csc420a2

Guanchen Zhang

October 2018

1 Question 1

1.1

Figure 1: Brown Corner Detection

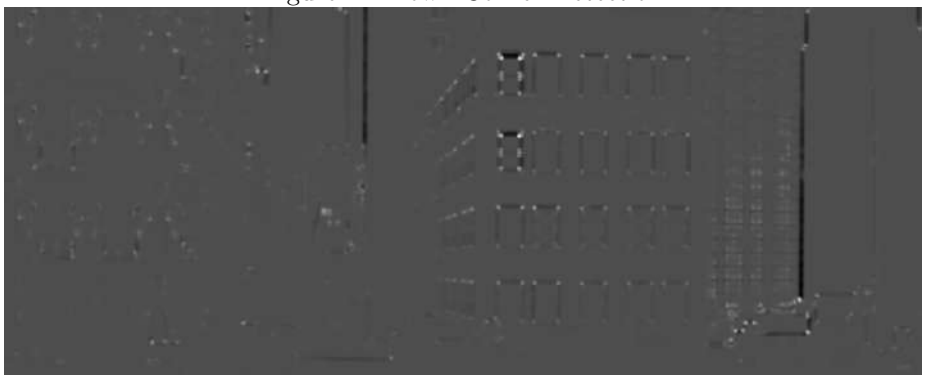


Figure 2: Harris Corner Detection

Both methods highlight the corners, while there are also some edges present using Harris method.

1.2

Figure 3: Non-maximum Suppression, r=20

By increasing the radii, r, less corner keypoints are included and the keypoints are displayed more sparse in the picture. This is because by increasing the radii, the value of the center point of interest must be larger than every other points in that circular mask, resulting in a stricter restriction in detecting the maximum than having a smaller radii. Therefore, more points are suppressed. However, if we keep a small radii, more false positive keypoints will be included which may not be significant corners. To conclude, we need to tune the radii carefully to pick a number which will include enough important corners but not too many.

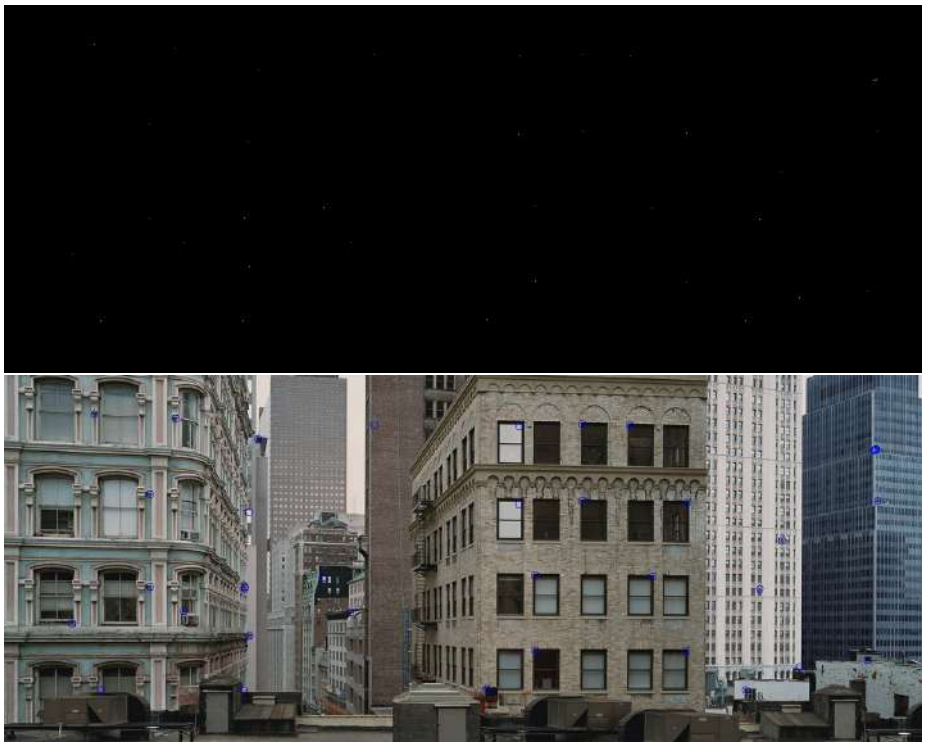
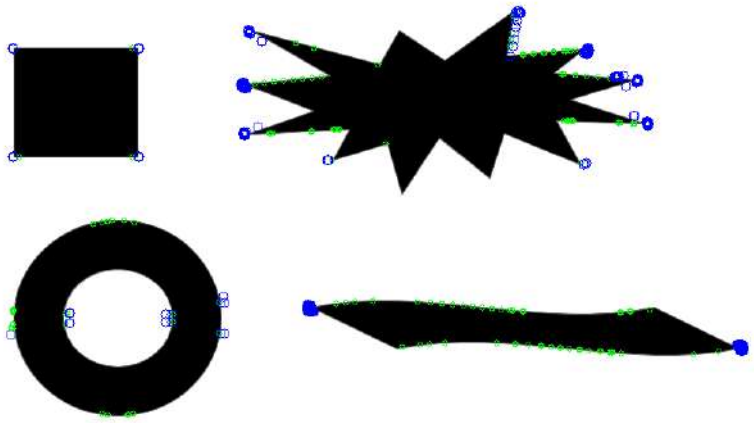


Figure 4: Non-maximum Suppression, r =50

1.3

see a1.py



BLOB detection

1.4

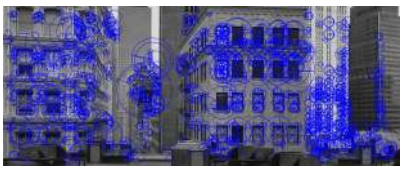


Figure 5: SIFT Detection on source image

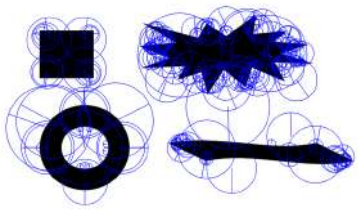


Figure 6: SIFT Detection on reference image

Similar to SIFT which approximates LoG with DoG for finding scale-space, SURF approximates LoG with Box Filter. The method is very fast because of the use of an integral image where the value of a pixel (x,y) is the sum of all values in the rectangle defined by the origin and (x,y). To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detect which can be computed with 3 integer operations using a precomputed integral image.
Unlike other methods, to measure interest points at different scale, scale spaces in SURF are implemented by applying box filters of different sizes. Accordingly, the scale space is analyzed by up-scaling the filter size rather than iteratively reducing the image size. Also, it is accelerated by computing its feature descriptor using the sum of the Haar wavelet response around the interest point.

2 Question 2

2.1

2.2

Since there is no way for us to know the ground-truth of matching so we are unable to plot the the percentage of true matches as a function of threshold, however, it is eligible for us to plot the number of matches as a function of threshold and we can still analyze which threshold gives the best feature correspondences.
We believe that setting the threshold to 0.8 is best because the number of matching grows smoothly until 0.8 and rockets from 0.8 to 0.9. It is reasonable to interpret the rapid increase as classifying too many mismatched pair as matched pairs.

Figure 7: SIFT Detection on source image

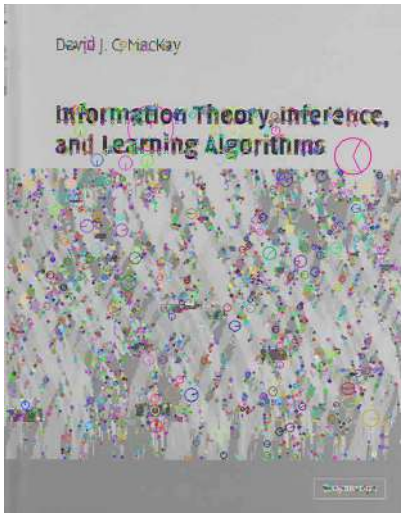


Figure 8: SIFT Detection on reference image

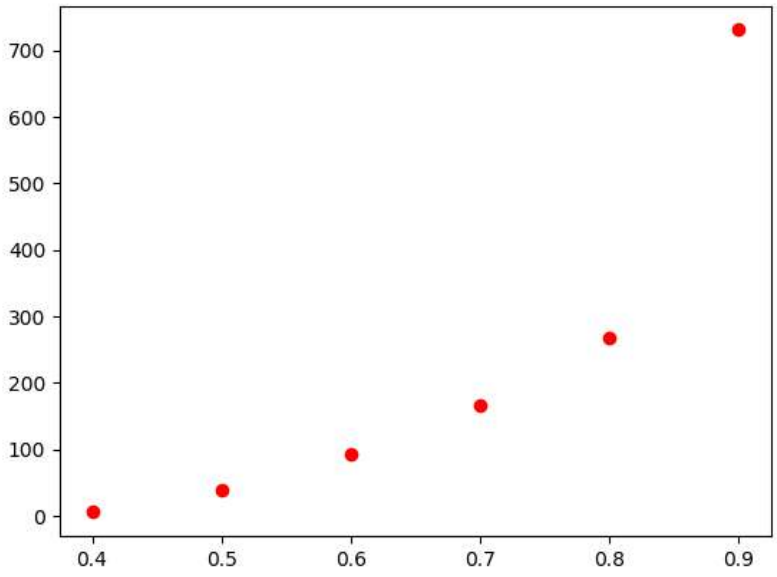


Figure 9: the number of matches as a function of threshold

Setting the threshold too high such as 0.9, resulted in too many false positives match because the closest match and second closest match may be too similar, meaning some pairs of matching that are equally false will not be filtered out by the high threshold. Setting the threshold too low results in too many false negatives, meaning that too many correct matches are missed.

2.3

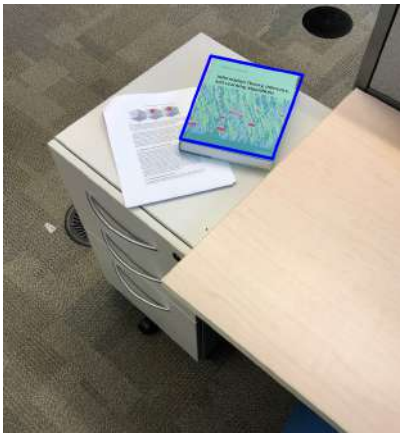


Figure 10: Affine transformation visualization, k=3

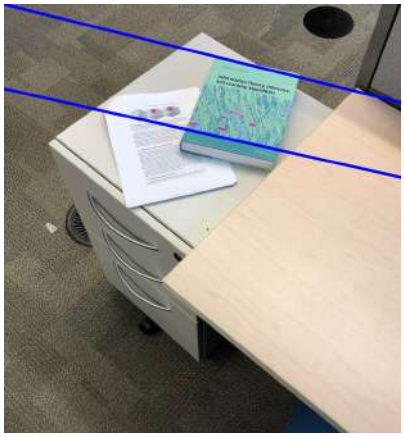


Figure 11: Affine transformation visualization, k=2

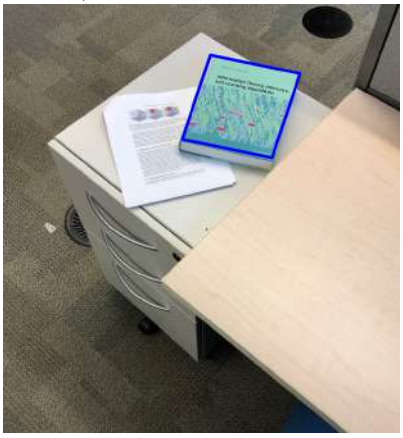


Figure 12: Affine transformation visualization, k=20

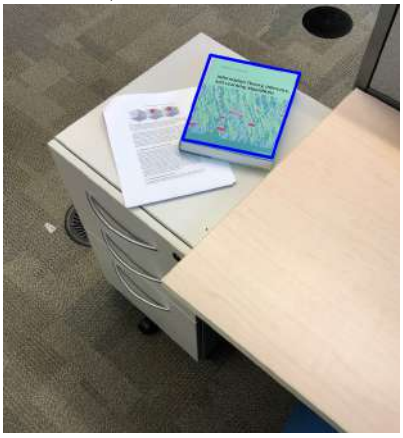


Figure 13: Affine transformation visualization, k=60

```
k=2: [ 14.94532788 -12.02895207   3.69497835  -2.68384172   0.73081127 0.17367556]
k=3:[3.90791505e-01  -1.01524478e-01   8.31315529e-02  \
```

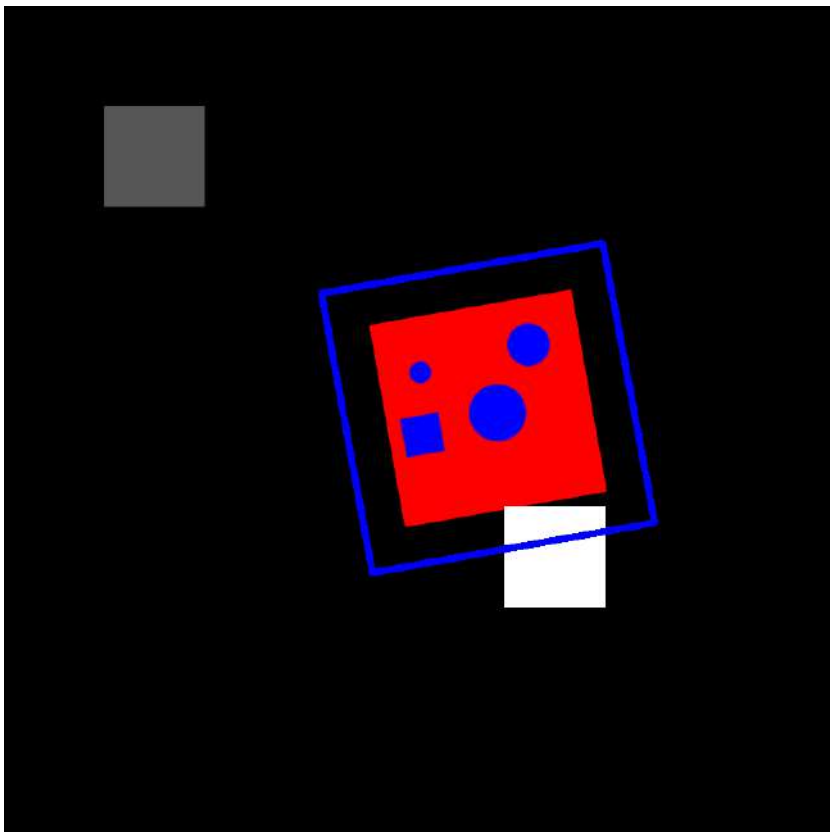
```
2.76063015e-01 4.94697976e+02   1.22756331e+02]
k=20:[3.80448766e-01  -9.31311578e-02   7.82305801e-02   \
2.79810913e-01 4.93526269e+02   1.20785114e+02]
k=60:[3.76244936e-01  -1.18803998e-01   7.73881220e-02   \
2.67849722e-01 5.06925911e+02   1.26283895e+02]
```

By playing with the number k which indicates the top k correspondences to solve for affine transformation, we can conclude that the minimum k required for solving the transformation is 3 because there are 6 variables in affine matrix need to be solved and 3 matches introduce 6 parameters. Which explains why we could not match the reference image to the origin image when $k = 2$
Compared to the visualization when $k = 20$, there is a slightly improvement to the visualization when $k = 2$. Interestingly, the visualization when $k = 60$ is even worse than that when $k = 3$ because the chances of including a false positive pair had increased.

2.4

See the figures above.

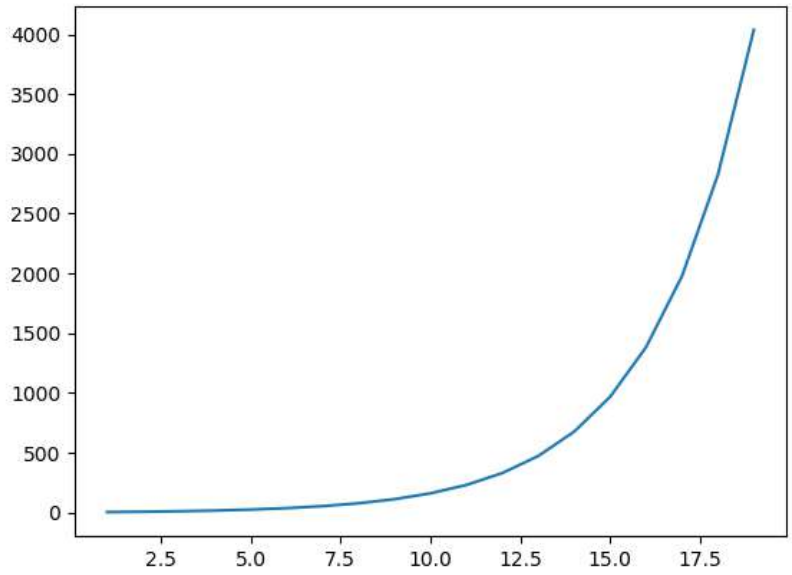
2.5



Addition to computing the distance between descriptors from grayscale image, to deal with rgb image input, we need to take the rgb channels into account when we calculate the distance between descriptors. I did so by appending the value of rgb channels into the descriptor corresponding to the keypoint. By multipling a weight to the rgb values in the descriptor, we are able to decide how much we should depends on the color of the keypoint. The rest of the algorithm is the same as the one we deal with grayscale image.

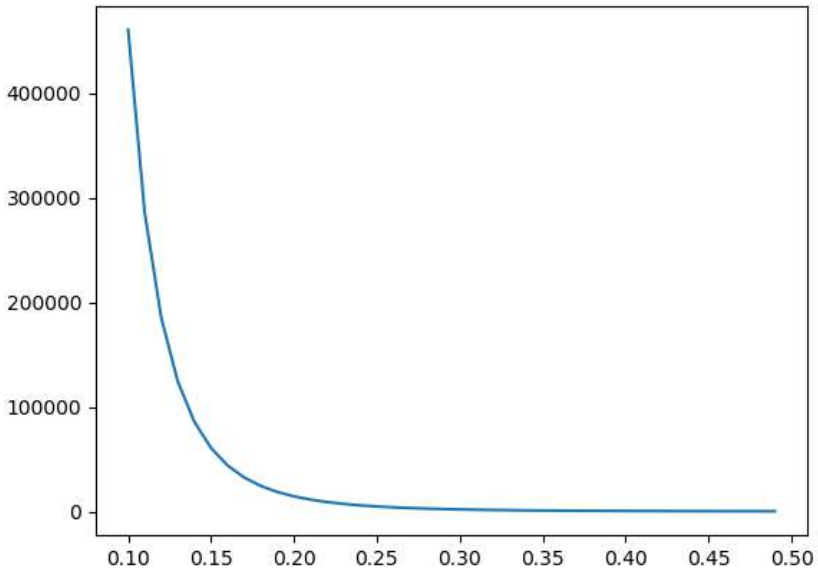
3 Question 3

3.1



Required iteration of RANSAC as a function of k

3.2



Required iteration of RANsAC as a function of percentage of outliers

3.3

The required number of iterations to recover the correct model with $P \geq 0.99$ is 14389.
In this case, there should be 300 inliers and at iteration 15, 450 points agrees with the current hypothesis. It is possible that all the 450 points found are the inliers in this hypothesis but not the ground-true inliers which are acquired under the model with largest set of inliers. In other words, it is still possible that in the later iterations, there is a model that can obtain even larger set of inliers (larger than 450).
It is also possible that it is haphazard that we get a very satisfying result at iteration 15 but next time we may find the set of inliers until the final iteration. Therefore, after iteration 14389, it is guaranteed that we can find the optimal set of inliers.
So the number of required iteration would not change.