CMPT 732

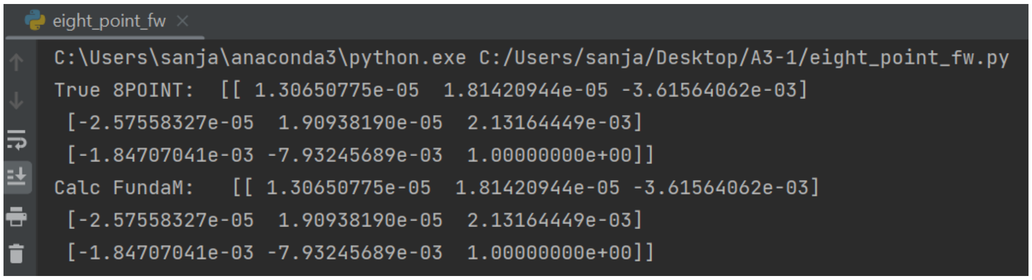
Practices in Visual Computing 1, Fall 2022

ASSIGNMENT 3

PART 1: EPIPOLAR GEOMETRY

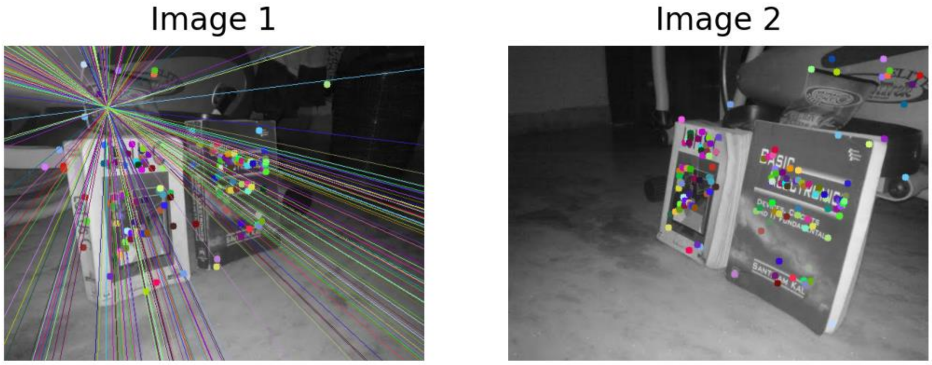
SANJANA MISHRA

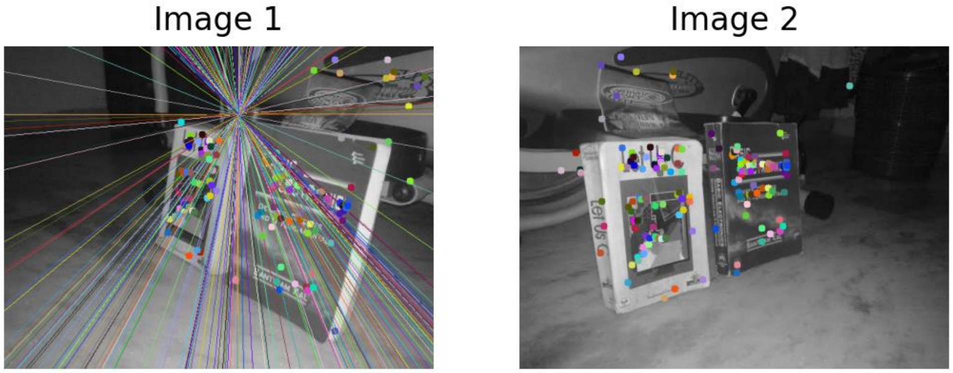
[1] For image pair: Myleft and Myright



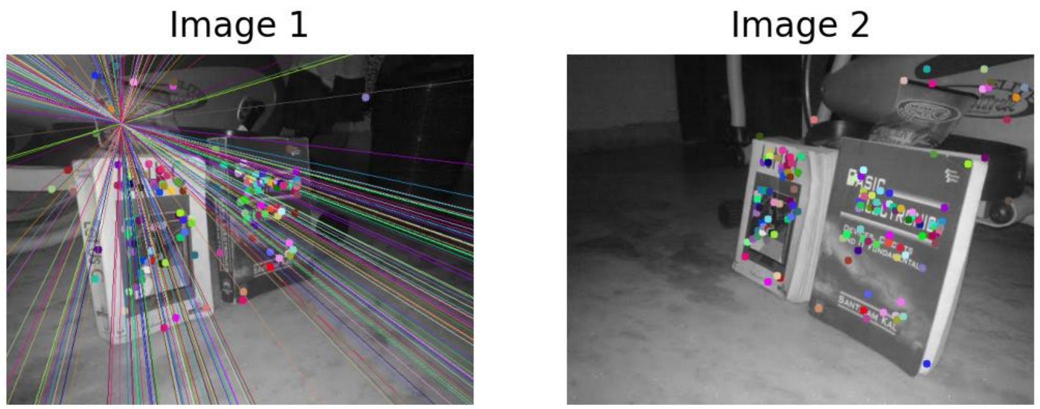
```
C:\Users\sanja\anaconda3\python.exe C:/Users/sanja/Desktop/A3-1/eight_point_fw.py
True 8POINT:  [[ 1.30650775e-05  1.81420944e-05 -3.61564062e-03]
 [-2.57558327e-05  1.90938190e-05  2.13164449e-03]
 [-1.84707041e-03 -7.93245689e-03  1.00000000e+00]]
Calc FundaM:   [[ 1.30650775e-05  1.81420944e-05 -3.61564062e-03]
 [-2.57558327e-05  1.90938190e-05  2.13164449e-03]
 [-1.84707041e-03 -7.93245689e-03  1.00000000e+00]]
```

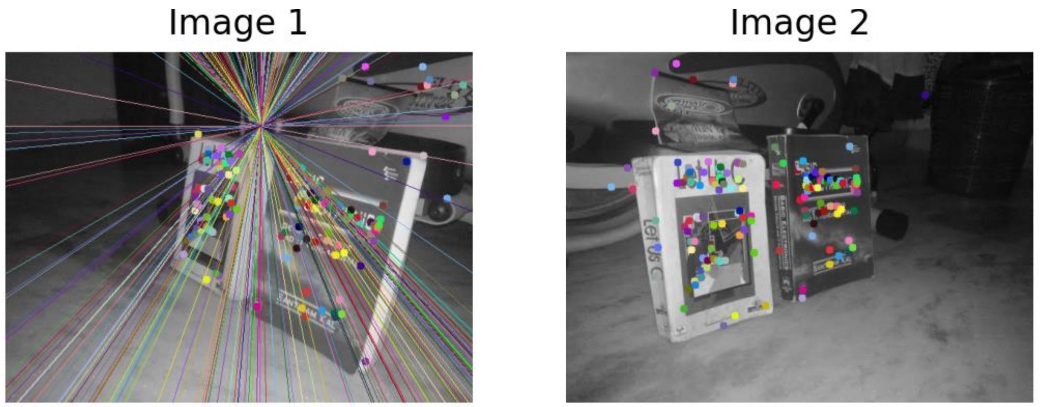
Identical true and calculated fundamental matrices are shown above.
True fundamental matrix using opencv:



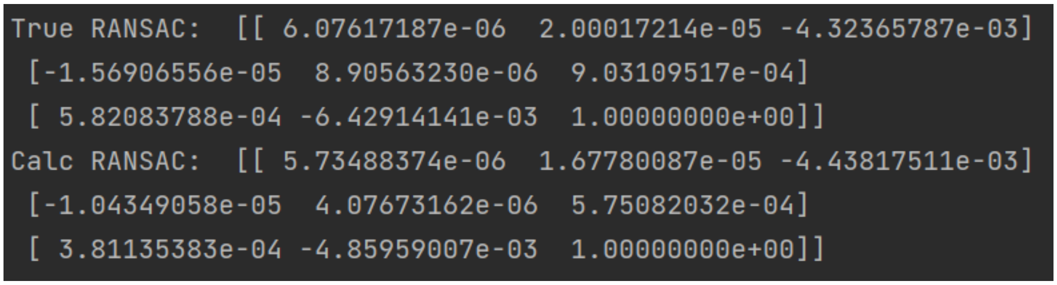


Calculated Fundamental Matrix:



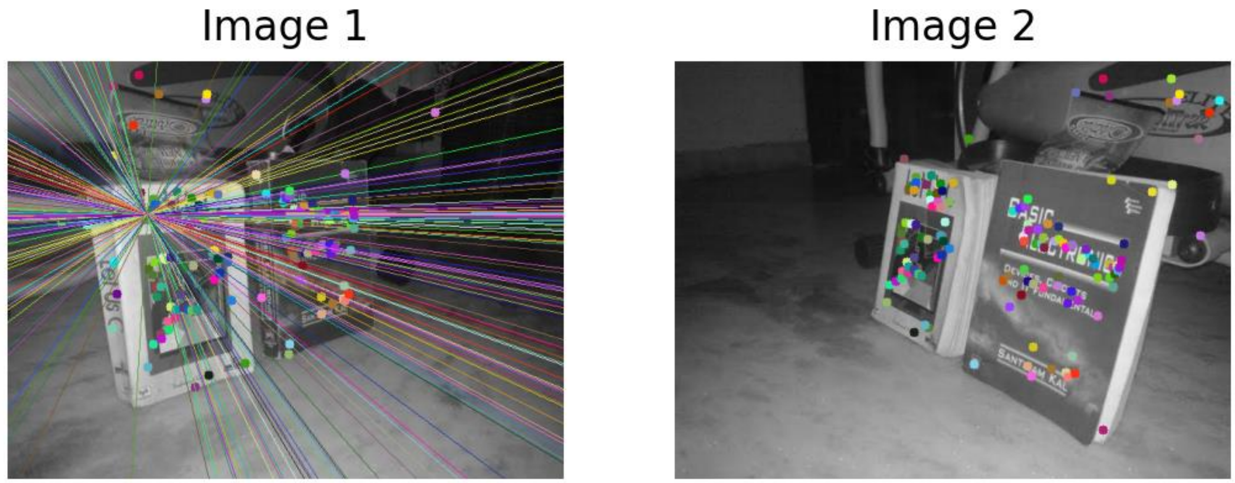


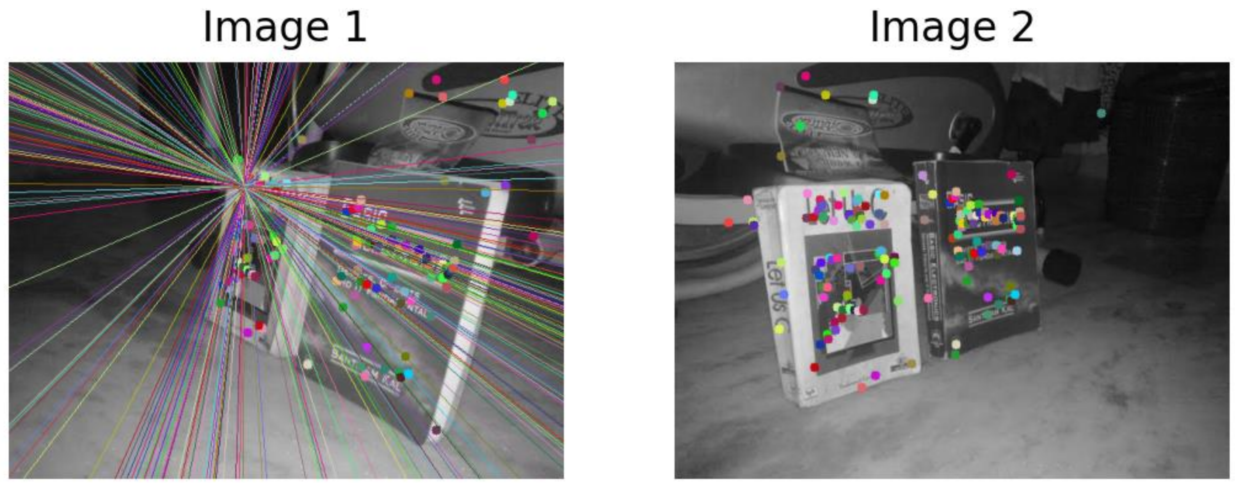
USING RANSAC:



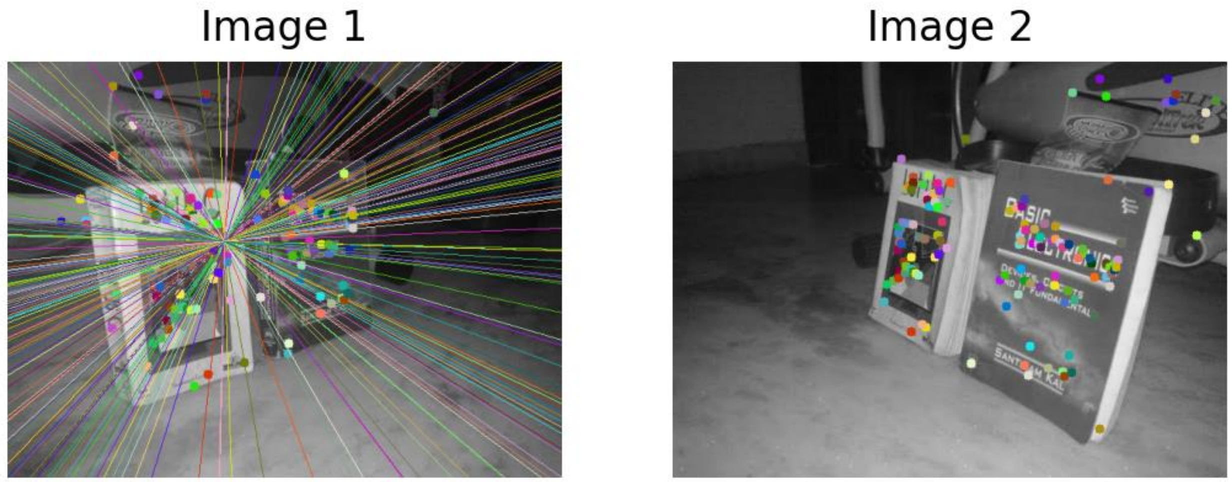
```
True RANSAC:  [[ 6.07617187e-06  2.00017214e-05 -4.32365787e-03]
 [-1.56906556e-05  8.90563230e-06  9.03109517e-04]
 [ 5.82083788e-04 -6.42914141e-03  1.00000000e+00]]
Calc RANSAC:  [[ 5.73488374e-06  1.67780087e-05 -4.43817511e-03]
 [-1.04349058e-05  4.07673162e-06  5.75082032e-04]
 [ 3.81135383e-04 -4.85959007e-03  1.00000000e+00]]
```

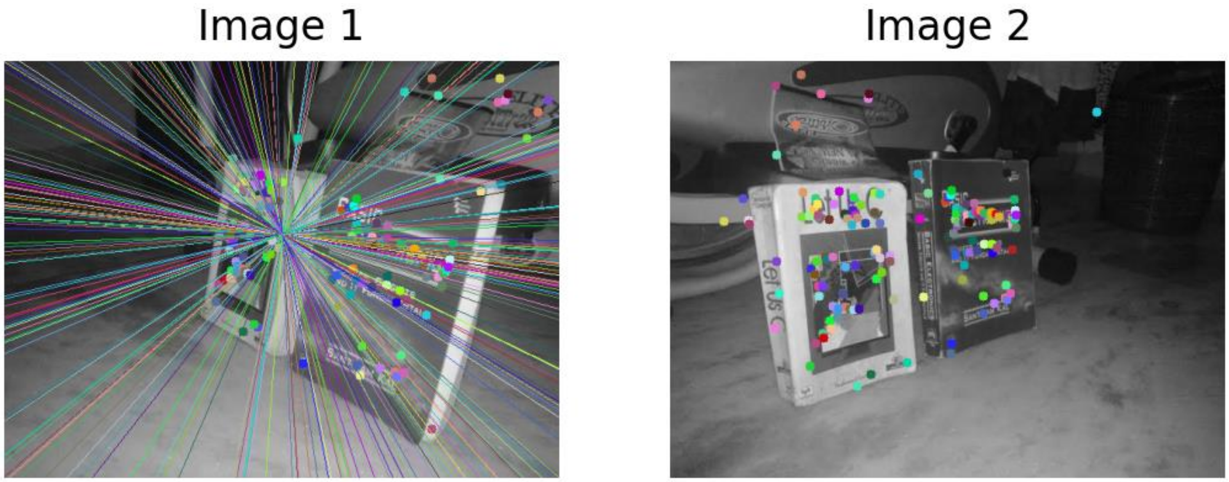
True fundamental matrix with RANSAC using opencv:





Calculated:





The above images show the outputs in the following order:

1. **True fundamental matrix as per OpenCV and calculated matrix**
2. **True fundamental matrix as per OpenCV image pair**
3. **Calculated matrix image pair**
4. **True fundamental matrix with RANSAC as per OpenCV image pair**
5. **Calculated RANSAC matrix image pair**

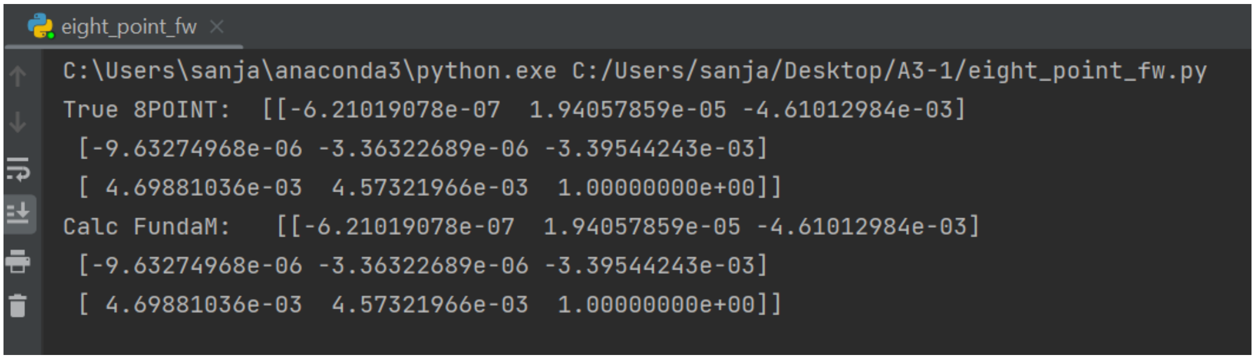
Here, (1) for all three image pairs in the dataset indicate that the two matrices are identical.

In each (2), (3), (4), (5) the first set of images (image 1 and image 2-top ones) show the epipoles in one view of the image and the second set of images(image 1 and image 2-bottom ones) show the same in the other view of the image. What we are checking for is if the key points in both images are found correctly. We are finding the fundamental matrix.

Now, there are some unmatched cases as well. For this, RANSAC is used to improve the final result by keeping a track of the inliers. (4) and (5) in all the image sets use RANSAC with the same fundamental matrix idea. The matrix with the highest number of corresponding inliers is chosen as the best matrix.

This above is applicable for all three image pairs.

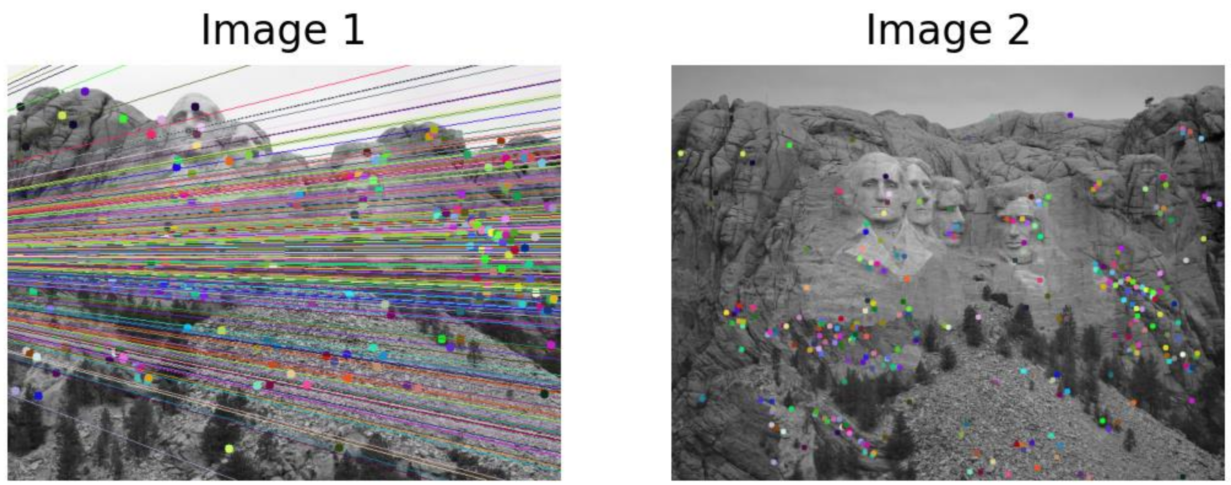
[2] For image pair: mount_rushmore_1 and mount_rushmore_2

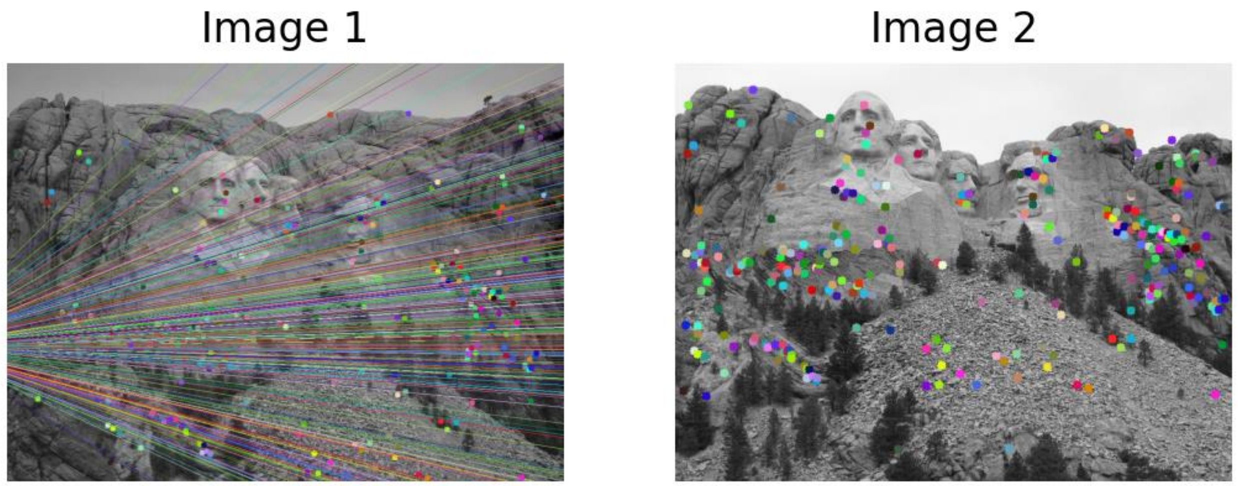


```
C:\Users\sanja\anaconda3\python.exe C:/Users/sanja/Desktop/A3-1/eight_point_fw.py
True 8POINT:  [[-6.21019078e-07  1.94057859e-05 -4.61012984e-03]
 [-9.63274968e-06 -3.36322689e-06 -3.39544243e-03]
 [ 4.69881036e-03  4.57321966e-03  1.00000000e+00]]
Calc FundaM:   [[-6.21019078e-07  1.94057859e-05 -4.61012984e-03]
 [-9.63274968e-06 -3.36322689e-06 -3.39544243e-03]
 [ 4.69881036e-03  4.57321966e-03  1.00000000e+00]]
```

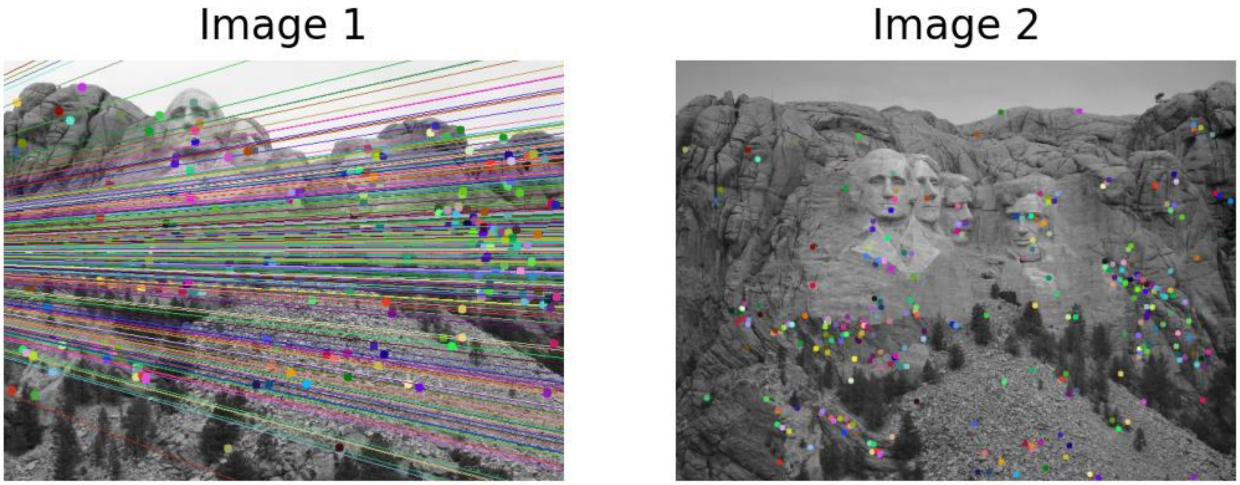
Identical true and calculated fundamental matrices are shown above.

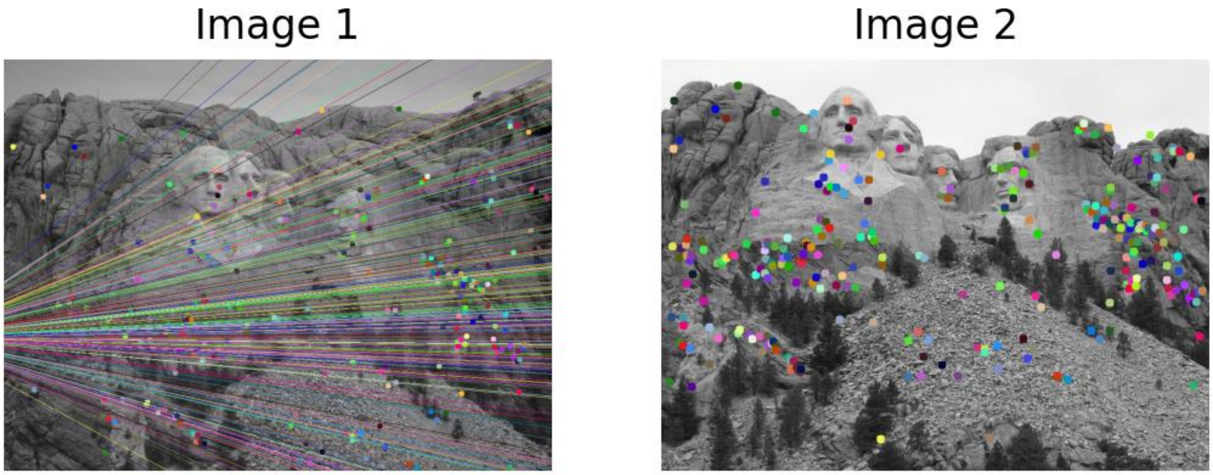
True fundamental matrix using opencv:



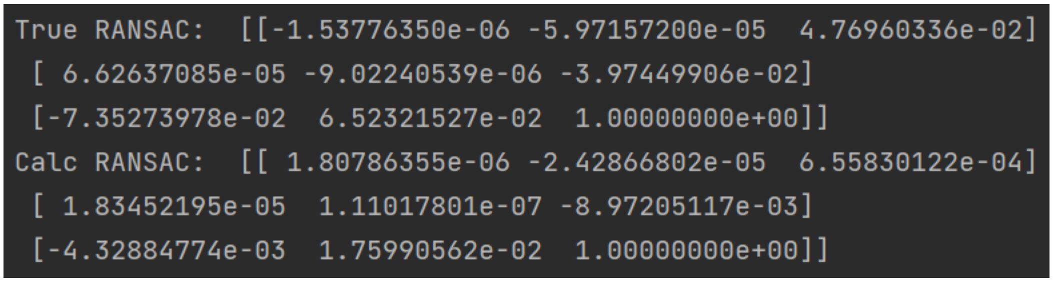


Calculated Fundamental Matrix:





USING RANSAC:



```
True RANSAC:  [[-1.53776350e-06 -5.97157200e-05  4.76960336e-02]
 [ 6.62637085e-05 -9.02240539e-06 -3.97449906e-02]
 [-7.35273978e-02  6.52321527e-02  1.00000000e+00]]
Calc RANSAC:  [[ 1.80786355e-06 -2.42866802e-05  6.55830122e-04]
 [ 1.83452195e-05  1.11017801e-07 -8.97205117e-03]
 [-4.32884774e-03  1.75990562e-02  1.00000000e+00]]
```

True fundamental matrix with RANSAC using opencv:

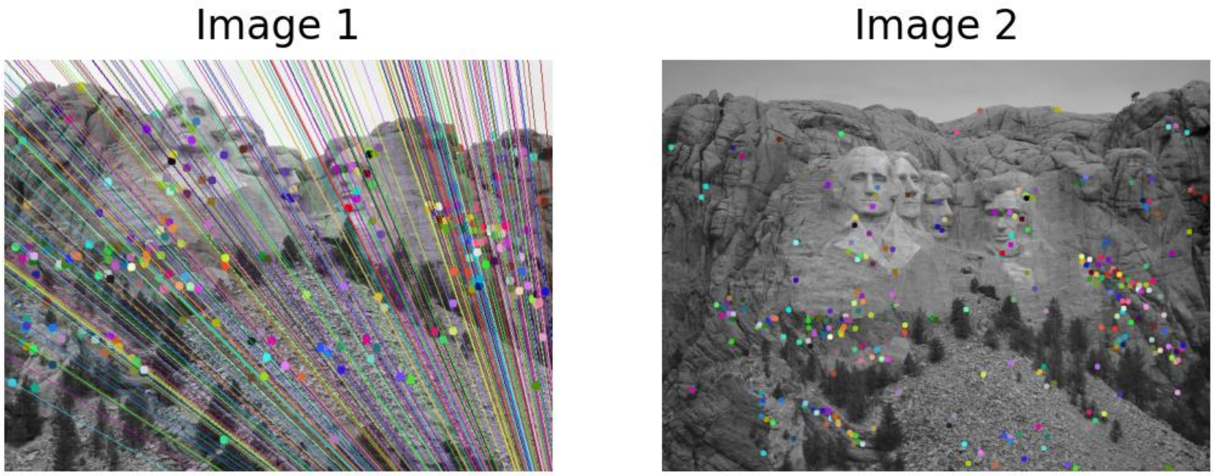


Image 1

Image 2

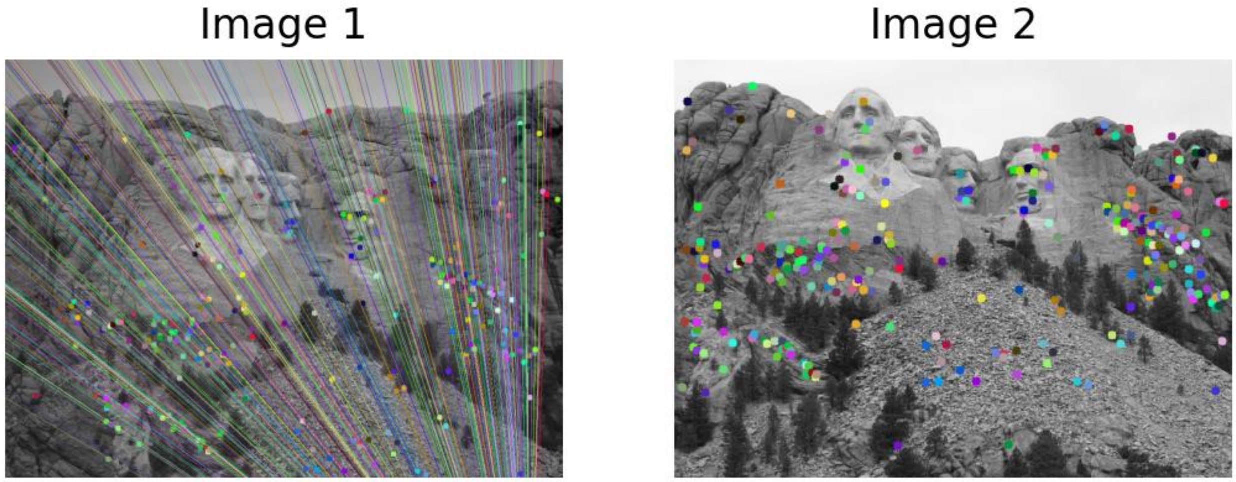


Image 1

Image 2

Calculated:

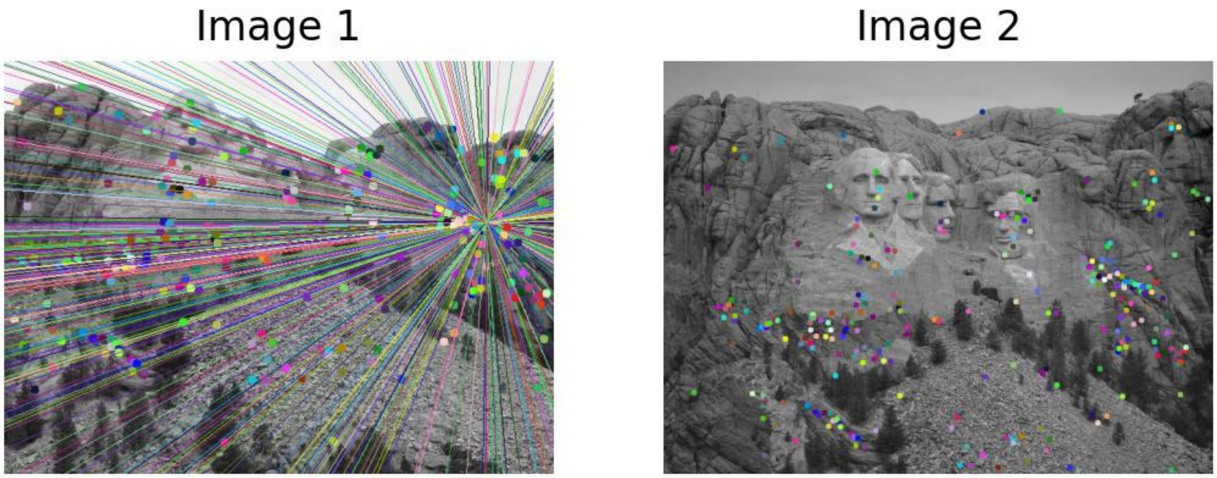


Image 1

Image 2

Image 1

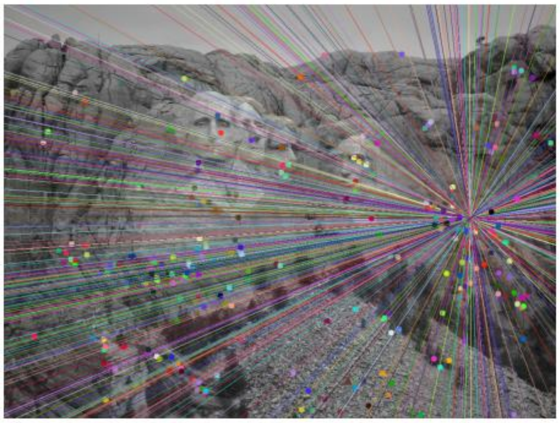
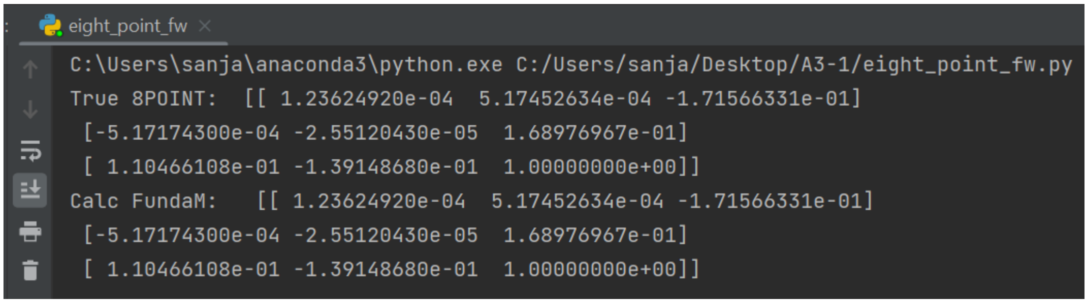


Image 2

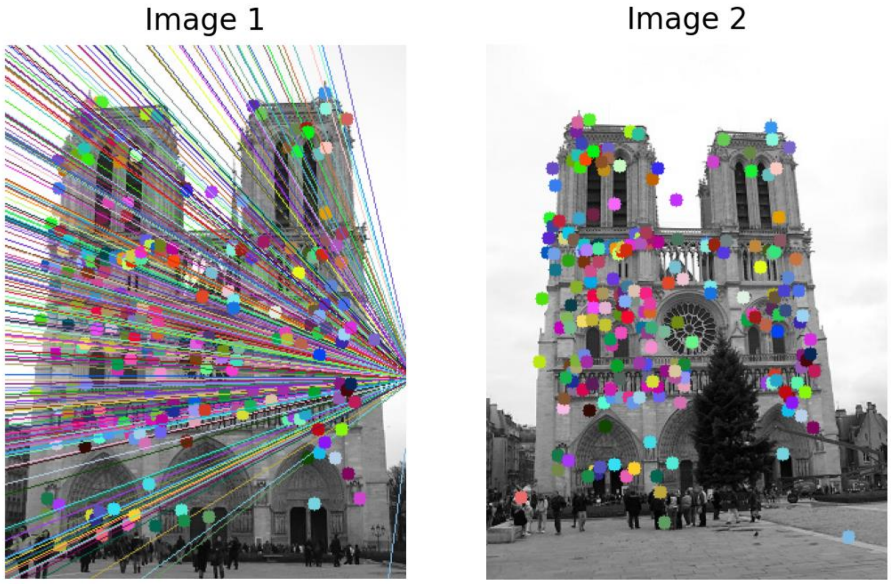
[3] For image pair: notredam_1 and notredam2

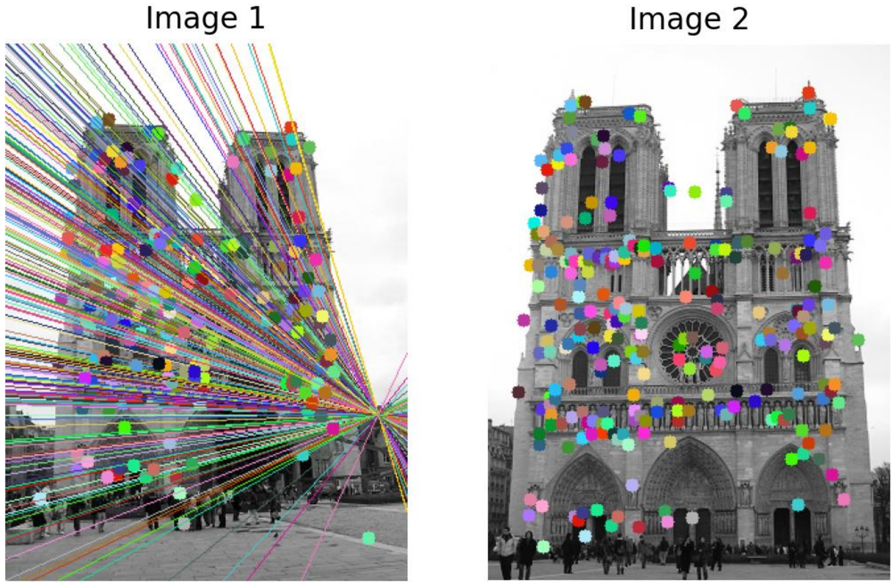


```
C:\Users\sanja\anaconda3\python.exe C:/Users/sanja/Desktop/A3-1/eight_point_fw.py
True 8POINT:  [[ 1.23624920e-04  5.17452634e-04 -1.71566331e-01]
 [-5.17174300e-04 -2.55120430e-05  1.68976967e-01]
 [ 1.10466108e-01 -1.39148680e-01  1.00000000e+00]]
Calc FundaM:   [[ 1.23624920e-04  5.17452634e-04 -1.71566331e-01]
 [-5.17174300e-04 -2.55120430e-05  1.68976967e-01]
 [ 1.10466108e-01 -1.39148680e-01  1.00000000e+00]]
```

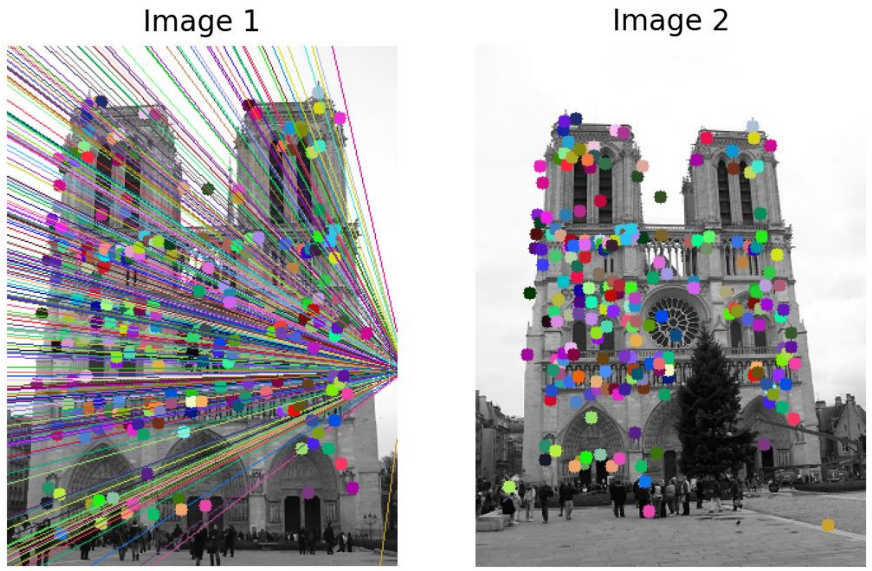
Identical true and calculated fundamental matrices are shown above.

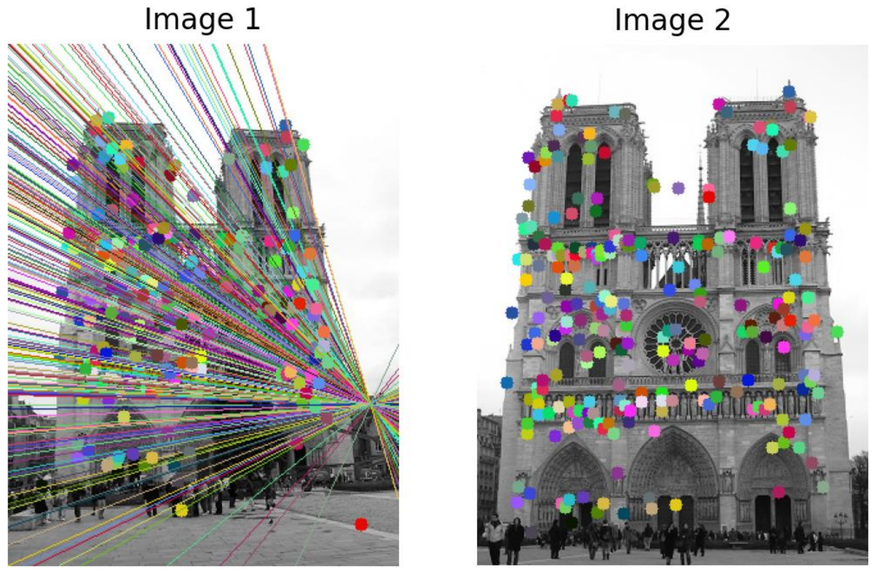
True fundamental matrix using opencv:



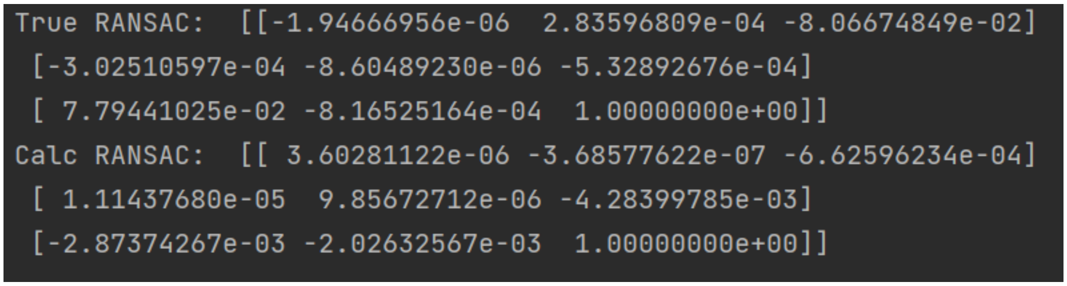


Calculated Fundamental Matrix:



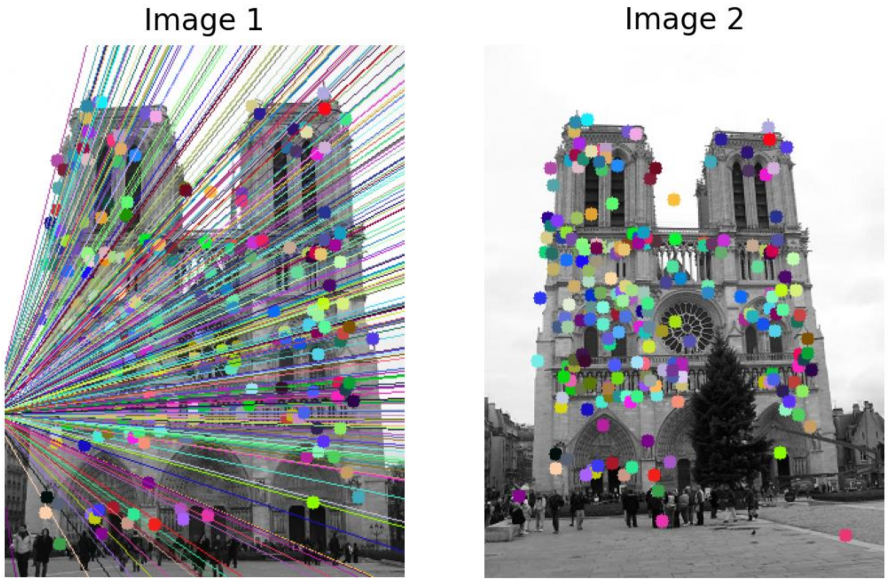


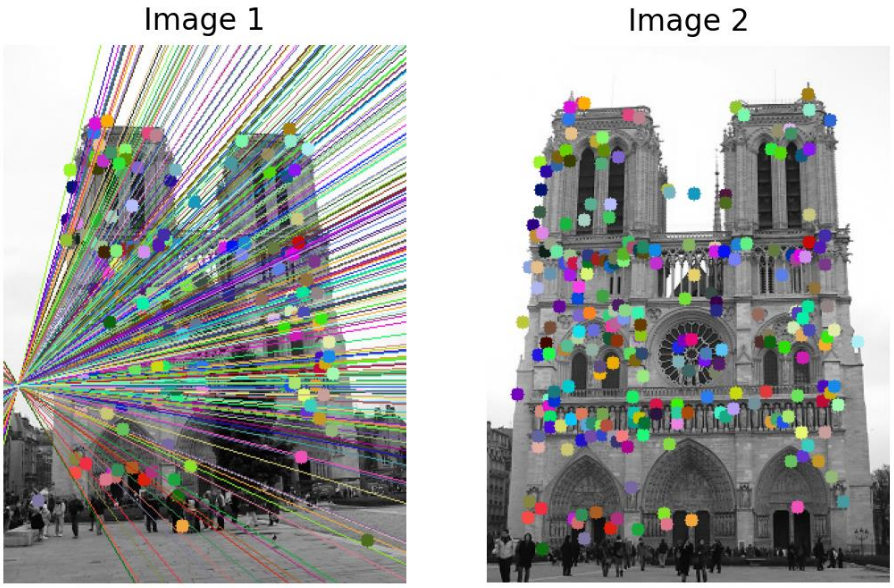
USING RANSAC:



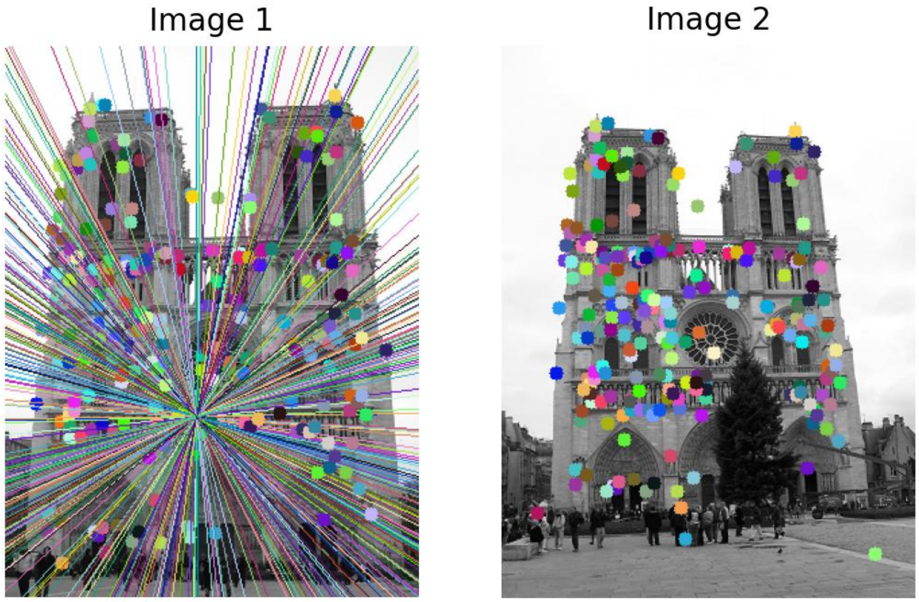
```
True RANSAC:  [[-1.94666956e-06  2.83596809e-04 -8.06674849e-02]
 [-3.02510597e-04 -8.60489230e-06 -5.32892676e-04]
 [ 7.79441025e-02 -8.16525164e-04  1.00000000e+00]]
Calc RANSAC:  [[ 3.60281122e-06 -3.68577622e-07 -6.62596234e-04]
 [ 1.11437680e-05  9.85672712e-06 -4.28399785e-03]
 [-2.87374267e-03 -2.02632567e-03  1.00000000e+00]]
```

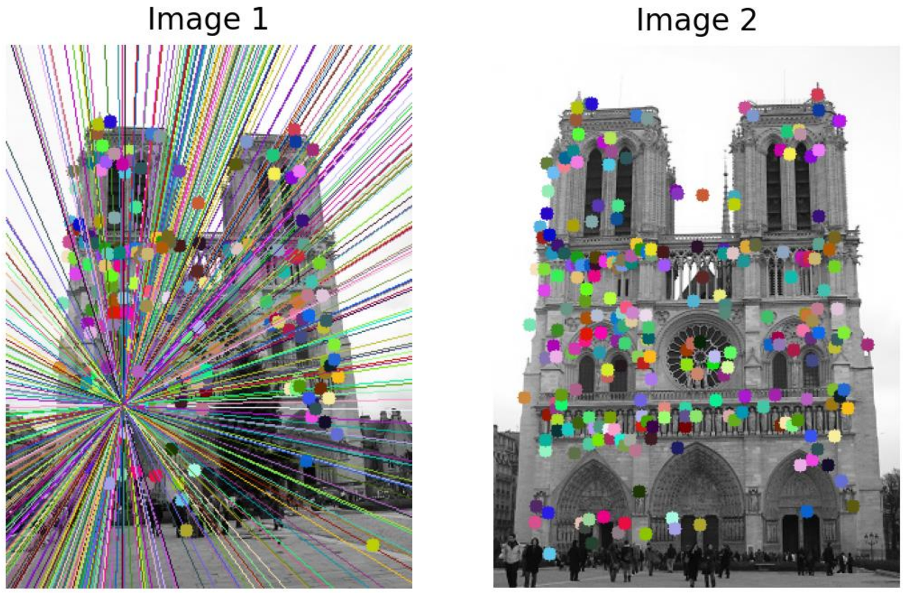
True fundamental matrix with RANSAC using opencv:



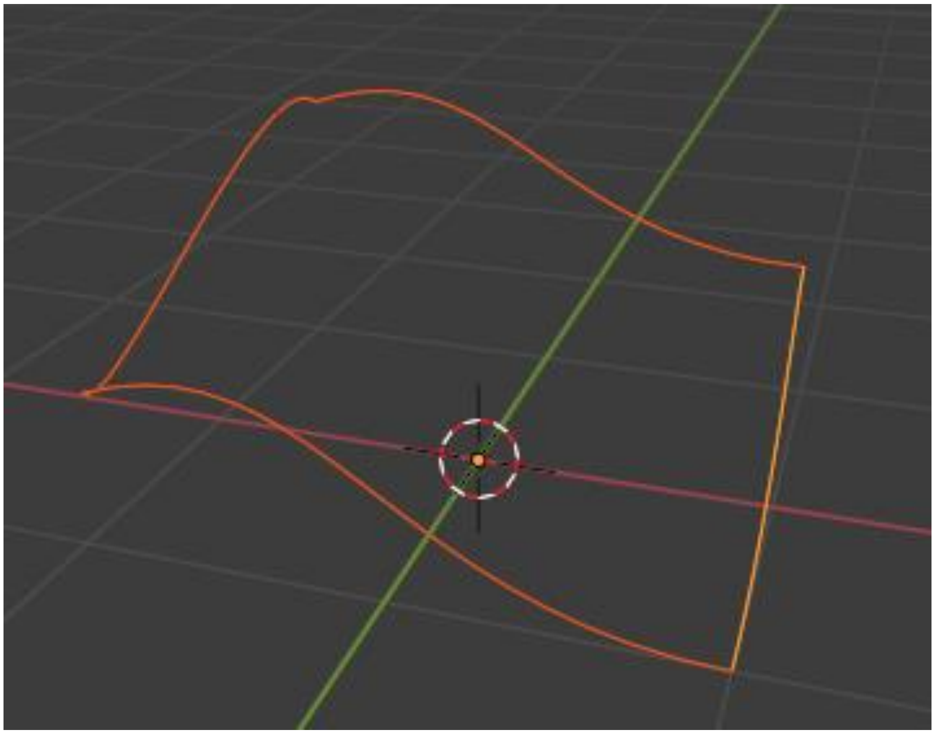


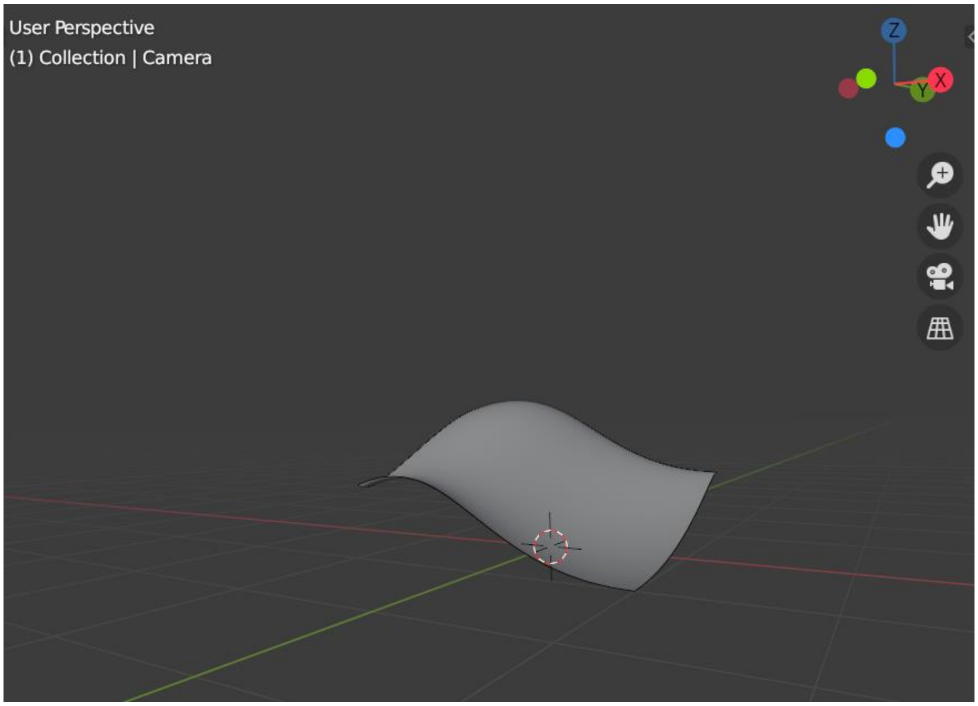
Calculated:

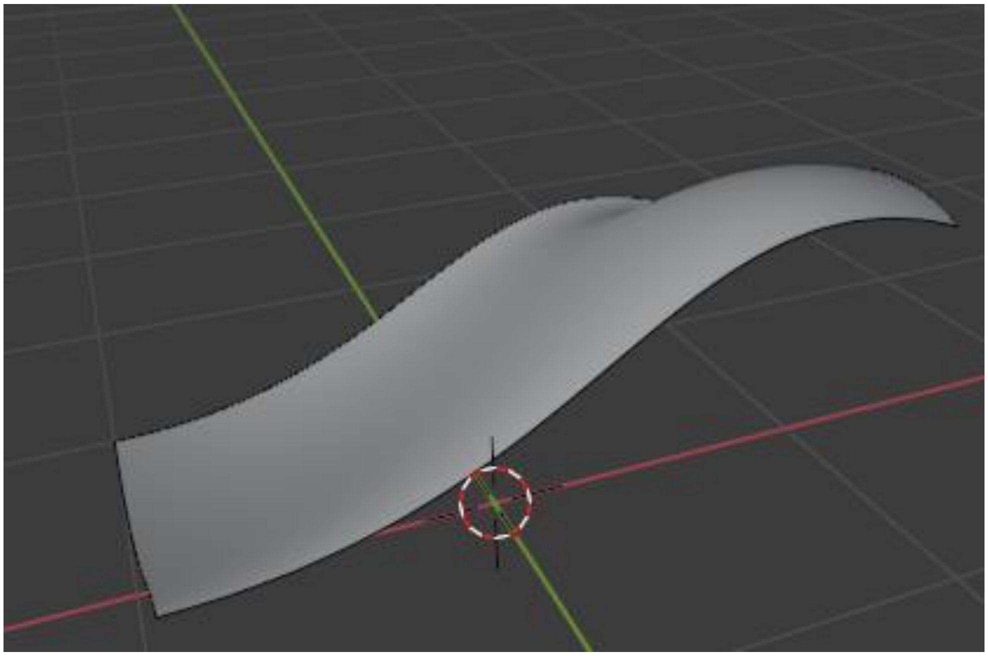




PART 2: BLENDER SCRIPTING







Methodology used:

- **Read vertices from text file**
- **Reshape lines in 4X4 where each line gives coordinates for 1 Bezier curve**
- **Split the coordinates**
- **Form 4 polylines and append the points onto them to get 4 Bezier curves as shown in the first image**
- **Interpolate 100 points each on the 4 curves**
- **Use the formulae for m1, m2 and m3**
- **m1 makes a surface between curves 1 and 2**
- **m2 makes a surface between curves 3 and 4**
- **m3 makes a plane surface joining the corner points (bilinear interpolation)**
  formulae:

  $$L_c(s,t) = (1-t)c_0(s) + tc_1(s)$$

  $$L_d(s,t) = (1-s)d_0(t) + sd_1(t)$$

  $$B(s,t) = c_0(0)(1-s)(1-t) + c_0(1)s(1-t) + c_1(0)(1-s)t + c_1(1)st.$$
- **Coons patch is obtained as: m1+m2-m3 (shown by images 2 and 3)**