

HW 05 – REPORT

소속 : 정보컴퓨터공학부

학번 : 202055536

이름 : 민 예 진

1. 서론

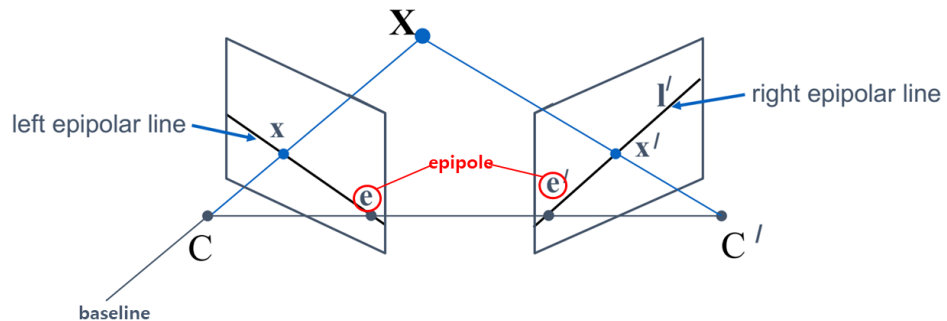
1) 실습 목표

Fundamental matrix를 활용하여 같은 대상을 서로 다른 카메라로 촬영한 이미지에 epipolar line을 출력한다.

2) 이론적 배경 기술

① Epipolar Geometry

스테레오 비전(stereo vision)에서의 기하를 뜻한다.



- epipole : 두 카메라 원점을 잇는 선과 이미지 평면이 만나는 점 e, e'
- epipolar line : 투영점 x 와 epipole을 잇는 직선
- baseline : 두 카메라 원점을 잇는 선

② Fundamental Matrix

두 이미지에서 x, x' 매치에 대해

$$x'^T F x = 0$$

을 만족시키는 F 를 Fundamental matrix라고 정의한다.

$$x = (u, v, 1)^T, x' = (u', v', 1)^T, F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \text{ 라고 할 때}$$

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

의 해를 통해 Fundamental matrix를 구할 수 있다.

2. 본론

1) Fundamental Matrix Estimation

- compute_fundamental

```
# build matrix for equations in Page 51
matrix_a = []
for e1, e2 in zip(t1, t2):
    matrix_a.append([e1[0]*e2[0], e1[1]*e2[0], e2[0], e1[0]*e2[1], e1[1]*e2[1], e2[1], e1[0], e1[1], 1])
matrix_a = np.asarray(matrix_a)

# compute the solution in Page 51
ATA = matrix_a.T @ matrix_a
eigvals, eigvecs = np.linalg.eig(ATA)
v = eigvecs[:, np.argmin(eigvals)]
F = v.reshape(3, 3)

U, S, V = np.linalg.svd(F)
# constrain F: make rank 2 by zeroing out last singular value (Page 52)
sigma = np.diag(S)
sigma[2][2] = 0

F = U @ sigma @ V
```

compute_fundamental 함수를 통해 근사된 Fundamental Matrix를 구한다.

이때, eigenvector를 이용하여 근사된 Fundamental Matrix를 구하고 Fundamental Matrix의 rank를 2로 만들기 위해 SVD의 Sigma의 마지막 원소 값을 0으로 한다.

2) Compute epipoles

- compute_epipoles

```
U,S,V = np.linalg.svd(F)    # solve Fe1 = 0 by SVD
e1 = V[-1]
e1 = e1/e1[2]
```

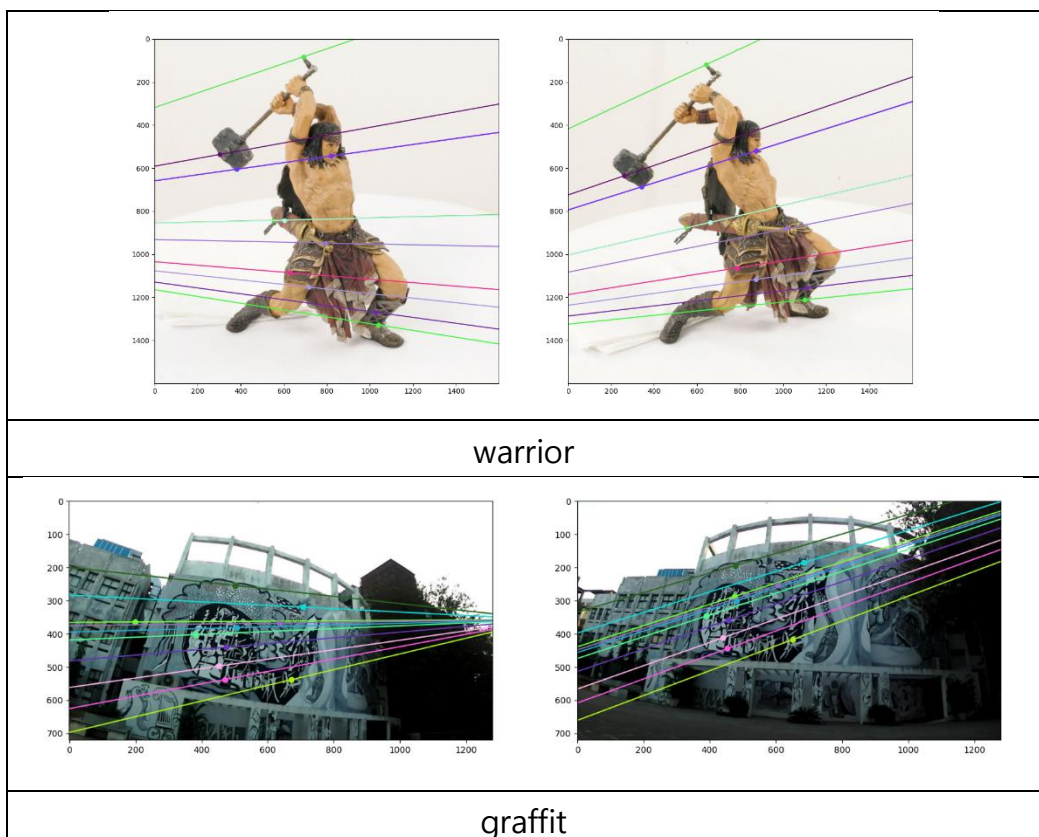
3) Epipolar lines

- draw_epipolar_lines

```
def cal_ab(x1, y1, x2, y2):  
    a = (y2 - y1) / (x2 - x1)  
    b = (x2*y1 - x1*y2) / (x2 - x1)  
    return a, b  
  
cor1 = np.transpose(cor1)  
cor2 = np.transpose(cor2)  
  
size = img1.shape  
for c1, c2 in zip(cor1, cor2):  
    color = (np.random.random(), np.random.random(), np.random.random())  
  
    x1, y1 = e1[:2]  
    x2, y2 = c1[:2]  
    a, b = cal_ab(x1, y1, x2, y2)  
    x = np.array([0, size[1]])  
    axes[0].plot(x, a*x+b, c = color)  
    axes[0].plot(x2, y2, 'o', c = color)
```

epipole와 point이 이루는 직선의 기울기와 y절편을 구한 후 이미지의 크기에 맞는 y좌표를 구하여 이를 출력한다. 이때 epipolar line별로 색을 다르게 하기 위하여 np.random.random()을 활용했다.

실행결과



3. 결론

일반 카메라로 찍은 사진은 3차원 물체를 2차원에 투영 시킨 것이기 때문에 3차원 정보 중 카메라에서 물체까지의 거리에 대한 정보가 사라지게 된다. 이 때문에 거리 정보를 알기 위해서는 같은 물체를 또 다른 위치에서 찍은 다른 사진이 필요하고 두 사진의 차이를 이용하여 거리 정보를 얻을 수 있다. 이와 관련된 학문 분야가 스테레오 비전이고 이번 실습을 통해 스테레오 비전과 관련된 개념을 코드로 직접 구현하는 활동을 해보았다.

스테레오 영상을 처리하면서 발생했던 이슈들을 정리해보고자 한다.

이슈	해결 방법
영상에서 3차원 정보를 얻고 싶음	서로 다른 위치에서 촬영한 다수 영상(최소 2장 이상)이 필요하다. 이를 위해 스테레오 카메라를 많이 사용한다.
스테레오 카메라에서 거리 정보를 얻고 싶음	Disparity(= baseline/depth)를 찾아야 한다. 이를 찾기 위해서는 matching이 잘 되어야 한다.
매칭이 느림	Epipolar constraint를 이용한다. Epipolar constraint를 찾기 위해서는 스테레오 카메라 calibration이 선행되어야 한다.
스테레오 카메라 calibration을 어떻게 하는가?	직접 알고리즘을 구현하면 좋지만 그렇지 못한 경우 OpenCV에서 제공하는 함수를 이용한다.
Rectification이란?	스테레오 카메라에서 찾아낸 epipolar line을 서로 평행하게 일치시키는 과정으로 이를 통해 매칭에 소요되는 시간을 대폭 줄일 수 있어 스테레오 매칭에서 필수적인 과정이다.

Camera calibration : 카메라의 파라미터(parameters)를 추정하는 과정

참고자료

'[13호]스테레오 영상을 처리하는데 있어 발생하는 이슈', Media Contents, 2012년 06월 25일 수정, 2023년 05월 20일 접속, <http://www.ntrexgo.com/archives/2280>