

HW 05 – REPORT

소속 : 정보컴퓨터공학부

학번 : 201824523

이름 : 안혜준

1. 서론

Epipolar Geometry에서 epipole을 찾고, epipole을 이용하여 epipolar line을 그린다.

서로 다른 위치에서 같은 대상을 찍은 두 이미지가 주어진다. 이미지와 함께 두 이미지에서 상응하는 특징점들의 배열 또한 주어진다. 특징점들을 이용하여 fundamental matrix를 찾을 수 있다.

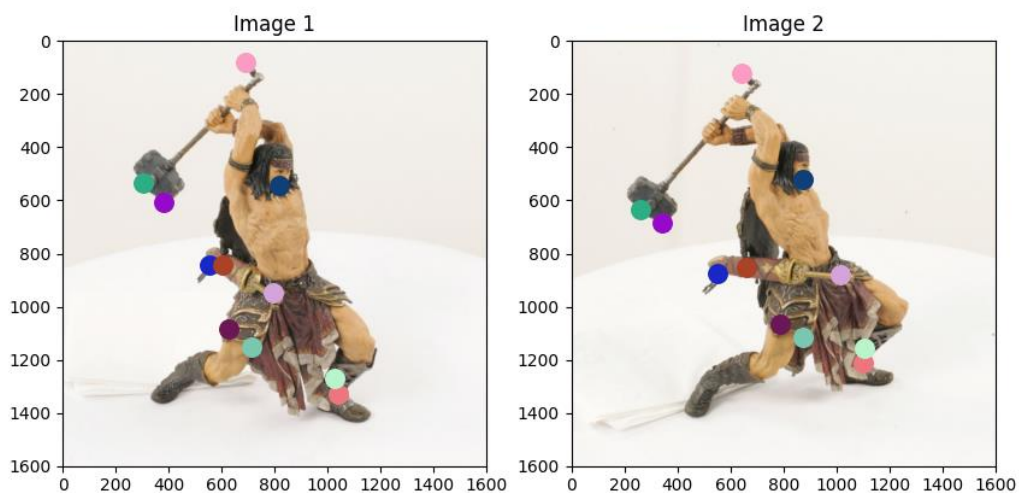
Fundamental matrix란 다음의 수식을 만족하는 3×3 matrix이다.

$$q^T F p = 0$$

- q : 한 이미지의 특징점 좌표(homogeneous coordinate)
- p : 다른 이미지의 특징점 좌표(homogeneous coordinate)
- F : Fundamental Matrix

이는 8개 이상의 특징점 쌍으로 8-point algorithm을 사용하여 구할 수 있다. 하지만 이때 구한 F 는 rank가 2가 아니다. F 의 rank는 2가 되어야 하므로, 특이값 분해로 나눠 마지막 특이값을 0으로 없애 rank를 2로 바꾼다.

Fundamental matrix를 사용하여 각 이미지의 epipole을 계산 가능하다. Epipole과 각 특징점을 잇는 직선을 그으면 epipolar line을 찾을 수 있다.



실습에 사용할 이미지쌍에 특징점을 마크하여 나타내었다. 특징점들을 사용하여 fundamental matrix를 구하고 epipolar line을 긋는 것이 이번 실습이 목표이다.

2. 본론

1. Fundamental Matrix Estimation

먼저 특징점들로부터 fundamental matrix를 만든다.

아래는 8-points algorithm의 계산식이다.

$$\begin{bmatrix} u_1 u_1' & v_1 u_1' & u_1' & u_1 v_1' & v_1 v_1' & v_1' & u_1 & v_1 & 1 \\ u_2 u_2' & v_2 u_2' & u_2' & u_2 v_2' & v_2 v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n u_n' & v_n u_n' & u_n' & u_n v_n' & v_n v_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

F를 구하기 위해 좌측의 A 행렬을 만들어야 한다.

```
A = np.array([]).reshape(0, 9)

for i in range(len(x1[0])):
    # for each matching points
    x = np.array([x1[0][i], x1[1][i], x1[2][i]])
    x_p = np.array([x2[0][i], x2[1][i], x2[2][i]])
    row = x_p @ x
    row = np.array(row.flatten())
    # add row to A
    A = np.vstack([A, row])
```

만든 A 행렬의 고유 값과 고유 벡터를 계산하고, 최저 고유 값에 대응되는 고유 벡터를 f 벡터로 선정한다. f는 다시 3x3 형태로 reshape되고, rank를 낮추기 위해 SVD를 수행해 rank를 낮춘다.

```
U, S, V = np.linalg.svd(F)
S_p = np.diag([S[0], S[1], 0])
F = U @ S_p @ V
```

2. Compute epipoles

$Fe = 0$ 을 만족시키는 점이 epipole이므로 F 를 다시 특이값 분해를 하여 마지막 영공간의 특이 벡터를 구하여 normalize하여 계산한다.

```
U, S, V = np.linalg.svd(F) # solve  $Fe_1 = 0$  by SVD
e1 = V[-1]
e1 = e1/e1[2] # normalization

U, S, V = np.linalg.svd(F.T) # solve  $F^T e_2 = 0$  by SVD
e2 = V[-1]
e2 = e2/e2[2] # normalization
```

각 이미지에 대하여 e_1 과 e_2 를 계산한다.

3. Epipolar lines

계산한 epipole과 각 특징점으로 epipolar line을 그려 시각화를 한다.

모든 특징점들에 대해 epipole과의 직선 방정식을 구한다.

```
#  $y = ax + b$ 
x1, y1 = cor1[0][i], cor1[1][i]
a1 = (y1 - e1[1]) / (x1 - e1[0])
b1 = y1 - a1 * x1

x2, y2 = cor2[0][i], cor2[1][i]
a2 = (y2 - e2[1]) / (x2 - e2[0])
b2 = y2 - a2 * x2
```

x좌표가 0일 때의 지점과, image의 최대 x 좌표인 경우 두 점을 계산한다.

```
# 앞/끝점에 대해 계산
min_x1 = 0
min_y1 = a1 * min_x1 + b1
max_x1 = img1.shape[1]
max_y1 = a1 * max_x1 + b1
min_x2 = 0
min_y2 = a2 * min_x2 + b2
max_x2 = img2.shape[1]
max_y2 = a2 * max_x2 + b2
```

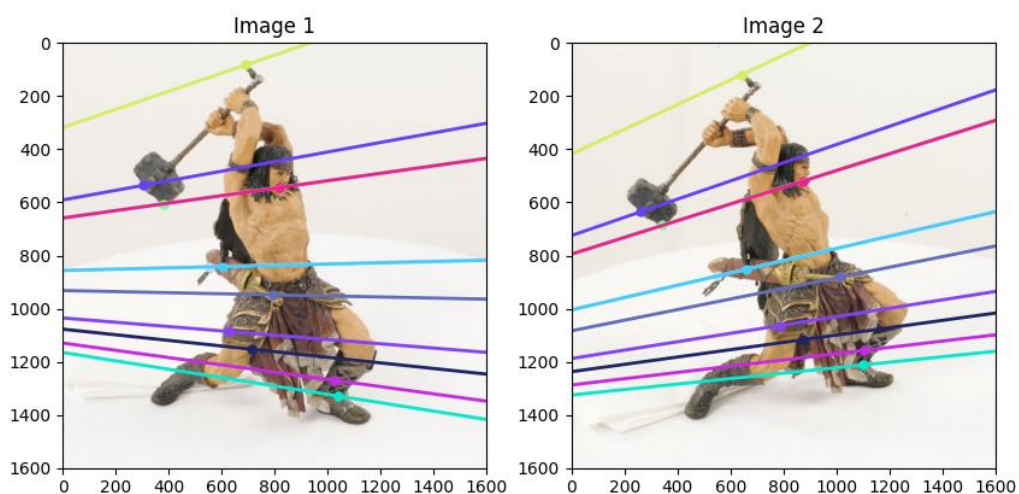
특징점들과 함께 직선을 그린다.

```
axes[0].scatter(x1, y1, color=colors, s=100, marker='.')
axes[1].scatter(x2, y2, color=colors, s=100, marker='.')

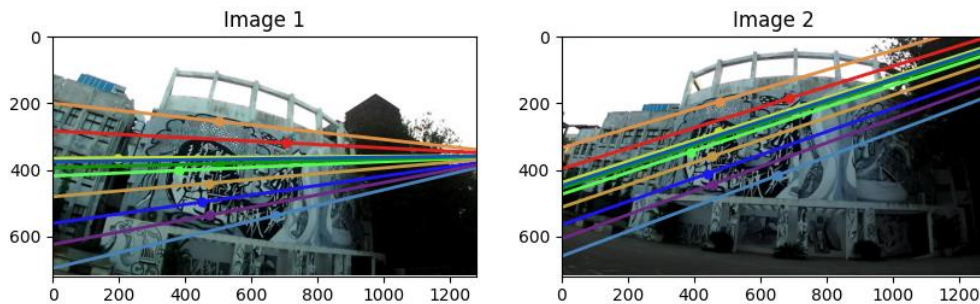
axes[0].plot([min_x1, max_x1], [min_y1, max_y1], color=colors)
axes[1].plot([min_x2, max_x2], [min_y2, max_y2], color=colors)
```

여기서 색은 각 특징점들에 대해 다르게 하기 위해 random color를 적용하였다.

```
colors = np.random.random(3)
```



Warrior 최종 이미지



Graffiti 최종 이미지

Warrior 이미지와 graffiti 이미지에 대한 최종 결과이다.

3. 결론 및 토의

같은 대상을 찍은 두 이미지로부터 매칭된 특징점쌍을 분석하여 epipolar line을 직접 그려보았다. 두 이미지는 내부의 각 좌표는 fundamental matrix와 함께 대응하는 특징점과 연관이 되어있다.

3d 공간의 객체는 카메라로 인해 이미지로 프로젝션된다. 정확하게는, 카메라와 객체를 잇는 광선(ray)이 각 카메라의 이미지 평면과 만나는 지점으로 프로젝션되는 것이다. 같은 객체를 찍은 두 카메라가 존재할 때, 두 카메라를 잇는 선을 베이스라인이라고 한다. 따라서 베이스라인을 연장했을 때, 이미지 평면과 만나는 지점이 바로 epipole이 된다.

Epipole은 한 이미지 평면에 다른 카메라를 투영시킨 지점이다. 카메라가 이미지 평면 뒤에 존재한다면, 실제 위치와 다르게 정반대로 투영될 것이다. 그렇기 때문에 결과 이미지와 같이 epipole이 둘다 왼쪽(warrior), 혹은 둘다 오른쪽(graffiti)으로 나올 수 있는 것이다