Assignment 2 Write up

Name: Kai Li

Andrew ID: kail2

1.1

Please refer to createGaussianPyramid.m for detailed implementation.

1.2

Please refer to createDoGPyramid.m for detailed implementation.

1.3

Please refer to computePrincipalCurvature.m for detailed implementation.

1.4

Please refer to getLocalExtrema.m for detailed implementation.

1.5

Please refer to DoGdetector.m for detailed implementation.

2.1

Please refer to makeTestPattern.m for detailed implementation.

‘testPattern.mat’ file is included in my submission.

2.2

Please refer to computeBrief.m for detailed implementation.

2.3

Please refer to briefLite.m for detailed implementation.

2.4



model\_chickenbroth.jpg and chickenbroth\_01.jpg



incline\_L.png and incline\_R.png



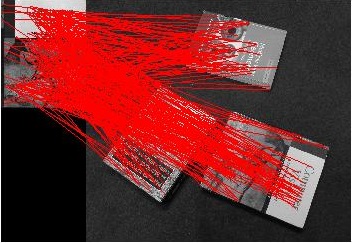
pf\_scan\_scaled.jpg and pf\_desk.jpg



pf\_scan\_scaled.jpg and pf\_floor.jpg



pf\_scan\_scaled.jpg and pf\_pile.jpg



pf\_scan\_scaled.jpg and pf\_floor\_rot.jpg



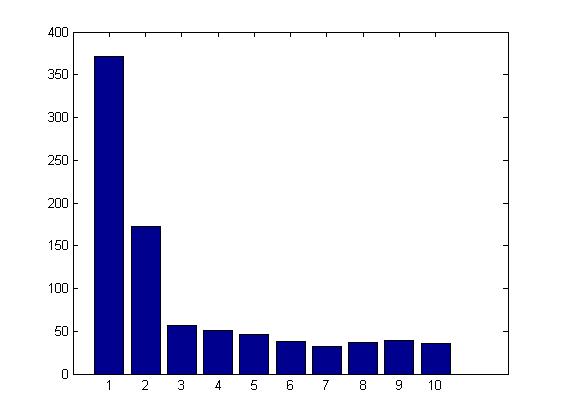
pf\_scan\_scaled.jpg and pf\_stand.jpg

When there is only translation involved, the mapping works pretty well, especially with similar scale (for example the first and second cases shown above); and if we rotate the image, with the rotation angle become larger and larger, the mapping results become worse and worse; also worth notice that the mapping cannot fully handle the scaling of an image, that is to say, if we scale the test image too much, the mapping will have much more errors compared to original size.

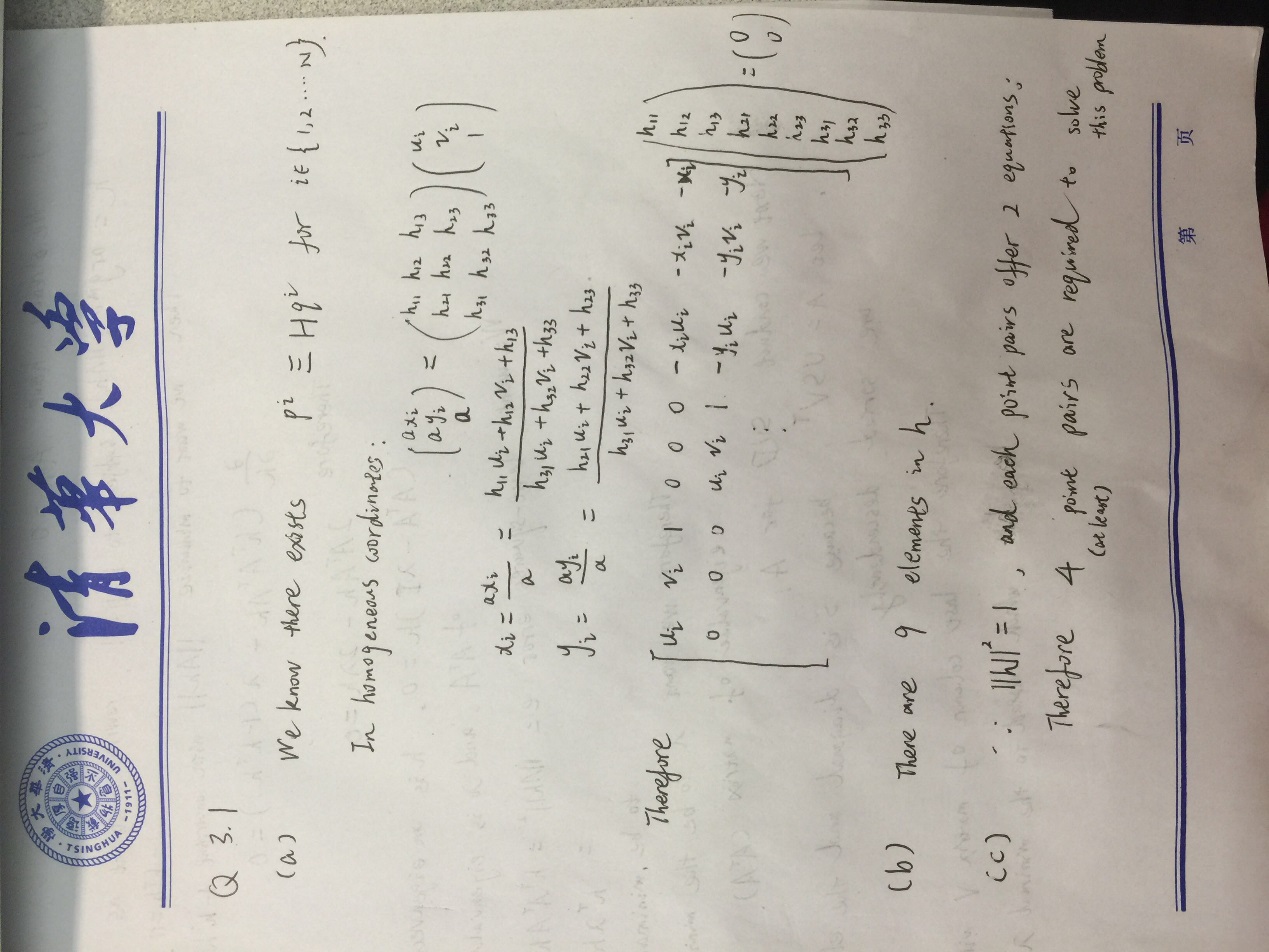
2.5

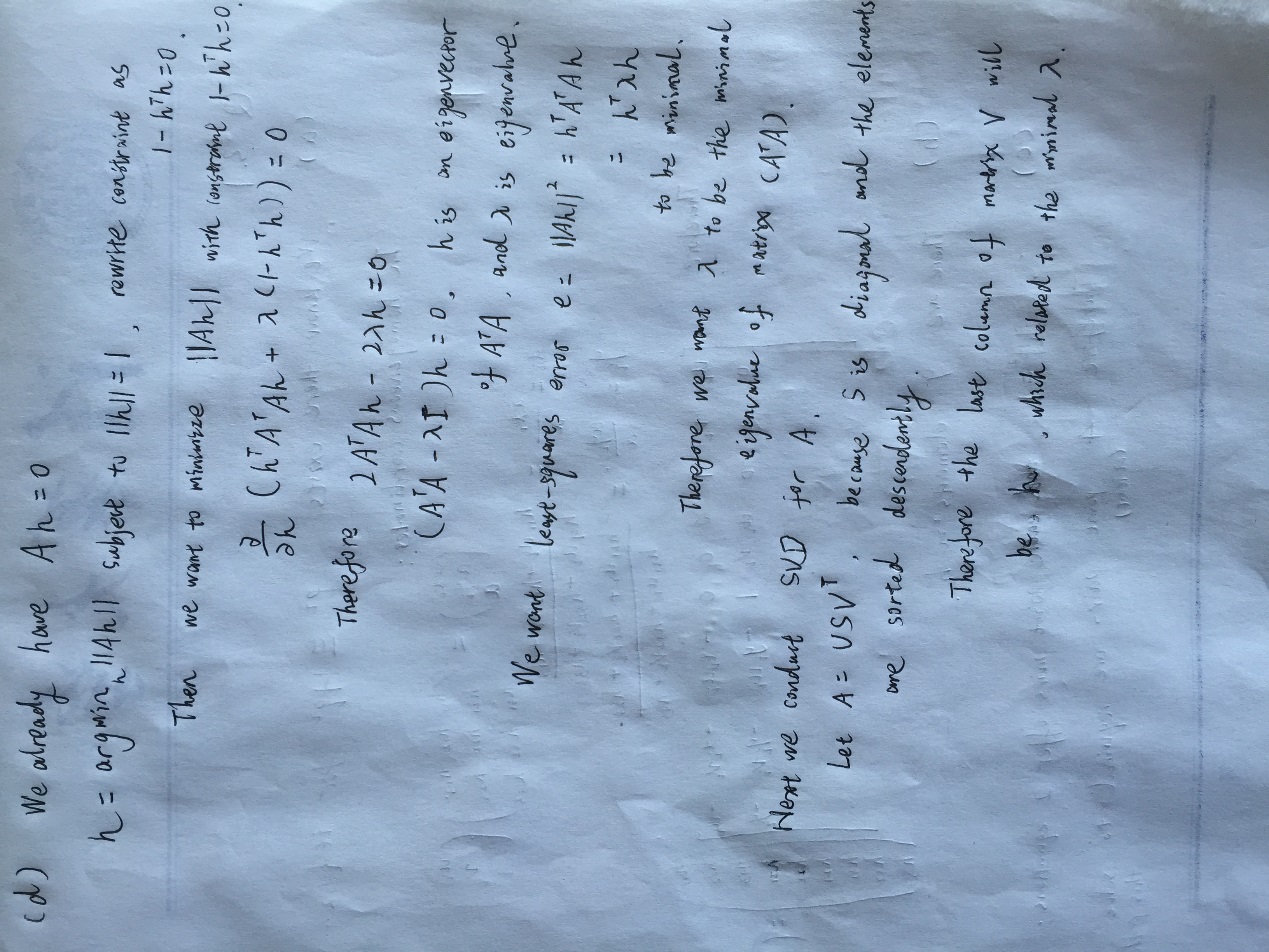
Here because of the time limit, I rotate the image from 0---90 degree.

The following bar is the amount of mappings. We can see with the rotate degree increases, the amount of mapping decrease. (From 0—90 degree) This is because of the BRIEF descriptor cannot handle well with the rotated image, so that the error will increase if we use the original image mapping with the rotated image.



3





4

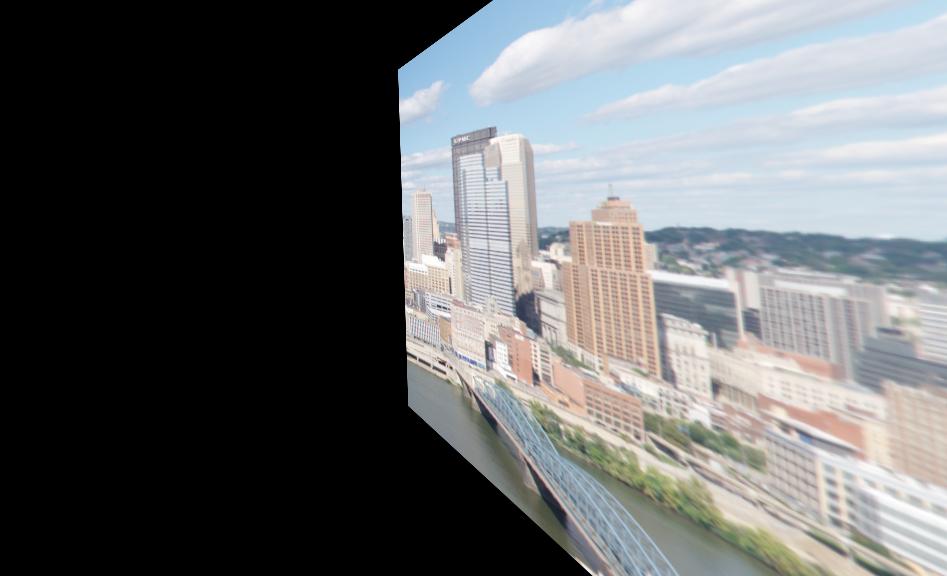
Please refer to computeH.m for detailed implementation.

5.1

Please refer to imageStitching.m for detailed implementation.

Already saved the q5\_1.jpg and q5\_1.mat in my submission.

**Note**: Because without Ransac, the error in computeH will ruin the wrapped image, so what I did here is to manually choose four correct mapping point pairs using ginput(), and formulate my H matrix, therefore I can the following wrapped image:



‘q5\_1.jpg’

5.2

Please refer to imageStitching\_noClip.m for detailed implementation.

Already saved ‘q5\_pan.jpg’ in my submission.



‘q5\_pan.jpg’

6.1

Please refer to ransacH.m for detailed implementation.

6.2

Please refer to generatePanorama.m for detailed implementation.



‘q6\_2.jpg’