SIMON FRASER UNIVERSITY

8-Point Algorithm + RANSAC

Arezou Fatemi, Kangxue Yin, Ali Mahdavi Amiri

October 13, 2020

1 NORMALIZED 8-POINT ALGORITHM (10 POINTS)

Implement the function “FM_by_normalized_8_point” in “FM.py”. You need to compute the Fundamental Matrix using the 8-point algorithm. To verify your implementation, you can compare your result with the following opencv function:

F, _= cv2.findFundamentalMat(pts1, pts2, cv2.FM_8POINT)

Here’s the general idea for normalizing the input points:

1. Find the centroid of the points (find the mean x and mean y value)
2. Compute the mean distance of all the points from this centroid
3. Construct a 3 by 3 matrix that would translate the points so that the mean distance would be sqrt(2)

(Let's say (x,y) is the centroid and m is the mean distance from centroid. This would be the matrix:

[[sqrt(2)/m, 0, -x(sqrt(2)/m)],

[0, sqrt(2)/m, -y(sqrt(2)/m)],

[0, 0, 1]]

Now you can use this matrix to normalize (and later de-normalize) the points.

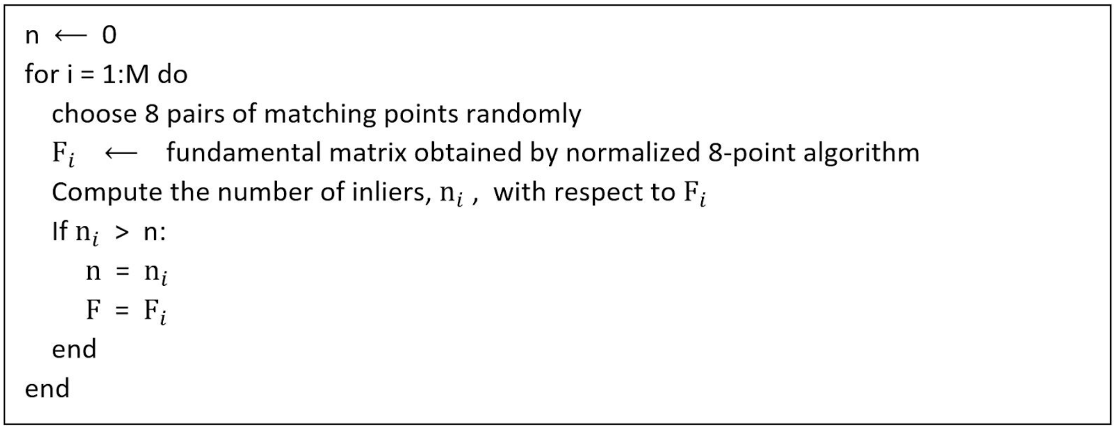
You can find more information about this in the wikipedia page.

- normalize the input points	+ 2 pts
- construct the coefficient matrix of the linear system correctly	+ 2 pts

- solve the linear least square problem correctly	+ 2 pts
- get correct results	+ 2 pts
- comments that explain in details how your code works	+ 2 pts

2 RANSAC (10 POINTS)

Implement the function “FM_by_RANSAC” in “FM.py”. You need to compute the Fundamental Matrix using RANSAC. Here is the pseudo code:



```
n ⟵ 0
for i = 1:M do
  choose 8 pairs of matching points randomly
  $F_i$ ⟵ fundamental matrix obtained by normalized 8-point algorithm
  Compute the number of inliers, $n_i$, with respect to $F_i$
  If $n_i$ > n:
     $n = n_i$
     $F = F_i$
  end
end
```

To verify your implementation, you can compare your result with the following opencv function:

F, mask = cv2.findFundamentalMat(pts1,pts2, cv2.FM_RANSAC)

- compute number of inliers correctly	+ 3 pts
- get correct results	+ 5 pts
- comments that explain in details how your code works	+ 2 pts