Computer Vision

Problem Set 2,3,4

Krunal Chande

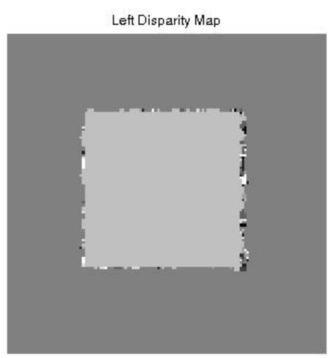
GTID - 902898449

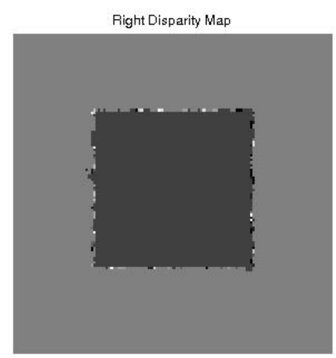
Date- 10/09/2014

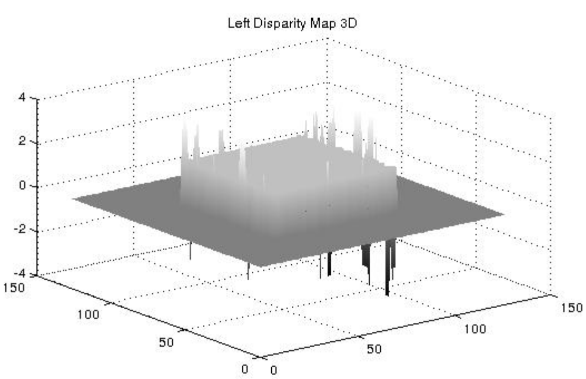
Problem Set 2:

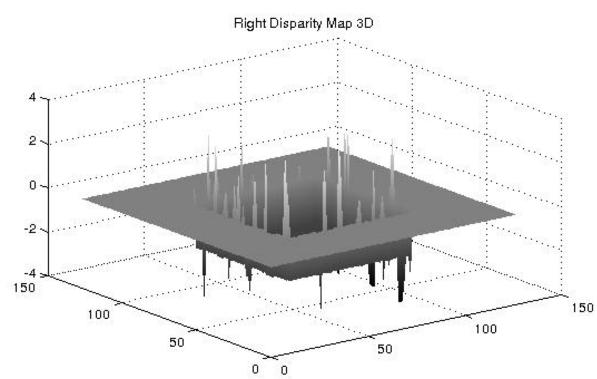
Q1.

A.







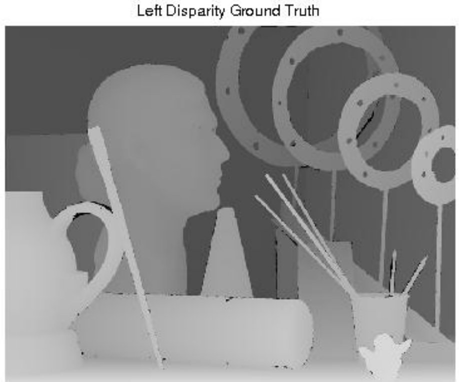


Q2.

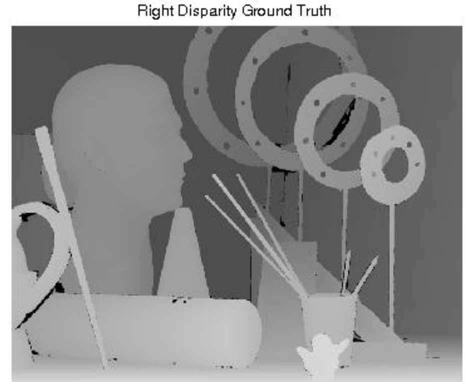
Left Disparity Map

Right Disparity Map

Left Disparity Ground Truth



Right Disparity Ground Truth



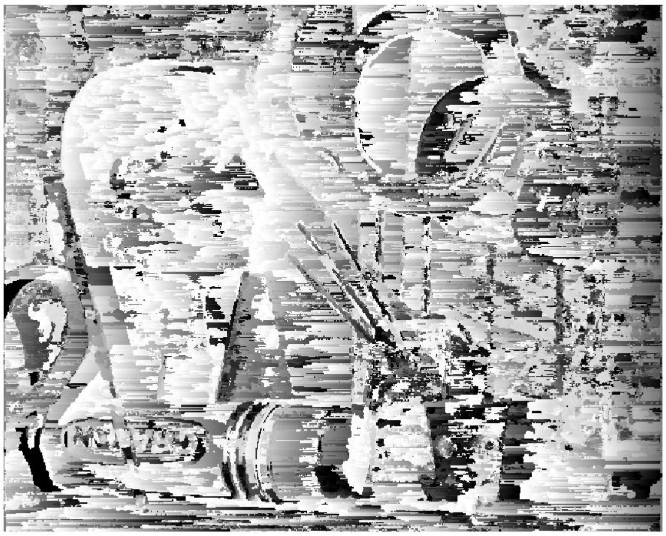
Description of Differences:

We see that the output of our method is nowhere close to ground truth, Varying the window sizes might help our case but not by much.

The interesting thing about this image is that many objects are overlapping one and another and this makes it difficult for the system to generate the disparity map out of it, (because of these high frequency depth variations).

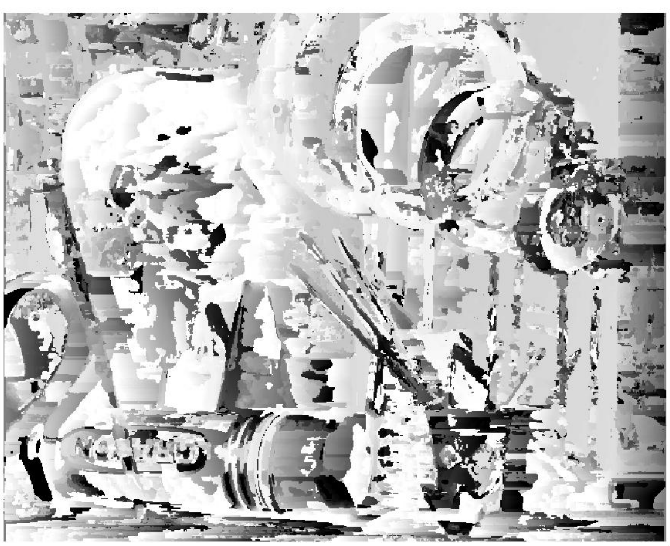
Q3.

A. Gaussian Noise



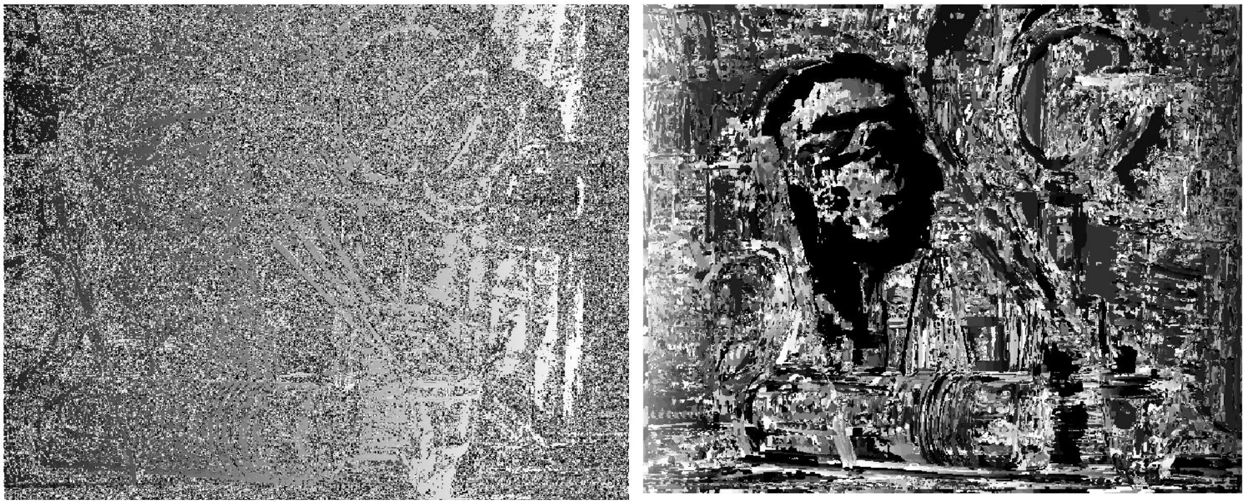

We only add noise in the left image. Adding noise throws the system off completely. The quality of our results dramatically drops down. With lots of high frequency noise in the image.

B. Increased Contrast




Even after increasing the contrast by 20% in the left image, the results are nowhere as bad as they are after adding noise. This also sort of makes sense because you will still get the best match in the same place.

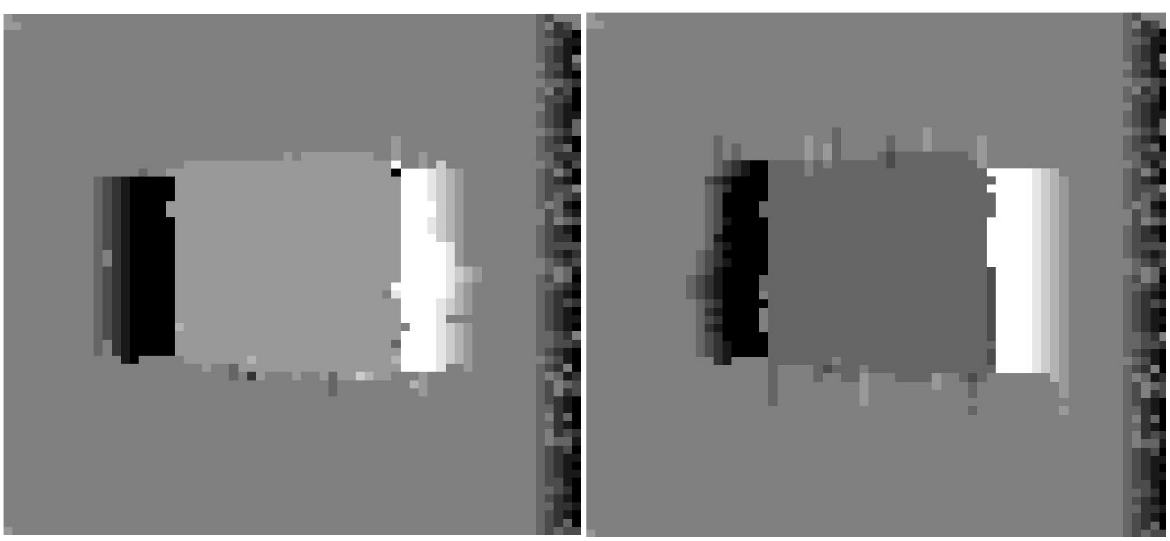
Q4. Using NCC
A.



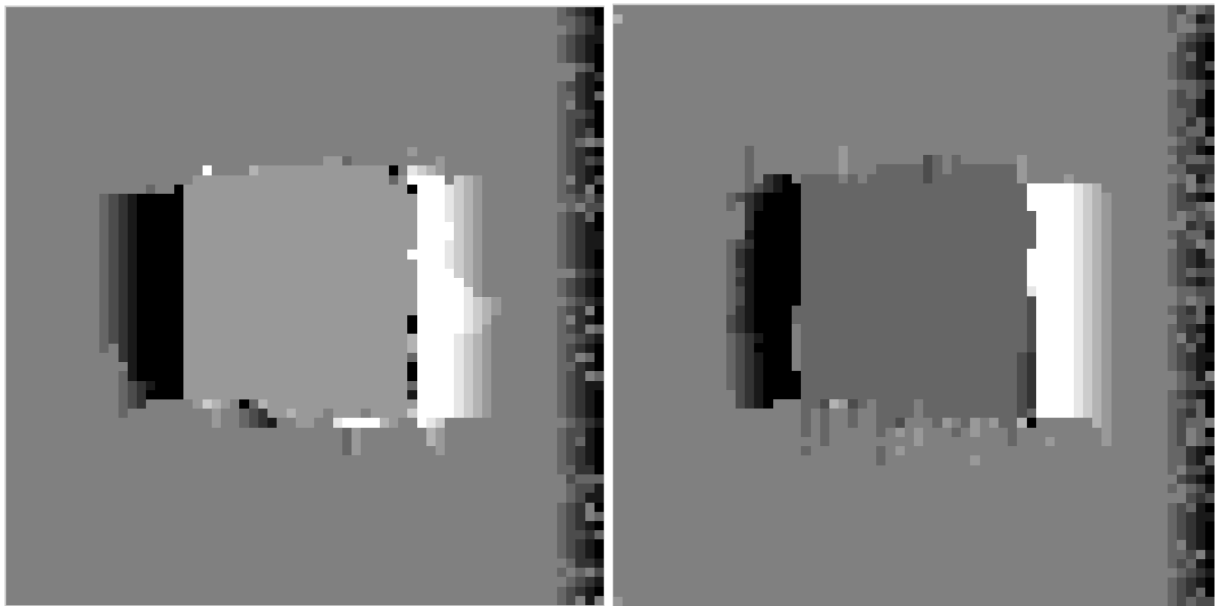
Normalized cross correlation was taking too much time and hence couldn't gather all results. Plus it seems that there was something wrong with the result, most probably the the parameters were incorrect (Window Size = 11, Max Disparity = 64), but did not have enough time to debug. But I am pretty pretty sure that the code is correct.

In theory Normalized cross correlation should work better than SSD because it is less susceptible to variations between two images, at least in terms of brightness and contrast. It also helps with the gaussian noise.

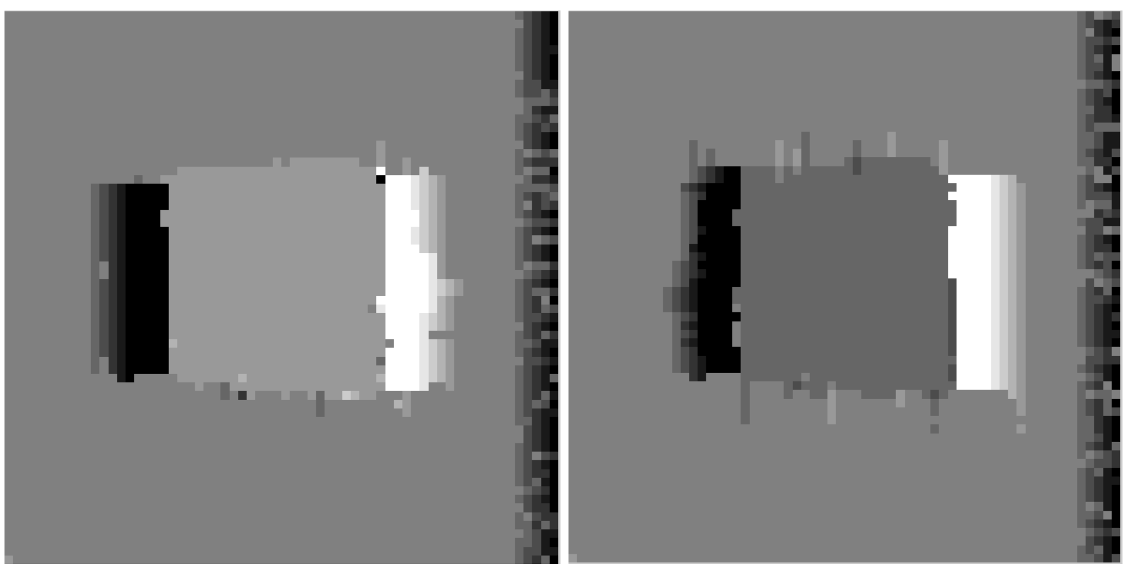
To demonstrate this, I will attach the results with the given test image.



Disparity Map with Normalized Crosscorrelation



Disparity map with Gaussian noise added to left image



Disparity map with contrast enhanced left image

We see that this method is much more robust to such inter-image variations.

Problem Set 3.

Q1.

A.

M_normA =

-0.458275543166123	0.294742369574437	0.013957455938145	-0.004025801919390
0.050855891008526	0.054584701993082	0.541059932899223	0.052375922470371
-0.109009583404807	-0.178345481048107	0.044267821489014	-0.596820496436617

<u,v> projection of last point =

Column 1

0.141906076749088

Column 2

-0.451843009596097

residualError =

0.001562136046217

B. We achieve different accuracies for different K. For the highest value of K i.e. 16, we see that the residual errors are the smallest. But it almost always performs worse on the test set. This is a common problem of overfitting data in Machine Learning and Pattern Recognition. We see that even more pronounced when we found M with all 20 samples in M with a residual error of the order of 10e-3.

M with lowest residual error (FOR K = 8 in this particular set of samples)

0.006931546878173	-0.004016844719508	-0.001326029273369	-0.826700553816969
0.001547687325295	0.001024527594947	-0.007274407158045	-0.562523255916803
0.000007609460522	0.000003709539886	-0.000001902032432	-0.003388077121813

C. Camera Center:

C =

1.0e+02 *

3.031000392419489
3.071842801573253
0.304216687438100

Q2.

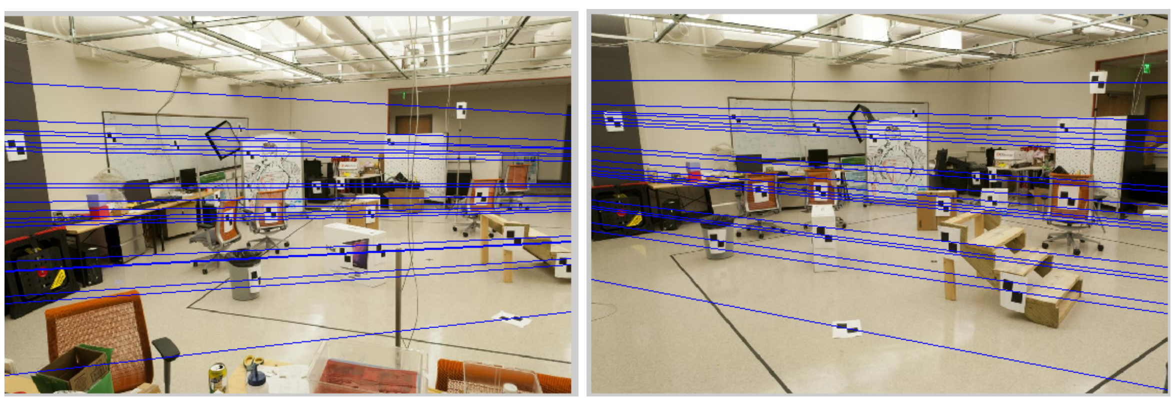
A. The matrix F

```
F =
   0.048677833709052   0.003526488513726  -0.000005067492291
  -0.996197527068972  -0.072169997490202   0.000103382517257
   0.000013944972837   0.000000000000005  -0.000000000000032
```

B. Matrix F

```
F_reducedRank =
   0.048677833712415  -0.003526492108205  -0.000000001130547
  -0.996197527070835   0.072170071511449   0.000000023137437
   0.000013799360751  -0.000002009952291  -0.000000001444042
Rank of F_reducedRank =
     2
```

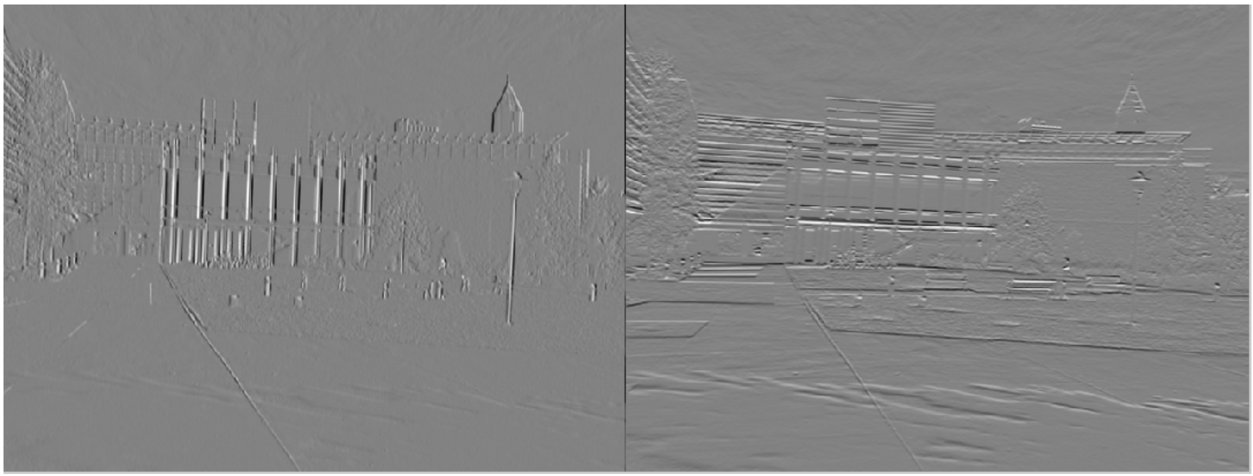
C. Images with Epipolar lines



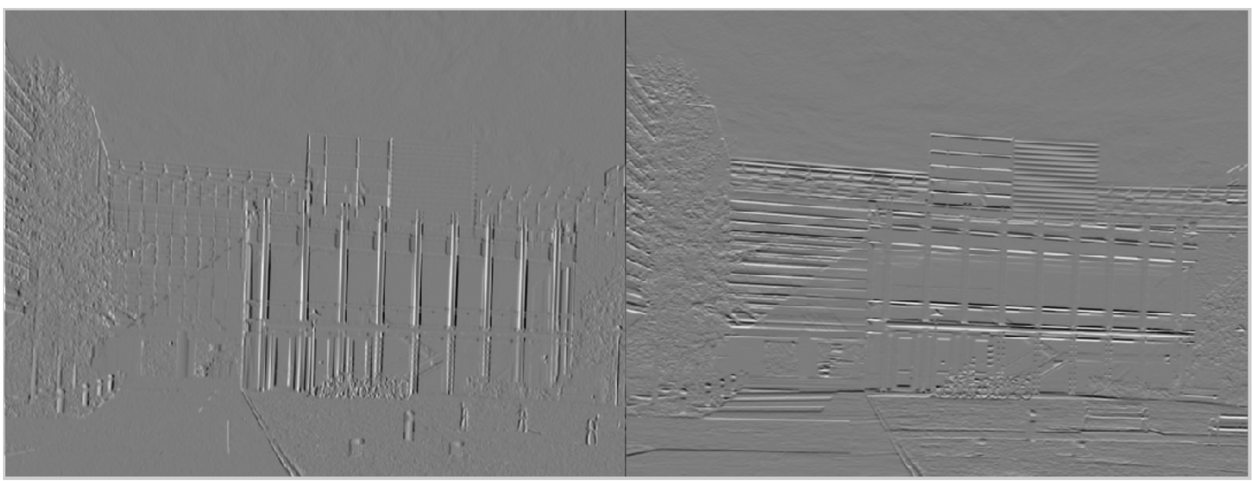
Problem Set 4.

Q1.

A. Gradient Pair Image for TransA and simA



Gradients for SimA and SimB



Gradients for TransA and TransB

B. Harris Value output for each image

Image	Harris Value
transA	1.017232264363703e+10
transB	8.847163520419640e+09
simA	6.765267567209656e+09
simB	9.966860448314156e+09

C. Images with Harris and Describe if something interesting

Harris Corners for images TransA and TransB

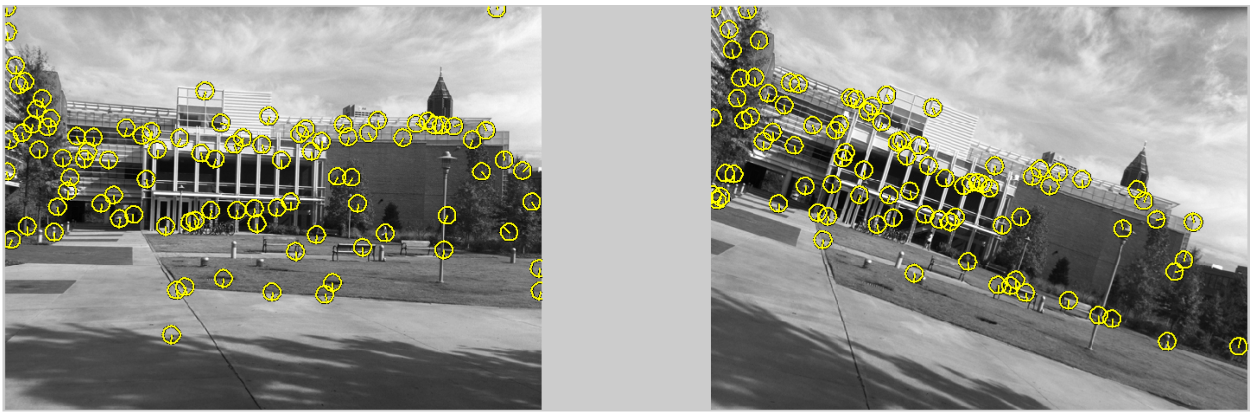
Harris Corners for images SimA and SimB

We see in these images that the interest points or corners found by the detector are nearly similar. Which is an extremely good thing, as it would help in the matching stage a lot. Another thing we see is that it finds interest points where you wouldn't expect any, like clouds. It seems that because there's strong contrast present in those areas it shows up as a corner point. Also some points like the tip of trees is neglected while searching for corners, due to lack of contrast. It seems that doing this over color images would be a good idea.

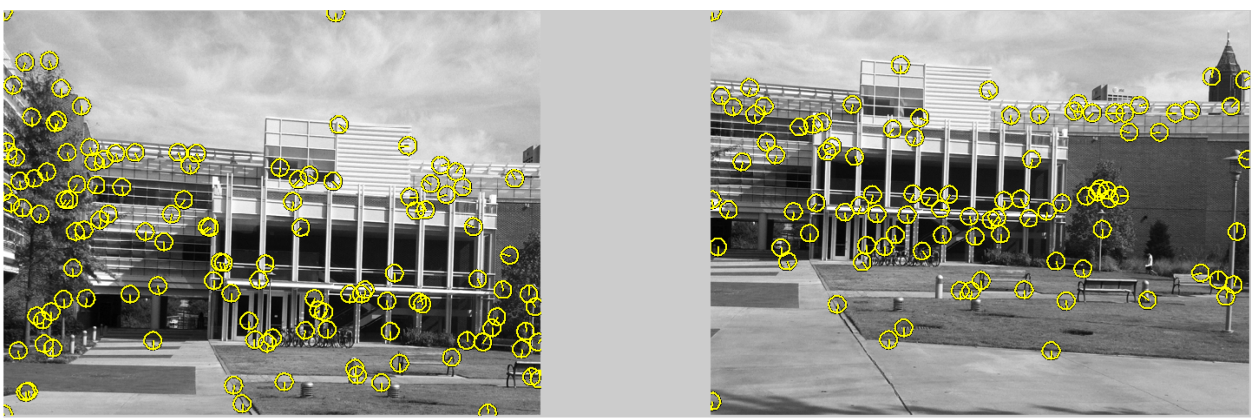
Q2.

A. Both Drawn on pairs

Gradient Direction (with a scale of 10 for visibility) for some subset of detected corners:

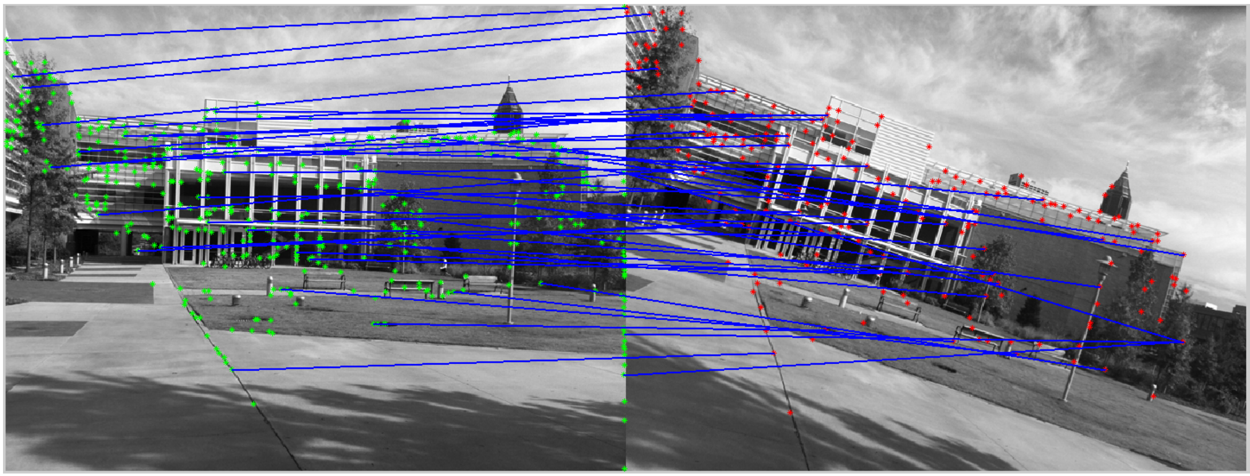


SimA and SimB pair of images

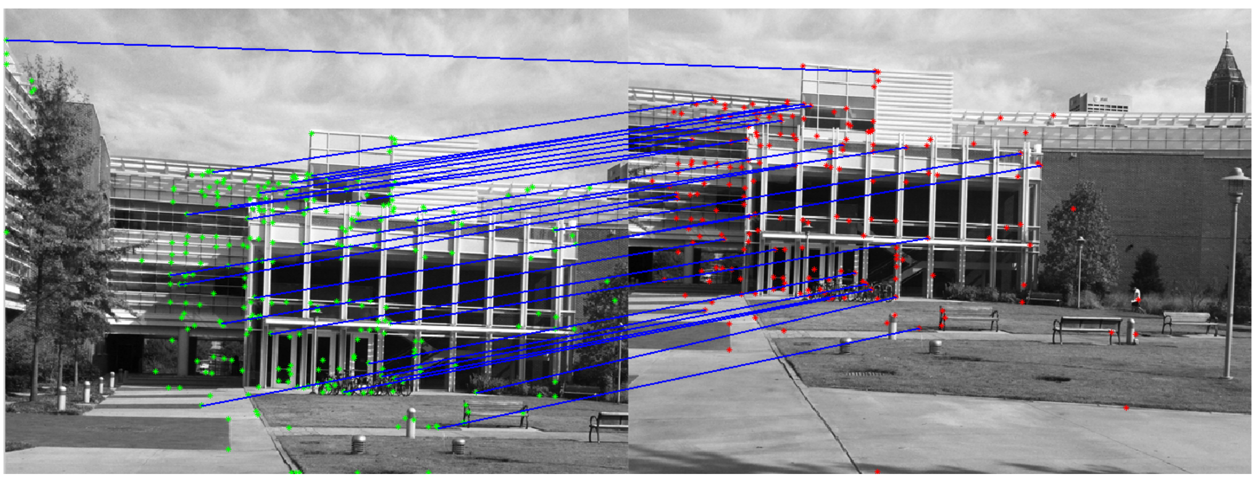


TransA and TransB pair of images

B. Both Putative Pair of images



SimA and SimB putative pair of images



TransA and TransB putative pair of images

Q3. RANSAC

A. Translation only:

H_k_best =

1.0e+02 *

```
  0.009010085337471   0.000169123351435  -1.104027928626840
 -0.000169123351435   0.009010085337471  -0.521377812257563
```

Inliers =

244

Percentage of Inliers: 0.8905

B. With Similarity transform

(Not sure what image to display here)

H_k_best =

```
   0.982266009852216  -0.301477832512315  38.797044334975574
   0.301477832512315   0.982266009852216 -66.978325123152587
```

Inliers =

   375

Percentage of Inliers: 0.8370