducing the thresholds greatly after running the RANSAC part to produce more corners. The most

surprising thing I found was that the corners on simB (the rotated image) seem to be extremely heavily clustered on the middle of the image. There was a lot more need for using a maximum radius there. Additionally, I thought it interesting that the line on the sidewalk was picked up in sim A, but *not* in sim B - looking at the Harris response, I can see why, but it's still odd that it was picked up at all. Finally, sim B doesn't have anything on the top of the tower, which to a human eye is a very noticeable 'point'.

2. SIFT features

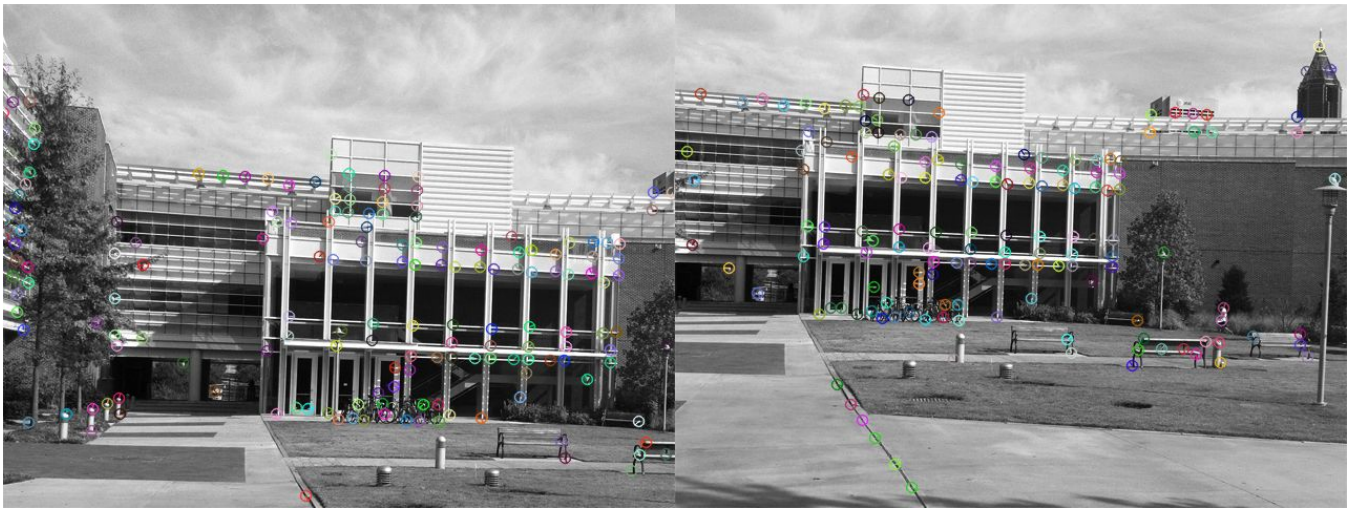
a.

Functions:

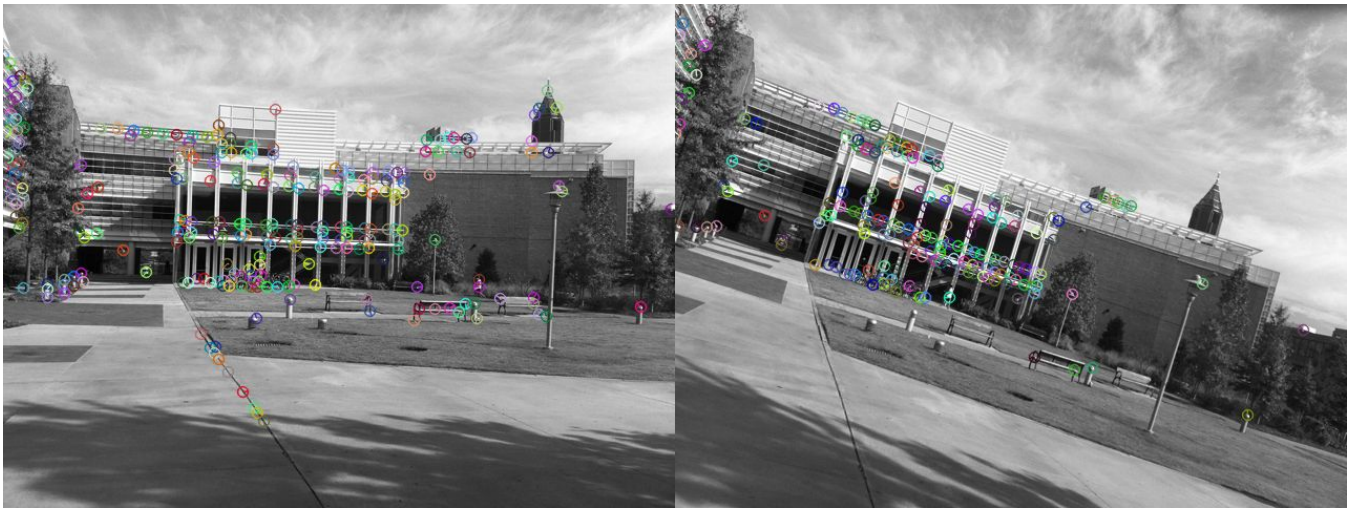
- `gradient_angle(Ix, Iy) -> angle`
- `get_keypoints(points, R, angle, _size, _octave) -> keypoints`

Output:

- Interest points with angles shown on (transA, transB) pair as `ps5-2-a-1.png`



- Interest points with angles shown on (simA, simB) pair as `ps5-2-a-2.png`



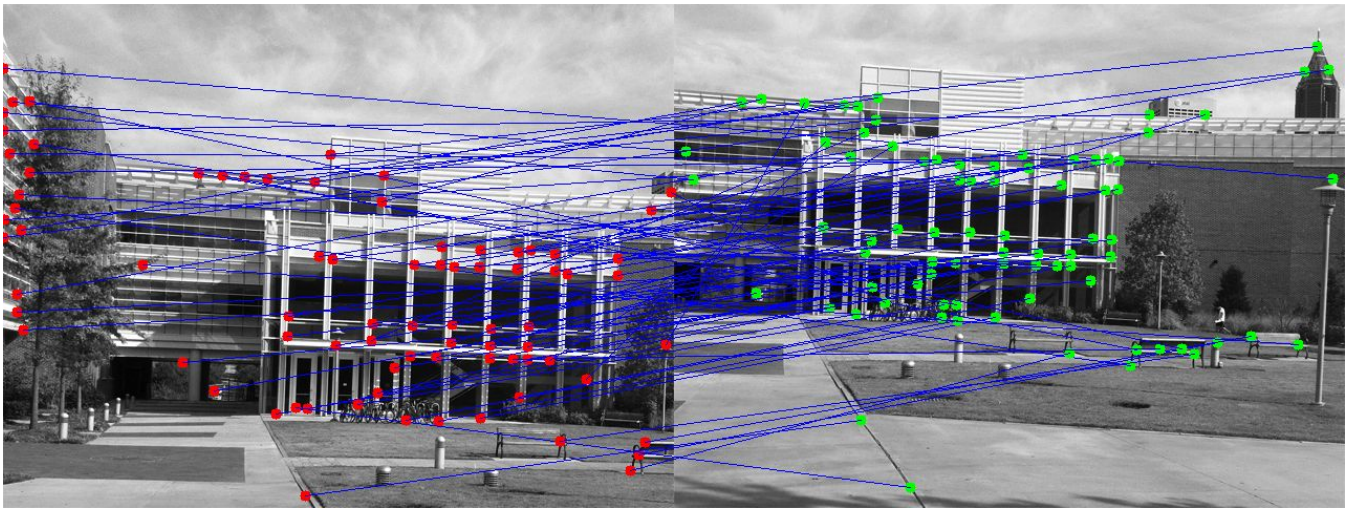
b.

Functions:

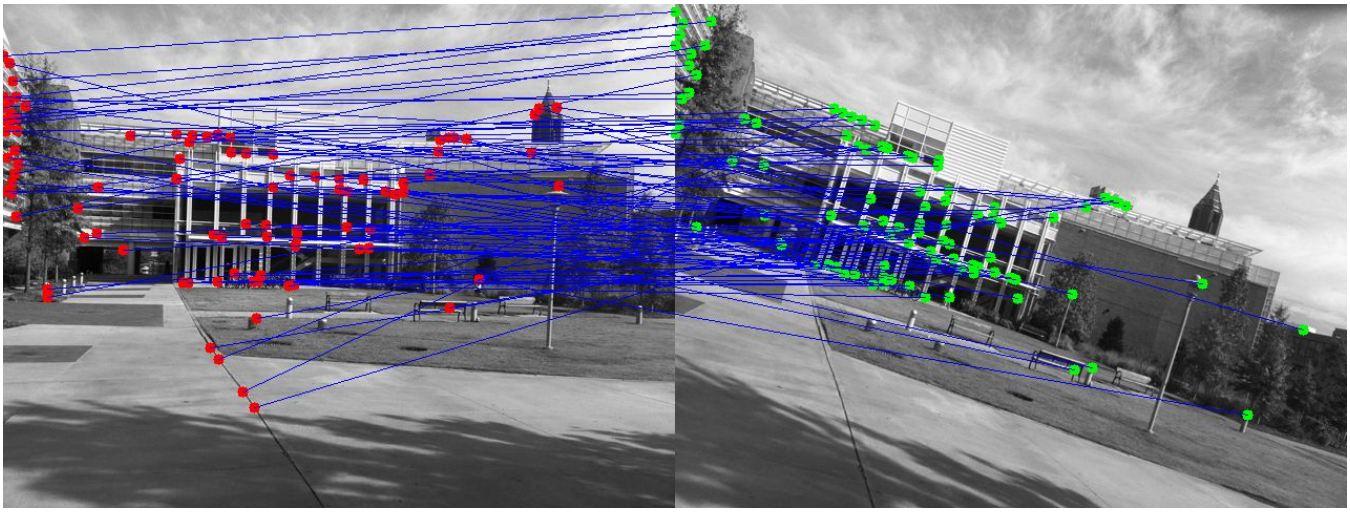
- `get_descriptors(image, keypoints) -> descriptors`
- `match_descriptors(desc1, desc2) -> matches`
- `draw_matches(image1, image2, kp1, kp2, matches) -> image_out`

Output:

- putative-pair-image for (transA–transB) as `ps5-2-b-1.png`



- putative-pair-image for (simA–simB) as `ps5-2-b-2.png`

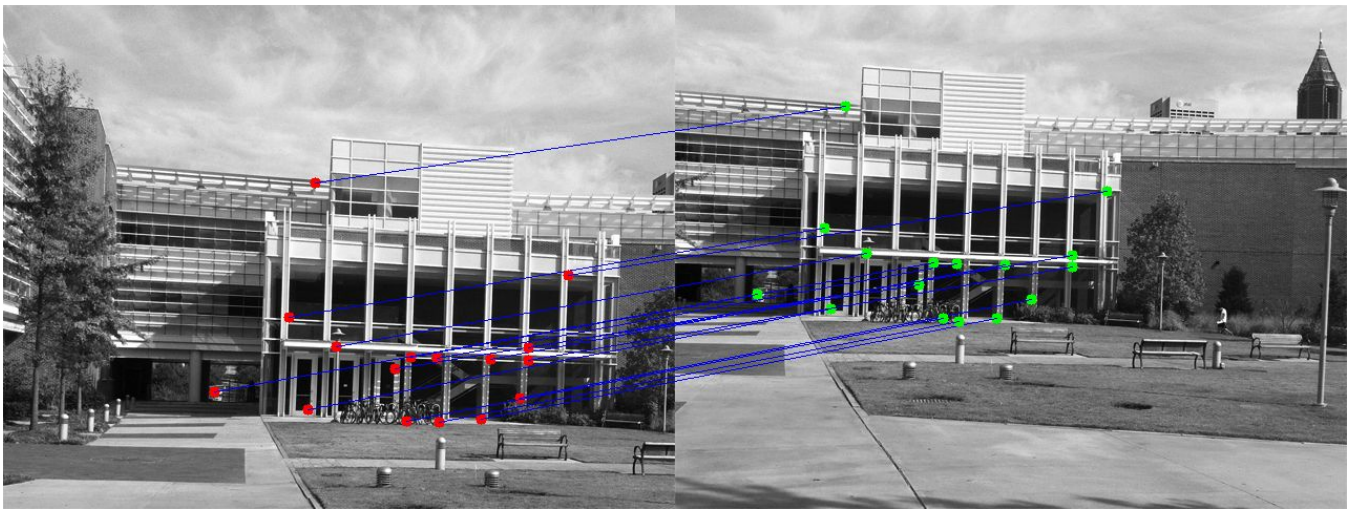


3. RANSAC

a.

Function: `compute_translation_RANSAC(kp1, kp2, matches) -> translation, good_matches`

Output: Biggest consensus set lines drawn on pair (transA, transB) as `ps5-3-a-1.png`



Output (Textual Response):

- What translation vector was used?

(-135.0, -89.0)

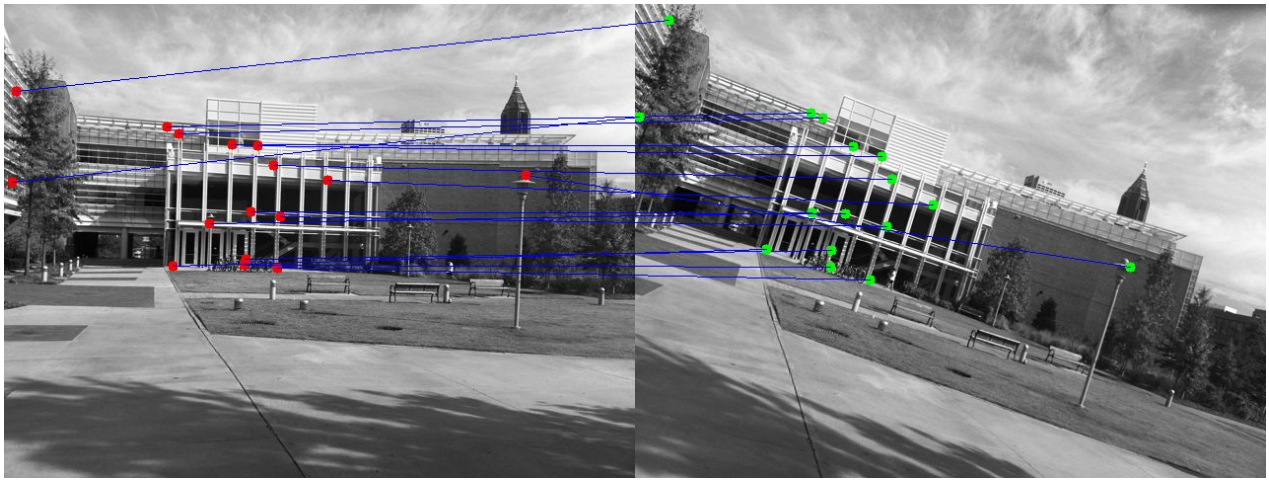
- What percentage of your matches was the biggest consensus set?

17 matches out of 80, so 21.2%

b.

Function: `compute_similarity_RANSAC(kp1, kp2, matches) -> transform, good_matches`

Output: Biggest consensus set lines drawn on pair (simA, simB) as `ps5-3-b-1.png`



Output (Textual Response):

- What is the transform matrix for the best set?

[[0.94355048 -0.33337659 56.75110304]
[0.33337659 0.94355048 -67.81053724]]

- What percentage of your matches was the biggest consensus set?

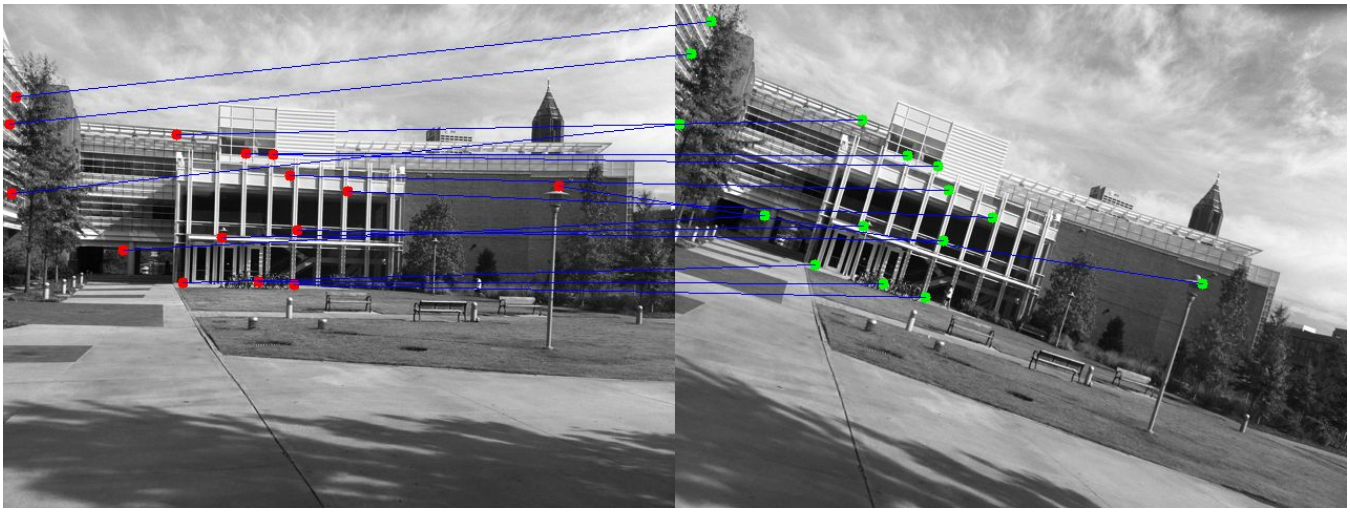
16 matches out of 91 or 17.6%

EXTRA CREDIT QUESTIONS (3-c, 3-d, 3-e)

c.

Output:

- Biggest consensus set lines drawn on pair (simA–simB) as `ps5-3-c-1.png`



Output (Textual Response):

- What is the transform matrix for the best set?

[[0.95817403 -0.32925524 52.08709283]
[0.30964215 0.97316103 -66.15755467]]

- What percentage of your matches was the biggest consensus set?

15 points out of 91 or 16.4%

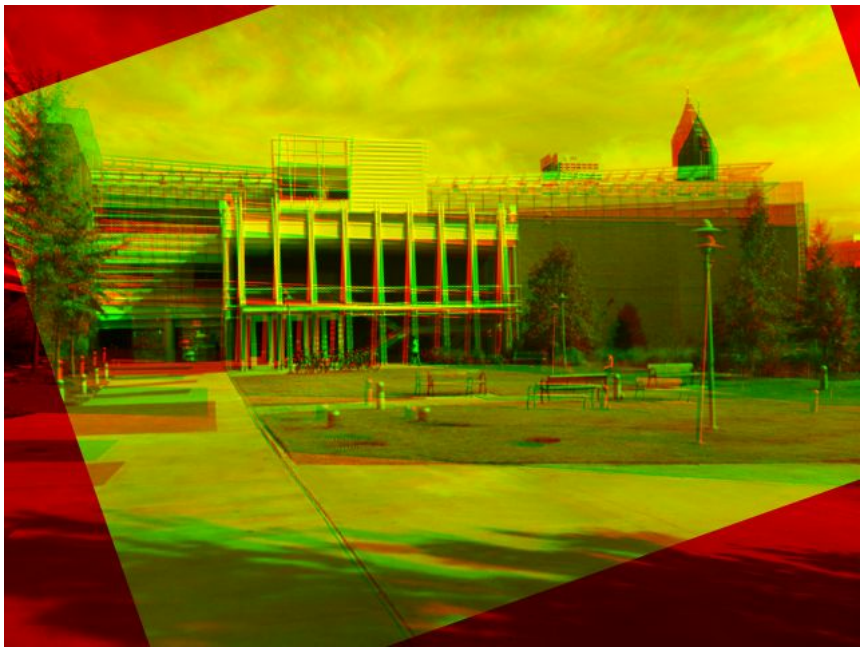
Note that 15 was my own limit. Given that this takes 3 points, running all possible transforms would take a long time. After playing with params, I found that 15 with a low threshold worked in a reasonable amount of time.

d.

Output:

- warpedB image as `ps5-3-d-1.png`

- the overlay image as `ps5-3-d-2.png`

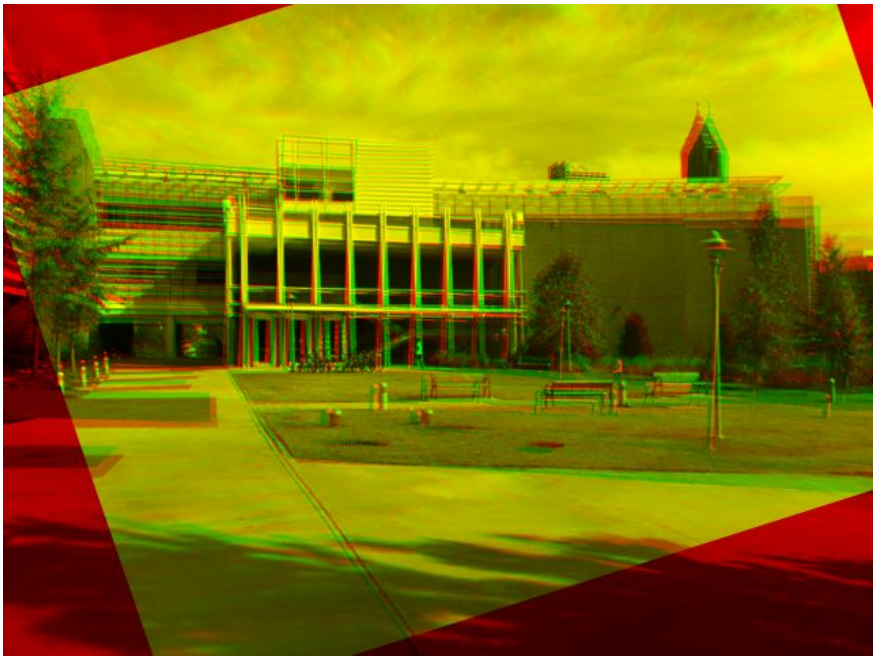


e. Do 3-d again but this time using the affine transform recovered in 3-c.

Output:

- warpedB image as `ps5-3-e-1.png`

- the overlay image as `ps5-3-e-2.png`



Output (Textual Response):

Comment as to whether using the similarity transform or the affine one gave better results, and why or why not.

It looks to me that affine transform definitely worked better. I think the extra degrees of freedom that come with the affine transform allows us to find a better fit overall. Also, I think if I gave the RANSAC algorithm a larger number of points to find (and waited a longer time) it would be possible to get even closer than it currently does. Overall, when you sample more points and have

more degrees of freedom, we should get a closer transformation.