

# Introduction to Computer Vision HW05

## - Epipolar Geometry -

2021-1학기 컴퓨터비전개론 059분반 201724437 김재현

### Part 1. Fundamental Matrix Estimation

주어진 match들의 좌표인  $x_1$ 과  $x_2$ 를 이용하여  $Af = 0$ 을 만족하는  $A$  행렬을 구하고, 여기서 fundamental matrix를 얻어내는 코드입니다.

$x_1$ 과  $x_2$ 를 적절히 변환하여 2개의  $(n,9)$  크기를 갖는 행렬을 얻고, 이들의 원소를 각각 곱하면  $(n,9)$  크기의  $A$  행렬을 얻을 수 있습니다.

이  $A$  행렬을 특이값 분해한 뒤 최소 고유값에 해당하는 고유벡터를 찾으면 fundamental matrix를 얻을 수 있습니다. 이 때 이 행렬의 rank는 3이지만 원하는  $F$ 의 rank는 2이므로 이에 맞게 수정할 필요가 있습니다. 따라서 얻은 행렬에 다시 특이값 분해를 적용하여 Sigma 행렬에서 가장 작은 고유값을 0으로 하고, 다시 이를 행렬곱하면 원하는  $F$  행렬을 얻을 수 있습니다.

hw5.py - Line 12~46

```
def compute_fundamental(x1,x2):
    n = x1.shape[1]
    if x2.shape[1] != n:
        exit(1)

    F = None

    A = (np.tile(x1, (3,1)).T) * (np.repeat(x2, 3, axis=0).T)

    Au,As,Avt = np.linalg.svd(A)
    Fp = Avt[As.argmax()]
    Fp = Fp.reshape((3,3))

    u, s, vt = np.linalg.svd(Fp)

    s[-1] = 0

    F = u @ np.diag(s) @ vt

    return F
    return matched_pairs
```

### Part 2. Compute epipoles

Epipole을 구하기 위해서는  $F@e_1=0$ ,  $F.T@e_2=0$ 을 만족하는  $e_1$ ,  $e_2$ 를 각각 구해야 합니다. 다만 실제 사진에서는 위 두 식을 완전히 만족하는 경우가 드물기 때문에,  $F$ 와  $F.T$ 에 대한 최소의 (= 최소의 고유값에 대한) 고유벡터를 구하는 것이 적절합니다.

따라서  $F$ ,  $F.T$ 에 대해 각각 특이값 분해를 적용하여 최소 고유벡터를 찾은 뒤, 이 homogeneous coordinate들을 normalize해주면 구하는 epipole들( $e_1$ ,  $e_2$ )을 얻을 수 있습니다.

hw5.py - Line 76~99

```
def compute_epipoles(F):
    e1 = None
    e2 = None

    u1, s1, vt1 = np.linalg.svd(F);
    e1 = vt1[s1.argmax()]
    e1 = e1 / e1[2]

    u2, s2, vt2 = np.linalg.svd(F.T);
    e2 = vt2[s2.argmax()]
    e2 = e2 / e2[2]

    return e1, e2
```

### Part 3. Epipolar lines

구한 epipole들과 기존의 match들을 각각 이어주되, plot 전체를 가로질러야 하므로 plt.pyplot.axline() 함수를 사용하였습니다. 또한 각각의 epipolar line들이 서로 다른 색을 띄어야 하므로, 이 axline의 값에 무작위의 1\*3 배열을 넘겨주어 서로 다른 RGB 값을 갖도록 했습니다.

hw5.py - Line 102~131

```
def draw_epipolar_lines(img1, img2, cor1, cor2):
    F = compute_norm_fundamental(cor1, cor2)

    e1, e2 = compute_epipoles(F)

    fig = plt.figure(figsize=(24,12))
    ax1 = fig.add_subplot(1, 2, 1)
    ax2 = fig.add_subplot(1, 2, 2)

    for point in cor1.T:
        ax1.axline(e1[:2],point[0:2], color=(np.random.rand(3,)), lw=2)
    for point in cor2.T:
        ax2.axline(e2[:2],point[0:2], color=np.random.rand(3,)), lw=2)

    ax1.imshow(img1)
    ax2.imshow(img2)
    plt.show()

    return
```

위 코드들을 실행하여 최종적으로 생성된 warrior 이미지와 graffiti 이미지는 아래와 같습니다.

>>

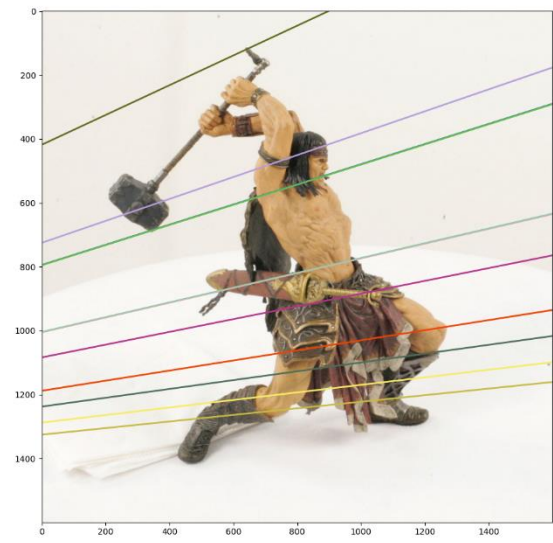
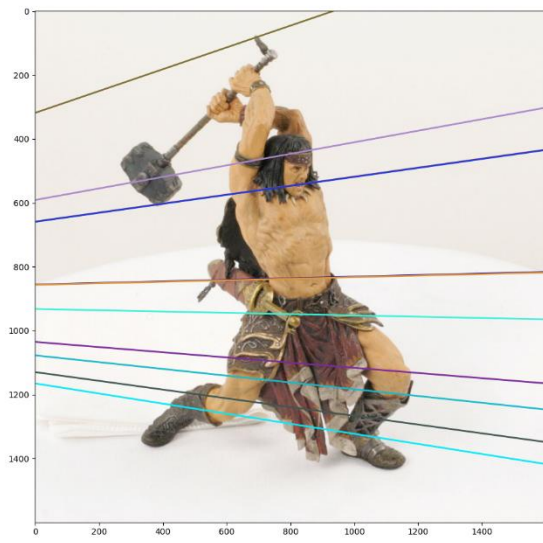


그림 1 – Warrior.png

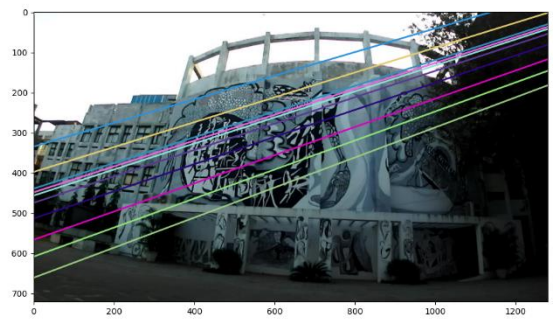
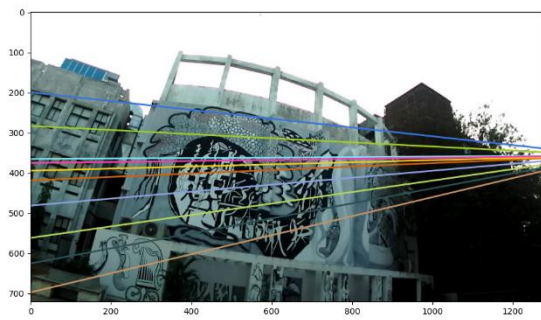


그림 2 – Graffiti.png