Automatic computation of F

Objective Compute the fundamental matrix between two images.

Algorithm

(i) **Interest points:** Compute interest points in each image.

(ii) **Putative correspondences:** Compute a set of interest point matches based on proximity and similarity of their intensity neighbourhood.

(iii) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5(p121):

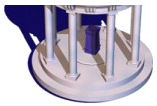
- (a) Select a random sample of 7 correspondences and compute the fundamental matrix F as described in section 11.1.2. There will be one or three real solutions.
- (b) Calculate the distance $d_\perp$ for each putative correspondence.
- (c) Compute the number of inliers consistent with F by the number of correspondences for which $d_\perp < t$ pixels.
- (d) If there are three real solutions for F the number of inliers is computed for each solution, and the solution with most inliers retained.

Choose the F with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.

(iv) **Non-linear estimation:** re-estimate F from all correspondences classified as inliers by minimizing a cost function, e.g. (11.6), using the Levenberg–Marquardt algorithm of section A6.2(p600).

(v) **Guided matching:** Further interest point correspondences are now determined using the estimated F to define a search strip about the epipolar line.

The last two steps can be iterated until the number of correspondences is stable.

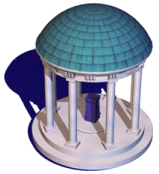


Feature points

- Extract feature points to relate images
- Required properties:
  - Well-defined

    (i.e. neigboring points should all be different)
  - Stable across views

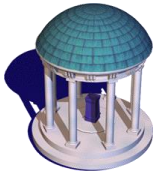
    (i.e. same 3D point should be extracted as feature for neighboring viewpoints)



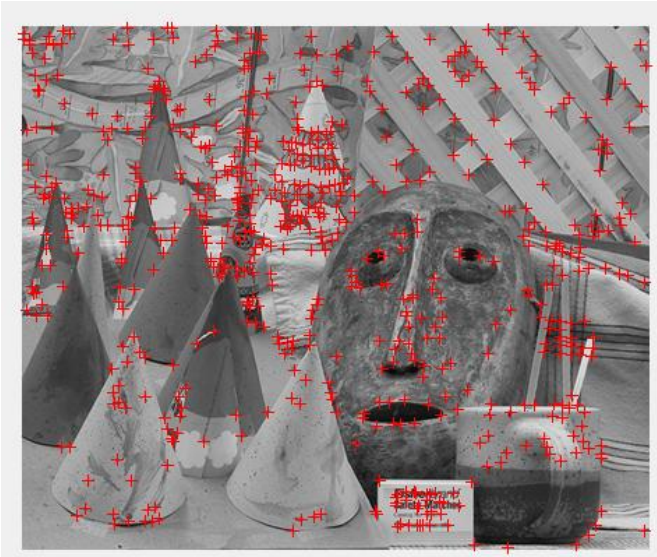
Harris Feature points

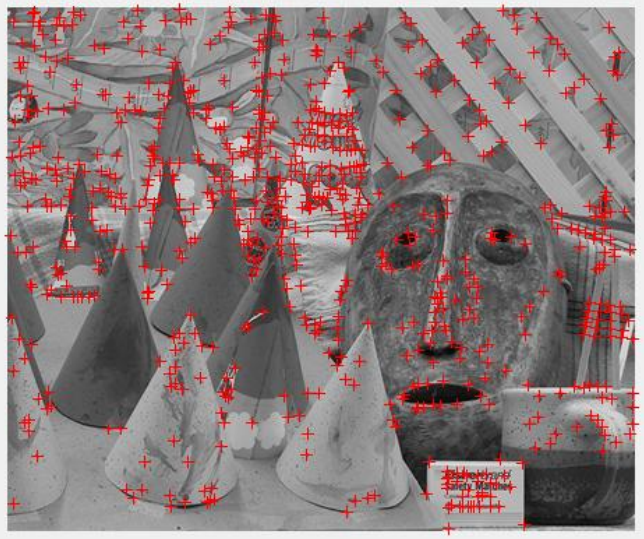
(e.g.Harris&Stephens´88; Shi&Tomasi´94)

Step 1. Find points that differ as much as possible from all neighboring points



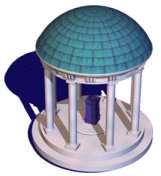
Feature points





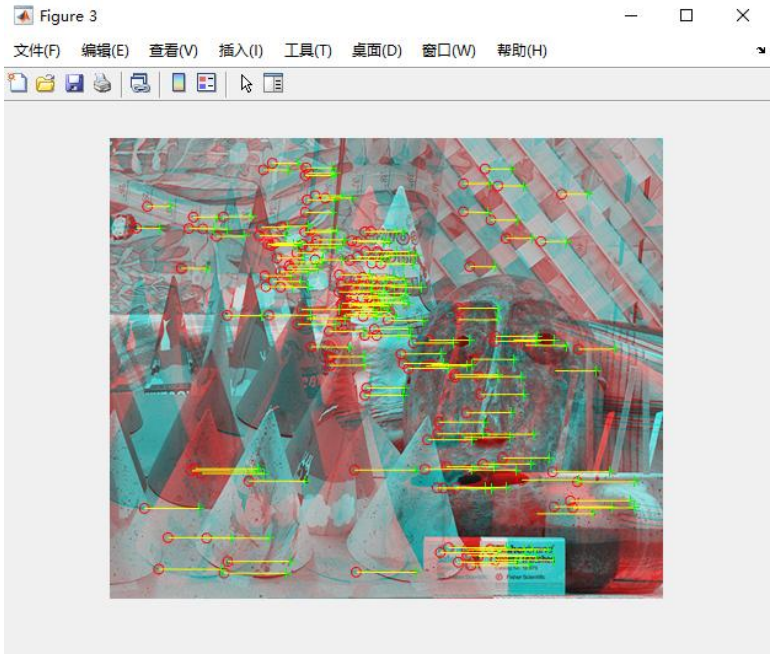
```
I1 = rgb2gray(imread("C:\Users\admin\Desktop\imR.png"));
I2 = rgb2gray(imread("C:\Users\admin\Desktop\imL.png"));
points1 =detectHarrisFeatures(I1);
points2 =detectHarrisFeatures(I2);

figure
imshow(I1);
hold on
plot(points1.Location(:,1),points1.Location(:,2),'r+');
figure
imshow(I2);
hold on
plot(points2.Location(:,1),points2.Location(:,2),'r+');
```

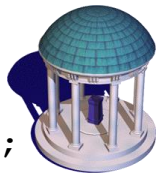


Feature matching

Step 2. Match the feature points




```
[features1,valid_points1] = extractFeatures(I1,points1);
[features2,valid_points2] = extractFeatures(I2,points2);
indexPairs = matchFeatures(features1,features2);
matchedPoints1 = valid_points1(indexPairs(:,1),:);
matchedPoints2 = valid_points2(indexPairs(:,2),:);
figure;
showMatchedFeatures(I1,I2,matchedPoints1,matchedPoints2);
```



RANSAC

Step 3. RANSAC: Purpose of RANSAC is to find out and discard the outliers

Do

select minimal sample (e.g. 7 matches)

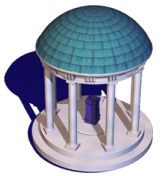
compute solution(s) for F determine inliers

until Γ(*#inliers*,*#samples*)>95%

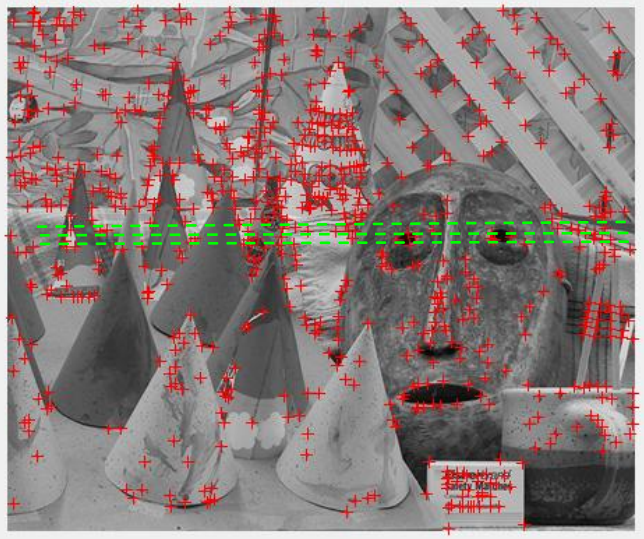
$$\Gamma = 1 - \left(1 - \left(\frac{\#inliers}{\#matches}\right)^7\right)^{\#samples}$$

#inliers	90%	80%	70%	60%	50%
#samples	5	13	35	106	382

Step 4. Compute F based on all inliers

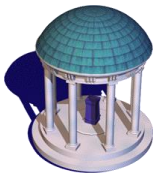


Finding more matches



Step 5. Find more mathes: restrict search range to neighborhood of epipolar line (e.g. ±1.5 pixels) and find more matches, using all matches calculate F.

Repeat step 5 till the number of matches become stable.



Degenerate cases:

- Planar scene
- Pure rotation