Homework Assignment 4

CS696, Applied Computer Vision

In the assignment, a local feature matching algorithm is created. The matching pipeline is intended to work for an instance-level matching which includes multiple views of the same physical scene.

For this project, I have implemented the three major functions of a local feature matching algorithm:

* Interest point detection in get\_interest\_points.m
* Local feature description in get\_features.m
* Feature Matching in match\_features.m

**ALGORITHM**

The 3 major steps of a local feature matching algorithm implemented are:

**Interest Point Detection:**

This function is written in get\_interest\_points.m file. The position of the features are first supposed to be selected. This will help create features to match images against their interest points. One of such reliable interest points can be considered as edges and corners. Corners are more reliable as they usually are different than the neighboring points. The Harris Corner Detector is an algorithm used for the detection of corners in an image.

* **Harris Corner Detector**

The book of Szeliski referred in the assignment is used for the algorithm of Harris Corner Detector. The Harris Corner detector works by finding the horizontal and vertical derivatives of the images Ix and Iy. To perform this, each image is filtered along with the Gaussian Derivative (Sobel filter). Next the derivatives are squared to form two squared derivatives images and also an x-derivative times y-derivative image. The 3 images corresponding to the outer products of the gradients are determined. These images are convoluted with a larger gradient. Lastly, local maxima of responses above a certain threshold are found and considered as detected feature point locations.

**Algorithm:**

1. Compute the horizontal and vertical derivatives of the image Ix and Iy by convolving the original image with derivatives of Gaussians.

2. Compute the three images corresponding to the outer products of these gradients. (The matrix A is symmetric, so only three entries are needed.)

3. Convolve each of these images with a larger Gaussian.

4. Compute a scalar interest measure using one of the formulas discussed above.

5. Find local maxima above a certain threshold and report them as detected feature point locations

**Local Feature Description:**

This function is written in get\_features.m file. Once the interest points are selected, the features must be created from those selected points. For the local feature description SIFT-like features are used. The following steps are used for the local feature description:

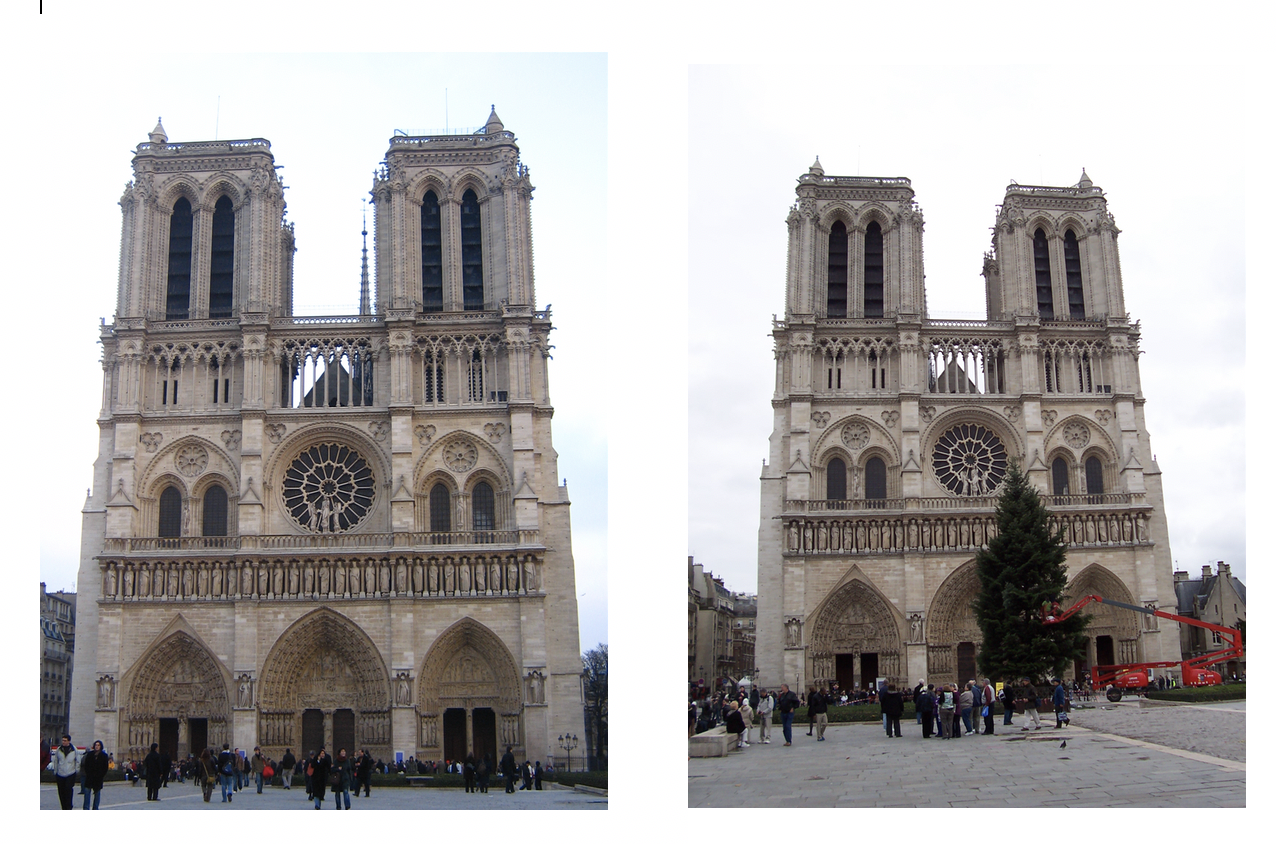
* SIFT features are formed by computing the gradient at each pixel in a 16×16 window around the detected feature point, using the appropriate level of the Gaussian pyramid at which the feature point was detected.
* The gradient magnitudes are decreased by a Gaussian fall-off function in order to reduce the influence of gradients far from the center, as these are more affected by smaller errors.
* In each 4 × 4 quadrant, a gradient orientation histogram is formed by adding the weighted gradient value to one of eight orientation histogram bins = 128 dimensions
* For simplicity, each pixel only contributes its gradient to the orientation it is closest to.
* Softly distributing values to adjacent histogram bins is generally a good idea in any application where histograms are being computed.
* The resulting 128 values form a raw version of the SIFT descriptor vector. To reduce the effects of contrast, the 128-D vector is normalized to unit length for an interest point.

**Feature Matching:**

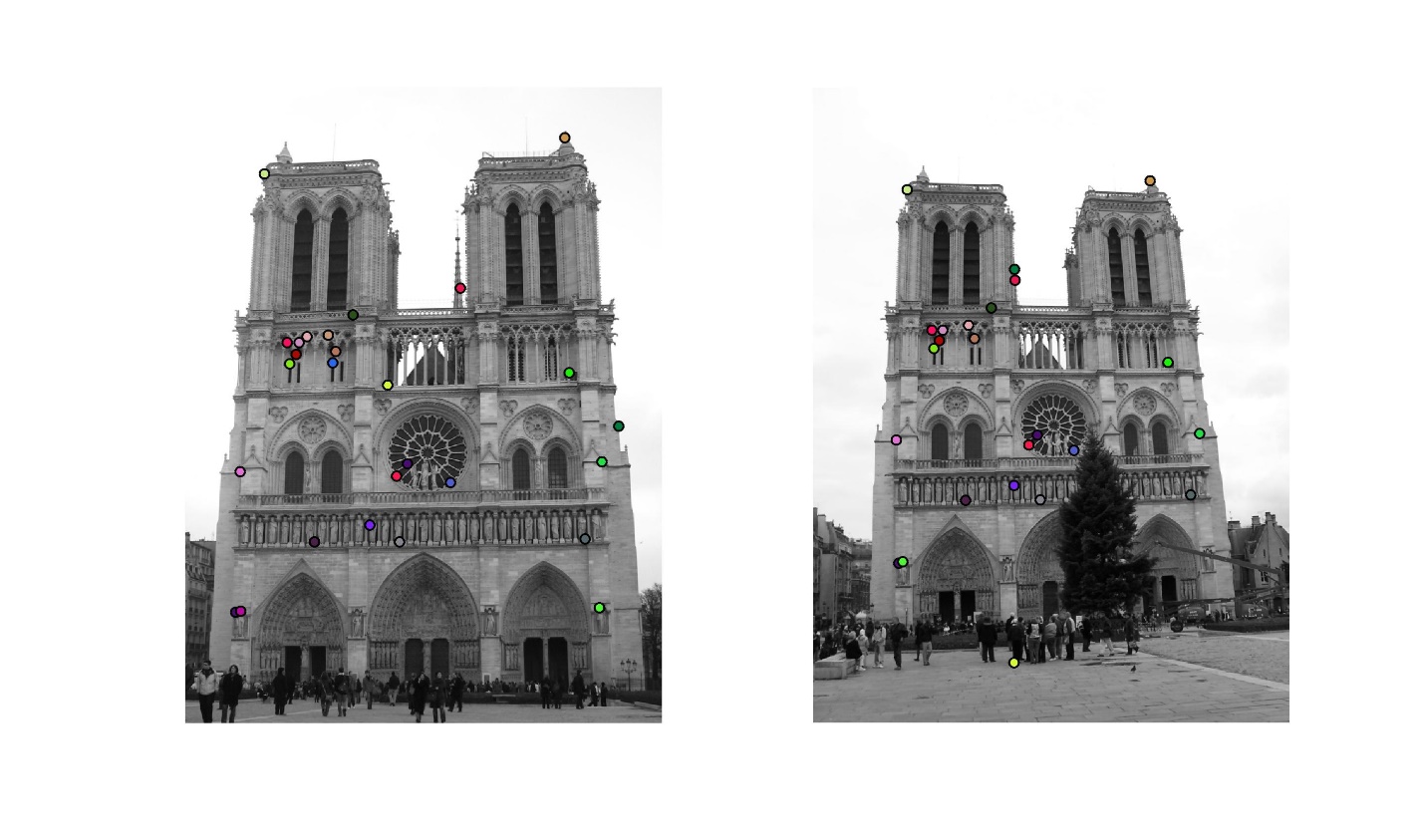
* This function is written in match\_features.m file. For the feature matching algorithm, The Euclidean distance of Image 1 feature vectors to Image 2 feature vectors is calculated.
* These distances are then used to calculate the Nearest Neighbor Distance Ratio (1st nearest neighbor/2nd nearest neighbor) which are then stored.
* These ratios are then sorted and if the value of each ratio is less than the threshold, the confidence value is divided by it.
* Different threshold values are then used to determine the optimum value.
* If the ratio is low enough (a threshold value that we can manipulate as a parameter) then it can be considered as a valid match.
* The threshold value considered by me was 7.5, as this was not too low to eliminate all points but was also not to concerning about which matches to choose.
* A higher threshold can determine more matches, but can result in less accuracy.

**Input:**

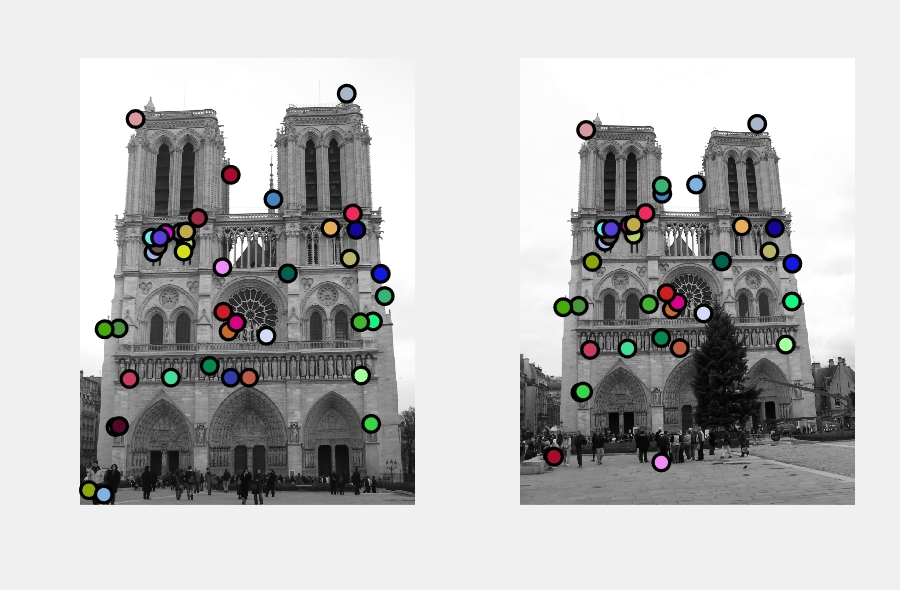
* 'features1' and 'features2' are the n x m dimensionality features from the two images.
* For each feature in image 1, the nearest neighbor distance ratio is computed with respect to all features in image 2.
* If the ratio is less than a threshold value (0.75), the nearest neighbor of the feature is considered a match and included in the result.
* Sort the matches so that the most confident ones are at the top of the list.

**Input Images: -**

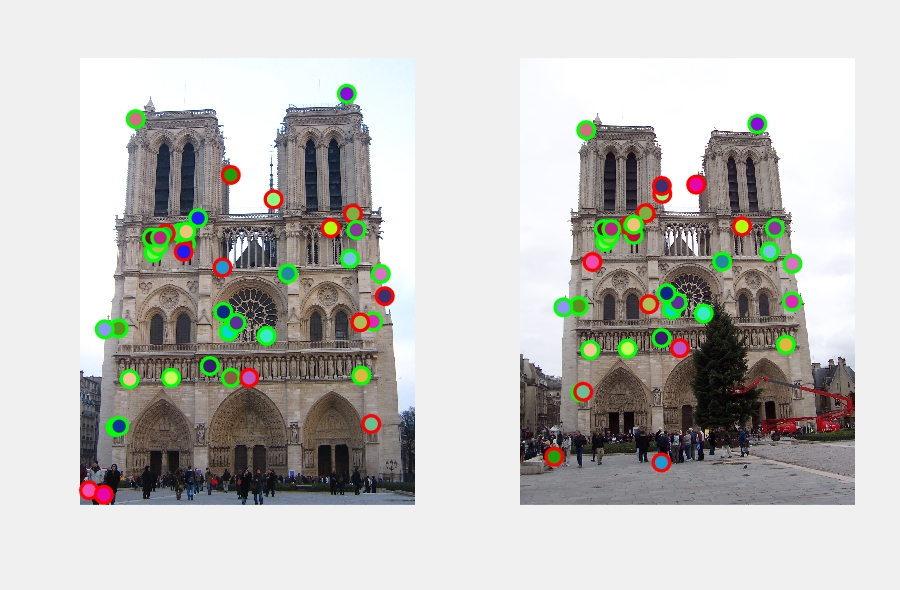
**Output:**

**1: Connected Components**

**(Ground-Truth:)**

****

**(Evaluation:)**

****

**Results:**

**Saving visualization to vis.jpg**

**( 486, 852) to ( 518, 804) g.t. point 34 px. Match error 4 px. correct**

**( 164, 1692) to ( 276, 1522) g.t. point 17 px. Match error 3 px. correct**

**( 476, 888) to ( 388, 836) g.t. point 4 px. Match error 121 px. incorrect**

**( 338, 892) to ( 388, 836) g.t. point 3 px. Match error 3 px. correct**

**( 682, 1254) to ( 692, 1144) g.t. point 67 px. Match error 10 px. correct**

**( 462, 800) to ( 498, 762) g.t. point 40 px. Match error 1 px. correct**

**( 418, 1466) to ( 490, 1322) g.t. point 65 px. Match error 5 px. correct**

**( 542, 734) to ( 572, 706) g.t. point 32 px. Match error 18 px. correct**

**( 654, 960) to ( 644, 1840) g.t. point 4 px. Match error 942 px. incorrect**

**( 180, 1688) to ( 288, 1518) g.t. point 5 px. Match error 2 px. correct**

**( 358, 862) to ( 406, 810) g.t. point 33 px. Match error 7 px. correct**

**( 1344, 1208) to ( 1238, 1108) g.t. point 56 px. Match error 9 px. correct**

**( 692, 1466) to ( 728, 1320) g.t. point 11 px. Match error 6 px. correct**

**( 1290, 1456) to ( 1210, 1304) g.t. point 44 px. Match error 1 px. correct**

**( 330, 824) to ( 382, 778) g.t. point 66 px. Match error 13 px. correct**

**( 1224, 164) to ( 1080, 300) g.t. point 19 px. Match error 4 px. correct**

**( 888, 648) to ( 648, 618) g.t. point 154 px. Match error 196 px. incorrect**

**( 394, 804) to ( 498, 762) g.t. point 31 px. Match error 68 px. incorrect**

**( 596, 1412) to ( 644, 1274) g.t. point 56 px. Match error 10 px. correct**

**( 178, 1240) to ( 268, 1128) g.t. point 16 px. Match error 3 px. correct**

**( 856, 1274) to ( 836, 1162) g.t. point 19 px. Match error 7 px. correct**

**( 1238, 920) to ( 1136, 880) g.t. point 20 px. Match error 2 px. correct**

**( 718, 1214) to ( 718, 1112) g.t. point 24 px. Match error 17 px. correct**

**( 256, 280) to ( 302, 328) g.t. point 15 px. Match error 4 px. correct**

**( 368, 824) to ( 416, 778) g.t. point 63 px. Match error 14 px. correct**

**( 1398, 1092) to ( 646, 582) g.t. point 69 px. Match error 742 px. incorrect**

**( 1336, 1678) to ( 288, 1518) g.t. point 2 px. Match error 974 px. incorrect**

**( 42, 1982) to ( 330, 928) g.t. point 231 px. Match error 879 px. incorrect**

**( 1252, 714) to ( 572, 706) g.t. point 58 px. Match error 565 px. incorrect**

**( 114, 1244) to ( 198, 1134) g.t. point 4 px. Match error 4 px. correct**

**( 1380, 990) to ( 1240, 936) g.t. point 27 px. Match error 12 px. correct**

**( 1268, 788) to ( 1160, 774) g.t. point 33 px. Match error 9 px. correct**

**( 692, 536) to ( 156, 1814) g.t. point 187 px. Match error 1412 px. incorrect**

**( 776, 1466) to ( 728, 1320) g.t. point 82 px. Match error 70 px. incorrect**

**( 488, 796) to ( 518, 758) g.t. point 65 px. Match error 6 px. correct**

**( 1286, 1212) to ( 588, 1120) g.t. point 7 px. Match error 610 px. incorrect**

**( 110, 2002) to ( 802, 576) g.t. point 187 px. Match error 1364 px. incorrect**

**( 954, 988) to ( 920, 926) g.t. point 52 px. Match error 2 px. correct**

**( 658, 1164) to ( 668, 1070) g.t. point 17 px. Match error 3 px. correct**

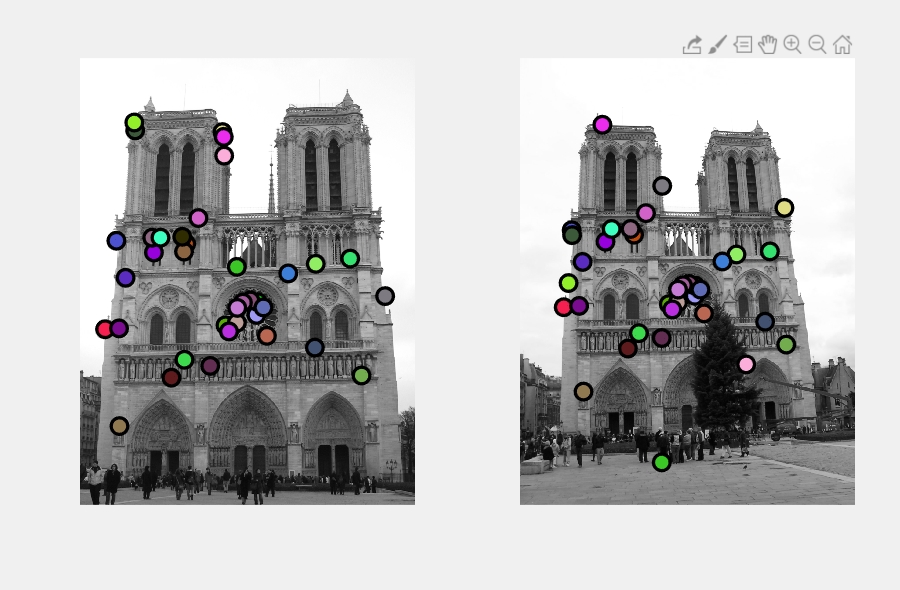
**( 1150, 780) to ( 1010, 766) g.t. point 77 px. Match error 38 px. incorrect**

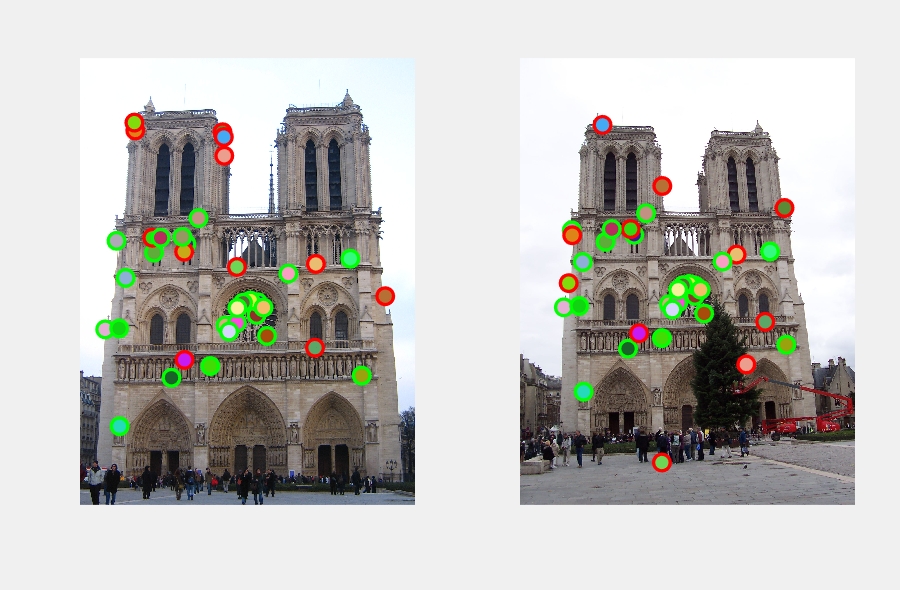
**( 228, 1472) to ( 320, 1328) g.t. point 35 px. Match error 4 px. correct**

**28 total good matches, 13 total bad matches**

**Saving visualization to eval.jpg**

**2: Sliding Window**

**(Ground-Truth:)**

**(Evaluated:)**

**Results:**

**Saving visualization to vis.jpg**

**( 488, 852) to ( 520, 804) g.t. point 34 px. Match error 4 px. correct**

**( 810, 1178) to ( 796, 1082) g.t. point 28 px. Match error 4 px. correct**

**( 668, 1226) to ( 678, 1120) g.t. point 46 px. Match error 9 px. correct**

**( 168, 838) to ( 236, 784) g.t. point 55 px. Match error 12 px. correct**

**( 478, 888) to ( 390, 836) g.t. point 3 px. Match error 121 px. incorrect**

**( 340, 892) to ( 390, 836) g.t. point 3 px. Match error 3 px. correct**

**( 182, 1688) to ( 290, 1518) g.t. point 5 px. Match error 2 px. correct**

**( 816, 1120) to ( 798, 1036) g.t. point 36 px. Match error 8 px. correct**

**( 1400, 1092) to ( 648, 582) g.t. point 68 px. Match error 742 px. incorrect**

**( 782, 1112) to ( 770, 1028) g.t. point 3 px. Match error 2 px. correct**

**( 1292, 1456) to ( 1212, 1304) g.t. point 42 px. Match error 1 px. correct**

**( 470, 820) to ( 504, 778) g.t. point 54 px. Match error 4 px. correct**

**( 116, 1244) to ( 200, 1134) g.t. point 3 px. Match error 4 px. correct**

**( 254, 332) to ( 238, 806) g.t. point 56 px. Match error 455 px. incorrect**

**( 544, 734) to ( 574, 706) g.t. point 34 px. Match error 18 px. correct**

**( 842, 1144) to ( 822, 1054) g.t. point 55 px. Match error 17 px. correct**

**( 662, 448) to ( 1030, 1394) g.t. point 101 px. Match error 974 px. incorrect**

**( 180, 1240) to ( 270, 1128) g.t. point 16 px. Match error 3 px. correct**

**( 720, 958) to ( 646, 1840) g.t. point 55 px. Match error 941 px. incorrect**

**( 332, 824) to ( 504, 778) g.t. point 66 px. Match error 124 px. incorrect**

**( 654, 340) to ( 1204, 682) g.t. point 30 px. Match error 645 px. incorrect**

**( 720, 1214) to ( 720, 1112) g.t. point 22 px. Match error 17 px. correct**

**( 748, 1122) to ( 742, 1034) g.t. point 36 px. Match error 5 px. correct**

**( 598, 1412) to ( 646, 1274) g.t. point 55 px. Match error 10 px. correct**

**( 370, 824) to ( 418, 778) g.t. point 61 px. Match error 14 px. correct**

**( 684, 1254) to ( 694, 1144) g.t. point 65 px. Match error 10 px. correct**

**( 858, 1274) to ( 838, 1162) g.t. point 18 px. Match error 7 px. correct**

**( 210, 1008) to ( 286, 928) g.t. point 29 px. Match error 4 px. correct**

**( 722, 1146) to ( 720, 1054) g.t. point 68 px. Match error 9 px. correct**

**( 1082, 944) to ( 986, 894) g.t. point 8 px. Match error 29 px. incorrect**

**( 420, 1466) to ( 492, 1322) g.t. point 66 px. Match error 5 px. correct**

**( 1078, 1328) to ( 1116, 1200) g.t. point 40 px. Match error 94 px. incorrect**

**( 480, 1384) to ( 542, 1252) g.t. point 82 px. Match error 26 px. incorrect**

**( 1240, 920) to ( 1138, 880) g.t. point 18 px. Match error 2 px. correct**

**( 250, 296) to ( 222, 1024) g.t. point 29 px. Match error 684 px. incorrect**

**( 660, 362) to ( 376, 304) g.t. point 52 px. Match error 280 px. incorrect**

**( 956, 988) to ( 922, 926) g.t. point 54 px. Match error 2 px. correct**

**25 total good matches, 12 total bad matches**

**Saving visualization to eval.jpg**

**Other additional Input pair:**

For the additional images, I have added the ground-truth manually:

**Steps:**

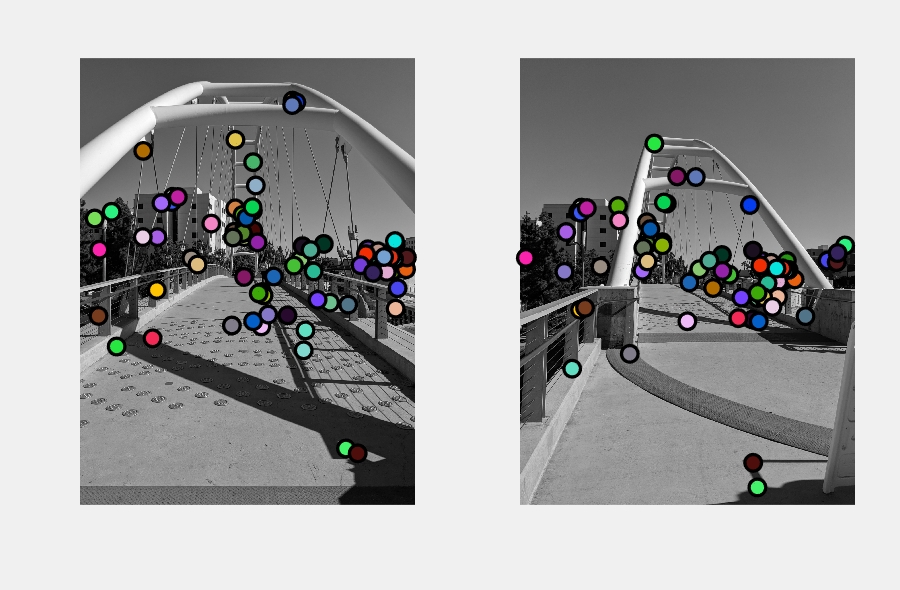
* Running the collect\_ground\_truth\_corr() file. The input of this file will be the above two images.
* After the images are displayed, we choose the corresponding points in both the images.
* We then click on the negative-ordinate above or left to stop it.
* This stops the script and saves it to mentioned file.
* We will be using this output file as ground-truth value while running the evaluate\_correspondence.m file to generate the evaluation image.

**Bridge Images:**

**Input:**

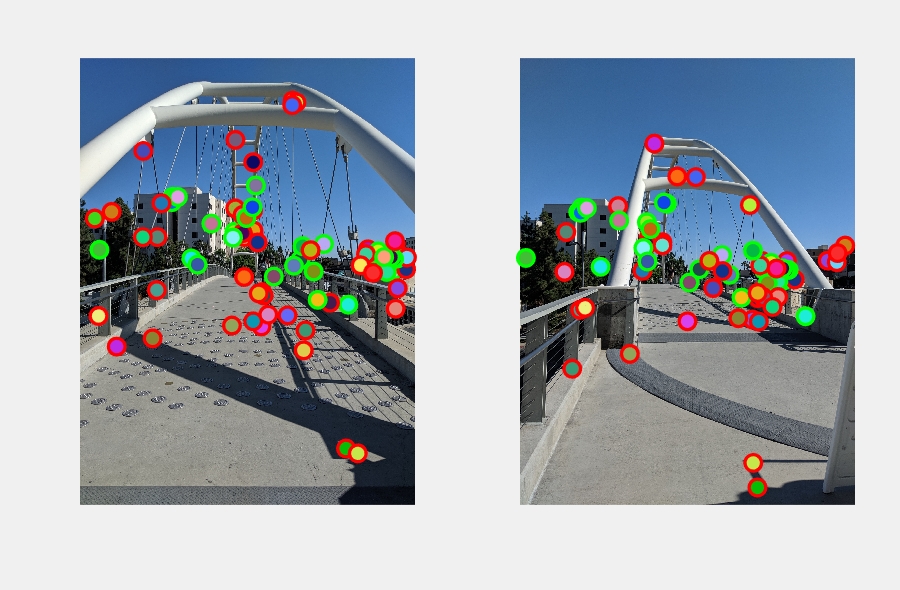
****

**Output:**

****

**(Ground-Truth)**

**(Evaluated:)**

****

**Results:**

**>> proj4**

**Saving visualization to vis.jpg**

**( 806, 1294) to ( 526, 1390) g.t. point 36 px. Match error 7 px. correct**

**( 838, 1252) to ( 556, 1350) g.t. point 54 px. Match error 3 px. correct**

**( 1006, 1808) to ( 728, 1884) g.t. point 3 px. Match error 2 px. correct**

**( 2778, 1888) to ( 2340, 1980) g.t. point 94 px. Match error 19 px. correct**

**( 2696, 1814) to ( 2270, 1914) g.t. point 42 px. Match error 1 px. correct**

**( 700, 1616) to ( 418, 1570) g.t. point 39 px. Match error 131 px. incorrect**

**( 1402, 1356) to ( 1146, 1482) g.t. point 2 px. Match error 74 px. incorrect**

**( 2744, 1890) to ( 2310, 1982) g.t. point 91 px. Match error 12 px. correct**

**( 2714, 1836) to ( 2286, 1936) g.t. point 42 px. Match error 3 px. correct**

**( 2202, 1678) to ( 1824, 1778) g.t. point 5 px. Match error 6 px. correct**

**( 830, 1300) to ( 548, 1396) g.t. point 12 px. Match error 8 px. correct**

**( 2824, 1920) to ( 2378, 2008) g.t. point 121 px. Match error 23 px. correct**

**( 2616, 1808) to ( 2200, 1908) g.t. point 120 px. Match error 9 px. correct**

**( 1992, 1818) to ( 1620, 1906) g.t. point 9 px. Match error 18 px. correct**

**( 884, 1254) to ( 604, 1354) g.t. point 8 px. Match error 3 px. correct**

**( 2636, 1776) to ( 2216, 1878) g.t. point 103 px. Match error 6 px. correct**

**( 1394, 1568) to ( 1146, 1482) g.t. point 1 px. Match error 11 px. correct**

**( 1568, 1588) to ( 1286, 1690) g.t. point 160 px. Match error 29 px. incorrect**

**( 1486, 1524) to ( 1164, 1618) g.t. point 79 px. Match error 21 px. correct**

**( 176, 1730) to ( 54, 1804) g.t. point 8 px. Match error 6 px. correct**

**( 2600, 1726) to ( 2186, 1832) g.t. point 155 px. Match error 13 px. incorrect**

**( 1468, 1416) to ( 886, 1336) g.t. point 40 px. Match error 333 px. incorrect**

**( 1920, 384) to ( 2076, 1326) g.t. point 263 px. Match error 779 px. incorrect**

**( 2944, 1906) to ( 2480, 1998) g.t. point 171 px. Match error 30 px. incorrect**

**( 2402, 3524) to ( 2144, 3876) g.t. point 1296 px. Match error 473 px. incorrect**

**( 2850, 1680) to ( 2410, 1832) g.t. point 45 px. Match error 368 px. incorrect**

**( 1566, 1556) to ( 1286, 1662) g.t. point 132 px. Match error 30 px. incorrect**

**( 2692, 1744) to ( 2266, 1848) g.t. point 71 px. Match error 4 px. correct**

**( 2868, 1808) to ( 2426, 1912) g.t. point 59 px. Match error 7 px. correct**

**( 1484, 1616) to ( 1164, 1618) g.t. point 97 px. Match error 98 px. incorrect**

**( 1954, 400) to ( 2076, 1326) g.t. point 262 px. Match error 749 px. incorrect**

**( 1564, 940) to ( 1300, 1304) g.t. point 140 px. Match error 145 px. incorrect**

**( 2868, 2078) to ( 2778, 1828) g.t. point 284 px. Match error 504 px. incorrect**

**( 2952, 1800) to ( 2860, 1842) g.t. point 91 px. Match error 64 px. incorrect**

**( 2422, 2226) to ( 2578, 2330) g.t. point 8 px. Match error 11 px. correct**

**( 2836, 1814) to ( 2398, 1916) g.t. point 30 px. Match error 10 px. correct**

**( 1932, 1874) to ( 1892, 1956) g.t. point 15 px. Match error 13 px. correct**

**( 1372, 2414) to ( 992, 2670) g.t. point 568 px. Match error 269 px. incorrect**

**( 332, 2604) to ( 1214, 772) g.t. point 531 px. Match error 2334 px. incorrect**

**( 2582, 1770) to ( 2170, 1872) g.t. point 157 px. Match error 13 px. incorrect**

**( 1874, 2322) to ( 2100, 2360) g.t. point 303 px. Match error 377 px. incorrect**

**( 1470, 1580) to ( 1148, 1684) g.t. point 77 px. Match error 195 px. incorrect**

**( 1186, 1494) to ( 896, 1466) g.t. point 19 px. Match error 12 px. correct**

**( 2010, 1698) to ( 2104, 1738) g.t. point 12 px. Match error 9 px. correct**

**( 694, 2094) to ( 544, 2272) g.t. point 121 px. Match error 122 px. incorrect**

**( 2770, 1930) to ( 2332, 2016) g.t. point 135 px. Match error 19 px. correct**

**( 134, 1448) to ( 2316, 1900) g.t. point 287 px. Match error 2336 px. incorrect**

**( 1588, 1150) to ( 1326, 1314) g.t. point 13 px. Match error 7 px. correct**

**( 2020, 2638) to ( 472, 2804) g.t. point 460 px. Match error 1416 px. incorrect**

**( 2808, 1796) to ( 2370, 1898) g.t. point 4 px. Match error 10 px. correct**

**( 1562, 1378) to ( 1286, 1690) g.t. point 30 px. Match error 365 px. incorrect**

**( 2534, 1870) to ( 2196, 2158) g.t. point 214 px. Match error 207 px. incorrect**

**( 2508, 3568) to ( 2106, 3654) g.t. point 1342 px. Match error 548 px. incorrect**

**( 286, 1388) to ( 2936, 1692) g.t. point 239 px. Match error 2945 px. incorrect**

**( 1644, 2112) to ( 2174, 2106) g.t. point 162 px. Match error 756 px. incorrect**

**( 2646, 1938) to ( 2868, 1760) g.t. point 165 px. Match error 705 px. incorrect**

**( 1658, 2142) to ( 2174, 2106) g.t. point 181 px. Match error 744 px. incorrect**

**( 1638, 2422) to ( 1512, 2376) g.t. point 453 px. Match error 129 px. incorrect**

**( 1482, 1976) to ( 1420, 1068) g.t. point 255 px. Match error 959 px. incorrect**

**( 1502, 1444) to ( 1178, 1544) g.t. point 8 px. Match error 18 px. correct**

**( 656, 2530) to ( 1972, 2348) g.t. point 516 px. Match error 1396 px. incorrect**

**( 1558, 1346) to ( 1300, 1304) g.t. point 16 px. Match error 11 px. correct**

**( 1700, 2316) to ( 402, 1930) g.t. point 339 px. Match error 1154 px. incorrect**

**( 2746, 1796) to ( 2316, 1900) g.t. point 11 px. Match error 6 px. correct**

**( 2258, 2208) to ( 2146, 2166) g.t. point 109 px. Match error 37 px. incorrect**

**( 736, 1310) to ( 1106, 1920) g.t. point 103 px. Match error 831 px. incorrect**

**( 168, 2328) to ( 588, 2254) g.t. point 227 px. Match error 614 px. incorrect**

**( 572, 838) to ( 1744, 2076) g.t. point 486 px. Match error 1841 px. incorrect**

**( 1604, 1664) to ( 1828, 1922) g.t. point 218 px. Match error 521 px. incorrect**

**( 2110, 1926) to ( 2236, 2032) g.t. point 3 px. Match error 9 px. correct**

**( 1564, 2376) to ( 2156, 2378) g.t. point 433 px. Match error 817 px. incorrect**

**( 1382, 1620) to ( 1114, 1714) g.t. point 12 px. Match error 10 px. correct**

**( 2850, 2262) to ( 2330, 2156) g.t. point 422 px. Match error 698 px. incorrect**

**( 1062, 1864) to ( 1152, 1836) g.t. point 6 px. Match error 18 px. correct**

**( 2036, 2458) to ( 472, 2804) g.t. point 285 px. Match error 1467 px. incorrect**

**( 2082, 1734) to ( 1710, 1830) g.t. point 74 px. Match error 474 px. incorrect**

**( 1404, 738) to ( 2196, 2158) g.t. point 212 px. Match error 1578 px. incorrect**

**( 2146, 2182) to ( 2000, 2162) g.t. point 15 px. Match error 6 px. correct**

**( 1750, 1972) to ( 1530, 2026) g.t. point 15 px. Match error 18 px. correct**

**( 1614, 2124) to ( 2146, 2120) g.t. point 190 px. Match error 758 px. incorrect**

**( 568, 1616) to ( 2280, 2246) g.t. point 143 px. Match error 2053 px. incorrect**

**( 2844, 1652) to ( 2316, 1900) g.t. point 45 px. Match error 481 px. incorrect**

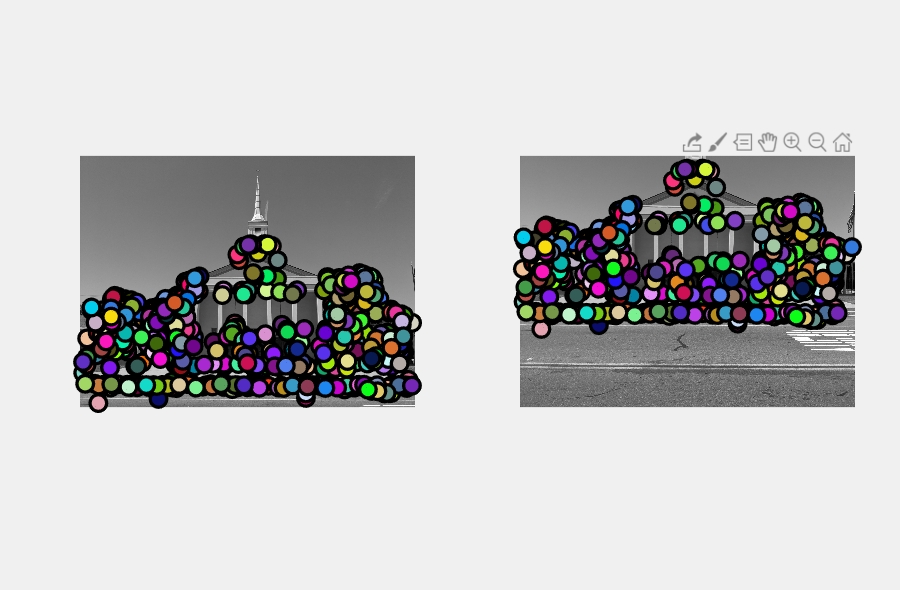
**( 1916, 424) to ( 1588, 1074) g.t. point 224 px. Match error 406 px. incorrect**

**35 total good matches, 48 total bad matches**

**Saving visualization to eval.jpg**

**Colored Building:**

**Input:**

**Output:**

**(Ground-Truth:)**

**(Evaluation:)**

**Results:**

**>> proj4**

**Saving visualization to vis.jpg**

**( 2244, 2514) to ( 2188, 1666) g.t. point 9 px. Match error 3819 px. incorrect**

**( 930, 2848) to ( 932, 1968) g.t. point 429 px. Match error 32 px. incorrect**

**( 1952, 2148) to ( 1908, 1324) g.t. point 64 px. Match error 6 px. correct**

**( 1732, 2534) to ( 1698, 1684) g.t. point 127 px. Match error 26 px. incorrect**

**( 1148, 2510) to ( 1128, 1660) g.t. point 150 px. Match error 28 px. incorrect**

**( 1916, 1256) to ( 1864, 374) g.t. point 51 px. Match error 25 px. correct**

**( 1144, 2478) to ( 1122, 1630) g.t. point 155 px. Match error 26 px. incorrect**

**( 3526, 1622) to ( 3510, 792) g.t. point 230 px. Match error 67 px. incorrect**

**( 1356, 2270) to ( 1320, 1434) g.t. point 88 px. Match error 12 px. correct**

**( 3746, 1850) to ( 3702, 1028) g.t. point 45 px. Match error 12 px. correct**

**( 1478, 2778) to ( 1458, 1904) g.t. point 237 px. Match error 20 px. incorrect**

**( 1896, 2788) to ( 1856, 1912) g.t. point 191 px. Match error 2 px. incorrect**

**( 1896, 1188) to ( 1842, 298) g.t. point 29 px. Match error 19 px. correct**

**( 3488, 2816) to ( 3346, 1938) g.t. point 310 px. Match error 38 px. incorrect**

**( 934, 1818) to ( 864, 978) g.t. point 360 px. Match error 20 px. incorrect**

**( 3068, 1714) to ( 3040, 886) g.t. point 267 px. Match error 34 px. incorrect**

**( 656, 2762) to ( 660, 1892) g.t. point 245 px. Match error 22 px. incorrect**

**( 1270, 1516) to ( 1186, 660) g.t. point 15 px. Match error 9 px. correct**

**( 628, 1918) to ( 560, 1080) g.t. point 204 px. Match error 19 px. incorrect**

**( 2142, 1168) to ( 2106, 276) g.t. point 76 px. Match error 4 px. correct**

**( 1074, 1822) to ( 1008, 984) g.t. point 359 px. Match error 4 px. incorrect**

**( 208, 2024) to ( 138, 1188) g.t. point 98 px. Match error 14 px. correct**

**( 188, 2008) to ( 114, 1168) g.t. point 98 px. Match error 17 px. correct**

**( 2700, 2084) to ( 2646, 1262) g.t. point 8 px. Match error 7 px. correct**

**( 1382, 1472) to ( 1300, 610) g.t. point 42 px. Match error 31 px. incorrect**

**( 376, 1838) to ( 3996, 1100) g.t. point 101 px. Match error 3716 px. incorrect**

**( 740, 2184) to ( 3742, 1170) g.t. point 85 px. Match error 3058 px. incorrect**

**( 798, 2760) to ( 802, 1890) g.t. point 281 px. Match error 22 px. incorrect**

**( 1022, 1856) to ( 954, 1018) g.t. point 395 px. Match error 19 px. incorrect**

**( 3098, 1908) to ( 3054, 1084) g.t. point 310 px. Match error 20 px. incorrect**

**( 3264, 1512) to ( 3256, 672) g.t. point 168 px. Match error 55 px. incorrect**

**( 1140, 1768) to ( 1072, 926) g.t. point 275 px. Match error 8 px. incorrect**

**( 3510, 2440) to ( 3402, 1598) g.t. point 147 px. Match error 12 px. correct**

**( 376, 1936) to ( 304, 1098) g.t. point 198 px. Match error 23 px. incorrect**

**( 38, 2482) to ( 2980, 1458) g.t. point 67 px. Match error 2953 px. incorrect**

**( 140, 1826) to ( 44, 982) g.t. point 192 px. Match error 4 px. incorrect**

**155 total good matches, 384 total bad matches**

**Saving visualization to eval.jpg**

**Observations:**

For the Notre Dame image pair, for case 1: 37 matches were found in total, of which 25 were correct, hinting at an accuracy of about 67.5% for this one image pair with the matching NNDR ratio threshold at 0.75 and for case 2, 41 matches were found in total, of which 28 were corrected with accuracy of about 68%.

For the Bridge Images, I have made the ground-truth value and generate the viz.jpg and eval.jpg. There were 35 correct matches found from 83 matches which gave the result of 42% accuracy.

Moreover, for the Colored Building Images, the results were about 115 correct matches out of 499 matches with the accuracy of about 23%.

For the images, the algorithm can have good or bad performance, image pair with large scale are harder to match by the algorithm, In contrast, the image pairs hat are similar can be matched easily.