

# Homography Estimation: RANSAC

When you take a panoramic image with your camera, you end up with a sequence of images of the same scene taken from different perspectives. Each image has a slightly difference appearance but in many cases it is possible to transform them so they can be combined into a single image or panorama. In this Lab you will program automatic homography estimation between two images. Although this lab is not dependent on labs from previous weeks, it builds on concepts from Week 4.

In this lab you will (1) compute the homography between two images when correspondences are known, (2) estimate correspondences between two sets of Harris Corners using Nearest Neighbor and the ratio test and (3) use RANSAC to find the best homography estimate from the estimated correspondences from (2).

## Your Script

 Save  Reset  MATLAB Documentation (<https://www.mathworks.com/help/>)

```
1 buildingDir = fullfile(toolboxdir('vision'), 'visiondata', 'building');
2 buildingScene = imageDatastore(buildingDir);
3
4 I1 = readimage(buildingScene, 1);
5 I2 = readimage(buildingScene, 2);
6
7 I1_gray = rgb2gray(I1);
8 I2_gray = rgb2gray(I2);
9
10 % get points
11 points1 = detectHarrisFeatures(I1_gray);
12 points2 = detectHarrisFeatures(I2_gray);
13
14 % get features
15 [features1, points1] = extractFeatures(I1_gray, points1);
16 [features2, points2] = extractFeatures(I2_gray, points2);
17
18 loc1 = points1.Location;
19 loc2 = points2.Location;
20
21 [match,match_fwd,match_bkwd] = match_features(double(features1.Features),double(features2.Features));
22
23 H = ransac_homography(loc1(match(:,1),:),loc2(match(:,2),:));
24
25 I = stitch(I1,I2,H);
26
27 figure()
28 imshow(I)
29
30 function best_H = ransac_homography(p1,p2)
31     thresh = sqrt(2); % threshold for inlier points
32     p = 1-1e-4; % probability of RANSAC success
33     w = 0.5; % fraction inliers
34
35     % n: number of correspondences required to build the model (homography)
36
37     n = 4;
38     % number of iterations required
39     % from the lecture given the probability of RANSAC success, and fraction of inliers
40     k = log(1-p)/log(1-w^n)+1; % might want to erase the '+1'
41
42     num_pts = size(p1,1);
43     best_inliers = 4;
44     best_H = eye(3);
45     for iter = 1:k
46         % randomly select n correspondences from p1 and p2
47         % use these points to compute the homography
48         N = length(p1);
49         p_indices = randperm(N,n);
50         p1_sample = p1(p_indices,:);
```

```

50 % get p2_sample from correspondence:
51 % p1 and p2 are listed in order according to correspondence
52
53
54 p2_sample = p2(p_indices,:);
55
56 H = compute_homography(p1_sample,p2_sample);
57
58 % transform p2 to homogeneous coordinates
59 p2_h = [p2 ones(N,1)];
60
61 % estimate the location of correspondences given the homography
62 p1_hat = (H*p2_h(:, :)')';
63
64 % convert to image coordinates by dividing x and y by the third coordinate
65 p1_hat = p1_hat./p1_hat(:,3);
66 p1_hat(:,3)=[];
67
68 % compute the distance between the estimated correspondence location and the
69 % putative correspondence location
70 %dist = [];
71 %for j = 1:length(p1)
72 % dist = [dist; pdist2(p1(j,:),p1_hat(j,:))];
73 %end
74 dist = diag(pdist2(p1(:, :),p1_hat(:, :)));
75
76 % inlying points have a distance less than the threshold thresh defined previously
77 num_inliers = sum(dist<thresh);
78
79 if num_inliers > best_inliers
80     best_inliers = num_inliers;
81     best_H = H;
82 end
83 end
84 end
85
86 function H = compute_homography(p1,p2)
87 % use SVD to solve for H as was done in the lecture
88
89 % note pj is an 8 vector representing 4 points
90 % based on http://ros-developer.com/2017/12/26/
91 % finding-homography-matrix-using-singular-value-decomposition-and-ransac-in-opencv-and-matlab/
92
93 x1 = p2(1,1);
94 y1 = p2(1,2);
95 x2 = p2(2,1);
96 y2 = p2(2,2);
97 x3 = p2(3,1);
98 y3 = p2(3,2);
99 x4 = p2(4,1);
100 y4 = p2(4,2);
101
102 xp1 = p1(1,1);
103 yp1 = p1(1,2);
104 xp2 = p1(2,1);
105 yp2 = p1(2,2);
106 xp3 = p1(3,1);
107 yp3 = p1(3,2);
108 xp4 = p1(4,1);
109 yp4 = p1(4,2);
110
111 % Construct A Matrix Based On Correspondences
112
113 A=[
114 -x1 -y1 -1 0 0 0 x1*xp1 y1*xp1 xp1;
115 0 0 0 -x1 -y1 -1 x1*yp1 y1*yp1 yp1;
116 -x2 -y2 -1 0 0 0 x2*xp2 y2*xp2 xp2;
117 0 0 0 -x2 -y2 -1 x2*yp2 y2*yp2 yp2;
118 -x3 -y3 -1 0 0 0 x3*xp3 y3*xp3 xp3;

```

```

118     0  0  0  -x3  -y3  -1  x3*yp3  y3*yp3  yp3;
120    -x4  -y4  -1  0  0  0  x4*xp4  y4*xp4  xp4;
121     0  0  0  -x4  -y4  -1  x4*yp4  y4*yp4  yp4];
122
123     [U,S,V] = svd(A);
124
125
126     X=V(:,end)/V(end,end);
127     H=reshape(X,3,3)';
128
129 end
130
131 function [match,match_fwd,match_bkwd] = match_features(f1,f2)
132     %% INPUT
133     %% f1,f2: [ number of points x number of features ]
134     %% OUTPUT
135     %% match, match_fwd, match_bkwd: [ indices in f1, corresponding indices in f2 ]
136
137     % get matches using pdist and the ratio test with threshold of 0.7
138     % fwd matching
139     match_fwd = [];
140
141     % get indices of f2 which are closest to f1
142     neighbors_wrt_second = knnsearch(f2,f1,'K',2);
143
144     number = length(neighbors_wrt_second);
145     for ind = 1:number
146         feature_1 = f1(ind,:);
147         current_pair = neighbors_wrt_second(ind,:);
148         d_1 = pdist2(feature_1, f2(current_pair(1),:));
149         d_2 = pdist2(feature_1, f2(current_pair(2),:));
150         d_ratio = d_1/d_2;
151         if d_ratio < .7
152             match_fwd = [match_fwd ; ind, current_pair(1)];
153         end
154     end
155
156     % bkwd matching
157     match_bkwd = [];
158
159     % get indices of f1 which are closest to f2
160     neighbors_wrt_first = knnsearch(f1,f2,'K',2);
161
162     number = length(neighbors_wrt_first);
163     for ind = 1:number
164         feature_2 = f2(ind,:);
165         current_pair = neighbors_wrt_first(ind,:);
166         d_1 = pdist2(f1(current_pair(1),:), feature_2);
167         d_2 = pdist2(f1(current_pair(2),:), feature_2);
168         d_ratio = d_1/d_2;
169         if d_ratio < .7
170             match_bkwd = [match_bkwd ; current_pair(1), ind];
171         end
172     end
173
174
175
176     % fwd bkwd consistency check
177     match = intersect(match_fwd, match_bkwd, 'rows');
178 end

```

▶ Run Script



Previous Assessment: All Tests Passed

✓ Is the Homography estimate correct?

## Output

