Part 2

1. Visualizing the inliers for the input and the query image, placing them side by side:

Figure 1: RANSAC inliers match between Londoneye_12.jpg and Londoneye_16.jpg

Figure 2: Euclidean SIFT match between Londoneye_12.jpg and Londoneye_16.jpg

Figure 3: RANSAC inliers match between empirestate_14.jpg and empirestate_23.jpg

Figure 4: Euclidean SIFT match between empirestate_14.jpg and empirestate_23.jpg

Figure 5: RANSAC inliers match between sanmarco_1.jpg and sanmarco_19.jpg

Figure 6: Euclidean SIFT match between sanmarco_1.jpg and sanmarco_19.jpg

Figure 7: RANSAC inliers match between colosseum_8.jpg and bigben_14.jpg

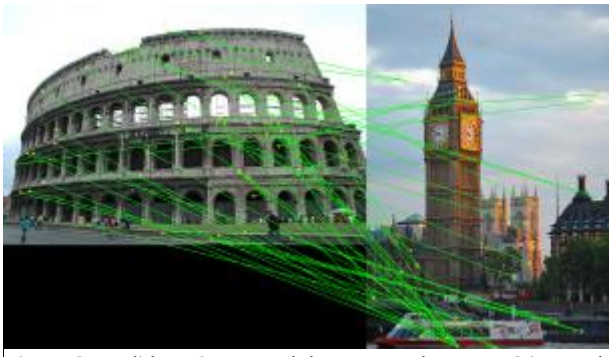

Figure 8: Euclidean SIFT match between colosseum_8.jpg and bigben_14.jpg

Do you notice cleaner correspondences?

Analysis:
Figure 1 – Figure 2:

- Figure 1: The number of points which matched with RANSAC inliers were fewer.
- Figure 2: Many of the points which correlated from the London eye passenger capsules from the left part of the image to the river Thames (black region) in right image were false positives and these points were removed with RANSAC (figure 1).

Figure 3 – Figure 4:

- Figure 3: The number of match points between the empirestate_14.jpg and empirestate_23.jpg are lesser as the angle from which the images are taken are drastically different and the intensity variation is more.
- Figure 4: There are lot of points which are matched from the building opposite to empire state building from the right image to empire state building in the left image which are false positives.

Figure 5 – Figure 6:

- Figure 5: A good number of points from the sanmarco building from the right image were mapped to the region, where there is nothing in the left image which suggest that these are not inliers. This figure shows some defects of random sampling from RANSAC which might have cause this kind of outliers to be matched.
- Figure 6: Euclidean Sift points matched well compared to RANSAC in this example. However, several points from people in the right image were mapped to the building in the second image, which suggest a need for clear correspondences.

Figure 7 – Figure 8:

- Figure 7: Since we are trying to map colosseum_8.jpg and bigben_14.jpg, which are completely different, the number of match points are zero. This represents the discrimination ability of RANSAC over simple Euclidean distance matching with SIFT points.
- Figure 8: Even though the two images (right and left image in figure 8) are completely different. Many points are matched because of intensity matching between the river on the right image with the building color of left image.

Repeat the experiments #3 of Part1. What is the impact on the precision?

Note: To run the code to get precision matching using RANSAC inliers, we have reused the multi_image_matching function from part1. To get precisions for RANSAC inliers, a small change is required in the input of multi_image_matching function. The change is 1 - RANSAC inliers.
The function is called in a2.cpp, but this function is implemented in ImageMatching.h header file.
The default value is 0 which does the Euclidean distance SIFT match between input and query image.

Analysis:
Results are as follows:

1. For input image trafalgarsquare_15.jpg :

Image : trafalgarsquare_25.jpg----matched by 2176.44 Euclidean Metric
Image : tatemodern_9.jpg----------matched by 1867.29 Euclidean Metric
Image : colosseum_6.jpg----------matched by 1765.44 Euclidean Metric
Image : tatemodern_2.jpg----------matched by 1748.45 Eculidian Metric
Image : notredame_1.jpg----------matched by 1679.65 Eculidian Metric
Image : trafalgarsquare_5.jpg-----matched by 1577.06 Eculidian Metric
Image : notredame_19.jpg----------matched by 1571.12 Eculidian Metric

Image : sanmarco_22.jpg----------matched by 1497.98 Eculidian Metric
Image : londoneye_17.jpg----------matched by 1480.6 Eculidian Metric
Image : empirestate_12.jpg--------matched by 1317.53 Eculidian Metric

2/10 = 20% precision

2. For input image louvre_16.jpg

Image : notredame_19.jpg----------matched by 3024.22 Eculidian Metric
Image : bigben_8.jpg----------matched by 2435.66 Eculidian Metric
Image : louvre_10.jpg----------matched by 2003.66 Eculidian Metric
Image : sanmarco_3.jpg----------matched by 1882.17 Eculidian Metric
Image : louvre_11.jpg----------matched by 1763.22 Eculidian Metric
Image : colosseum_12.jpg----------matched by 1694.61 Eculidian Metric
Image : louvre_14.jpg----------matched by 1536.64 Eculidian Metric
Image : notredame_8.jpg----------matched by 1484.23 Eculidian Metric
Image : sanmarco_20.jpg----------matched by 1458.53 Eculidian Metric
Image : sanmarco_13.jpg----------matched by 1383.61 Eculidian Metric

3/10 = 30%

3. For input image sanmarco_14.jpg

Image : empirestate_25.jpg--------matched by 2059.26 Eculidian Metric
Image : empirestate_27.jpg--------matched by 1748.28 Eculidian Metric
Image : eiffel_7.jpg----------matched by 1662.36 Eculidian Metric
Image : londoneye_13.jpg----------matched by 1643.9 Eculidian Metric
Image : sanmarco_20.jpg----------matched by 1636.55 Eculidian Metric
Image : louvre_4.jpg----------matched by 1502.28 Eculidian Metric
Image : empirestate_12.jpg--------matched by 1471.74 Eculidian Metric
Image : eiffel_15.jpg----------matched by 1341.52 Eculidian Metric
Image : empirestate_14.jpg--------matched by 1264.97 Eculidian Metric
Image : louvre_11.jpg----------matched by 1238.91 Eculidian Metric

2/10 = 20%

4. For input image bigben_14.jpg

Image : notredame_3.jpg----------matched by 2595.58 Eculidian Metric
Image : londoneye_13.jpg----------matched by 2410.1 Eculidian Metric
Image : tatemodern_4.jpg----------matched by 2013.23 Eculidian Metric
Image : bigben_3.jpg----------matched by 2011.45 Eculidian Metric
Image : bigben_12.jpg----------matched by 1918.96 Eculidian Metric
Image : notredame_1.jpg----------matched by 1854.75 Eculidian Metric
Image : bigben_8.jpg----------matched by 1731.19 Eculidian Metric
Image : tatemodern_9.jpg----------matched by 1722.91 Eculidian Metric
Image : bigben_16.jpg----------matched by 1714.27 Eculidian Metric
Image : trafalgarsquare_6.jpg-----matched by 1589.88 Eculidian Metric
Image : bigben_10.jpg----------matched by 1576.91 Eculidian Metric

5/10 = 50%

5. For input image colosseum_4.jpg

Image : colosseum_5.jpg----------matched by 16972.7 Eculidian Metric
Image : tatemodern_4.jpg----------matched by 2267.79 Eculidian Metric
Image : colosseum_6.jpg----------matched by 1532.28 Eculidian Metric
Image : eiffel_3.jpg----------matched by 1466.02 Eculidian Metric
Image : louvre_11.jpg----------matched by 1306.56 Eculidian Metric

Image : eiffel_6.jpg----------matched by 1187.14 Eculidian Metric
Image : tatemodern_13.jpg---------matched by 1184.41 Eculidian Metric
Image : louvre_10.jpg----------matched by 1170.42 Eculidian Metric
Image : trafalgarsquare_21.jpg----matched by 1163.47 Eculidian Metric
Image : eiffel_2.jpg----------matched by 1121.98 Eculidian Metric

2/10 = 20%

6. For input image notredame_4.jpg

Image : notredame_5.jpg----------matched by 9094.7 Eculidian Metric
Image : notredame_3.jpg----------matched by 9038.49 Eculidian Metric
Image : londoneye_16.jpg----------matched by 3928.67 Eculidian Metric
Image : louvre_10.jpg----------matched by 2389.67 Eculidian Metric
Image : colosseum_15.jpg----------matched by 2375.81 Eculidian Metric
Image : sanmarco_20.jpg----------matched by 2219.03 Eculidian Metric
Image : louvre_14.jpg----------matched by 2185.34 Eculidian Metric
Image : tatemodern_2.jpg----------matched by 2169.94 Eculidian Metric
Image : louvre_9.jpg----------matched by 2088.42 Eculidian Metric
Image : bigben_8.jpg----------matched by 1966.35 Eculidian Metric

2/10 = 20%

7. For input image londoneye_9.jpg

Image : londoneye_2.jpg----------matched by 4488.84 Eculidian Metric
Image : bigben_14.jpg----------matched by 4243.47 Eculidian Metric
Image : sanmarco_19.jpg----------matched by 3835.03 Eculidian Metric
Image : bigben_2.jpg----------matched by 3404.26 Eculidian Metric
Image : sanmarco_13.jpg----------matched by 3256.04 Eculidian Metric
Image : bigben_10.jpg----------matched by 2906.36 Eculidian Metric
Image : londoneye_23.jpg----------matched by 2527.01 Eculidian Metric
Image : louvre_14.jpg----------matched by 2522.16 Eculidian Metric
Image : londoneye_21.jpg----------matched by 2445.41 Eculidian Metric
Image : bigben_12.jpg----------matched by 2382.42 Eculidian Metric

2/10 = 20%

8. For input image eiffel_5.jpg

Image : bigben_12.jpg----------matched by 2297.09 Eculidian Metric
Image : colosseum_15.jpg----------matched by 1850.02 Eculidian Metric
Image : trafalgarsquare_15.jpg----matched by 1724.28 Eculidian Metric
Image : notredame_19.jpg----------matched by 1614.37 Eculidian Metric
Image : bigben_6.jpg----------matched by 1536.17 Eculidian Metric
Image : empirestate_27.jpg--------matched by 1523.46 Eculidian Metric
Image : eiffel_2.jpg----------matched by 1447.92 Eculidian Metric
Image : louvre_15.jpg----------matched by 1440.84 Eculidian Metric
Image : eiffel_22.jpg----------matched by 1387.24 Eculidian Metric
Image : sanmarco_18.jpg----------matched by 1333.73 Eculidian Metric

2/10 = 20%

9. For input image bigben_8.jpg

Image : bigben_6.jpg----------matched by 17783.8 Eculidian Metric
Image : londoneye_8.jpg----------matched by 1650.15 Eculidian Metric
Image : sanmarco_22.jpg----------matched by 1596.58 Eculidian Metric
Image : bigben_12.jpg----------matched by 1542.5 Eculidian Metric
Image : londoneye_21.jpg----------matched by 1499.08 Eculidian Metric

Image : notredame_25.jpg----------matched by 1454.9 Eculidian Metric
Image : colosseum_12.jpg----------matched by 1411.2 Eculidian Metric
Image : bigben_14.jpg----------matched by 1387.89 Eculidian Metric
Image : empirestate_22.jpg--------matched by 1363.94 Eculidian Metric
Image : sanmarco_18.jpg----------matched by 1340.47 Eculidian Metric

3/10 = 30%

10. For input image sanmarco_14.jpg

Image : londoneye_22.jpg----------matched by 3432.12 Eculidian Metric
Image : eiffel_15.jpg----------matched by 2674.46 Eculidian Metric
Image : sanmarco_3.jpg----------matched by 2629.01 Eculidian Metric
Image : eiffel_22.jpg----------matched by 2355.04 Eculidian Metric
Image : tatemodern_24.jpg---------matched by 2323.7 Eculidian Metric
Image : tatemodern_9.jpg----------matched by 2308.08 Eculidian Metric
Image : louvre_15.jpg----------matched by 2026.61 Eculidian Metric
Image : bigben_8.jpg----------matched by 1961.92 Eculidian Metric
Image : trafalgarsquare_8.jpg-----matched by 1890.08 Eculidian Metric
Image : bigben_7.jpg----------matched by 1796.06 Eculidian Metric

1/10 = 10%

The average overall precision is 24% which is still better than random guessing. We did get some good precision on some easy images and some hard images has fewer precision points.

2. Does the algorithm speed up the image retrieval, and if so, by how much?

Analysis: The algorithm does speed up the image retrieval process.

Input images	**Projection algorithm running time in seconds**	**Euclidean SIFT matching algorithm running time in seconds**
eiffel_1.jpg & eiffel_3.jpg	0.26	0.47
trafalgarsquare_21.jpg & trafalgarsquare_16.jpg	0.64	1.06
tatemodern_16.jpg & tatemodern_24.jpg	0.180000	0.260000
londoneye_8.jpg & londoneye_13.jpg	0.10000	0.150000

These values are reported with k = 10, and w = 25. The reason for choosing this is explained in the next part.
We conclude from the table that, the running time for the projection algorithm is 50% of the running time for the Euclidean SIFT matching algorithm.

Does the approximation affect the quality of the results?

Analysis:

Figure 9: Euclidean SIFT match between Londoneye_8.jpg and Londoneye_13.jpg

Figure 10: Projection algorithm SIFT match between Londoneye_8.jpg and Londoneye_13.jpg

Figure 11: Euclidean SIFT match between tatemodern_16.jpg and tatemodern_24.jpg

Figure 12: Projection algorithm SIFT match between tatemodern_16.jpg and tatemodern_24.jpg

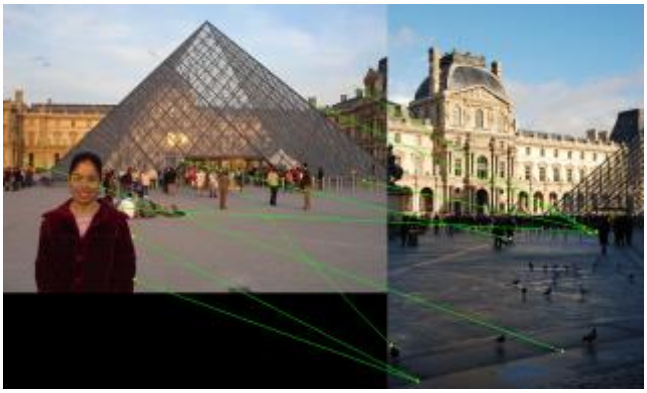

Figure 11: Euclidean SIFT match between louvre_4.jpg and louvre_8.jpg

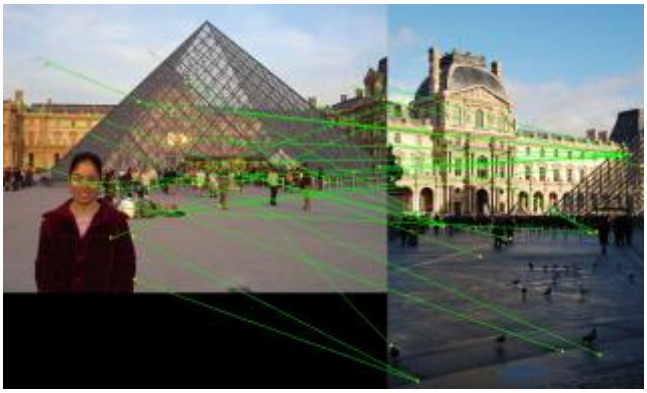

Figure 12: Projection algorithm SIFT match between louvre_4.jpg and louvre_8.jpg

As we can observe from figure 9 through figure 12, we could not find a huge tradeoff between the corresponding matches between the images with Projection algorithm and Euclidean SIFT matches.
The k value chosen for the above images is 10, along with L2norm of vector x for w.
Since the projection parameter w is a function of the random probability vector x, the value of k did not have an impact on the number of SIFT matches found between the input and the query image.

What is the best trade-off you can achieve between running time and accuracy, by adjusting w, k and the number of trials?

Analysis:
We have used a random uniform distribution which takes values between open interval 0 and 1 are considered for projection algorithm. The random distribution is of length 128 and this is multiplied with the SIFT point, which gives the expectation of that 128 dimension SIFT vector with random distribution. We choose k such expected values for each SIFT vector to represent it with

lesser dimensions. A projection parameter w is divided to get the final projection point of each SIFT vector from a higher dimension to lower.
For experiments are as follows:
For K = 1:

W parameter	Naïve SIFT match running time in seconds	Projection algorithm running time in seconds	Number of SIFT points match using Naïve SIFT match	Number of SIFT points match using Projection algorithm
25	0.15	0.1	45	66
50	0.14	0.16	45	62
100	0.15	0.18	45	63

For K = 10:

W parameter	Naïve SIFT match running time in seconds	Projection algorithm running time in seconds	Number of SIFT points match using Naïve SIFT match	Number of SIFT points match using Projection algorithm
25	0.15	0.05	45	39
50	0.15	0.08	45	65
100	0.15	0.18	45	61

For K = 20:

W parameter	Naïve SIFT match running time in seconds	Projection algorithm running time in seconds	Number of SIFT points match using Naïve SIFT match	Number of SIFT points match using Projection algorithm
25	0.14	0.05	45	39
50	0.14	0.07	45	57
100	0.15	0.14	45	55

From the tables, we conclude that, as the value of w increases beyond 50, for values of k below 20, the SIFT matches are increasing.
Reasoning:
As the value of k decreases the number of points representing SIFT vector decreases which suggest compression of data and hence loss of data. Because of this there will be many SIFT points between input and query image which will have similar summary vectors calculated from the projection algorithm resulting in increases number of SIFT match points. However, we can control the degree of projection using parameter w. The parameter w should be large for differentiation between two separate vectors. But large values of w again cause the same problem as the lower k values cause, which is explained before.
We must choose a value of lesser value of k to give us a gain in algorithm's time complexity and value of w large to separate two dissimilar SIFT vectors.
This is difficult task, but based on trial and error and observing the output we have concluded that **k = 10 and w = 50** would provide SIFT matches with running time less than 50% of the naïve SIFT match algorithm.