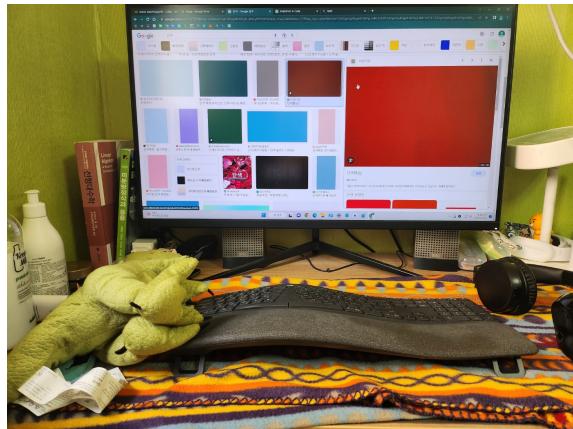




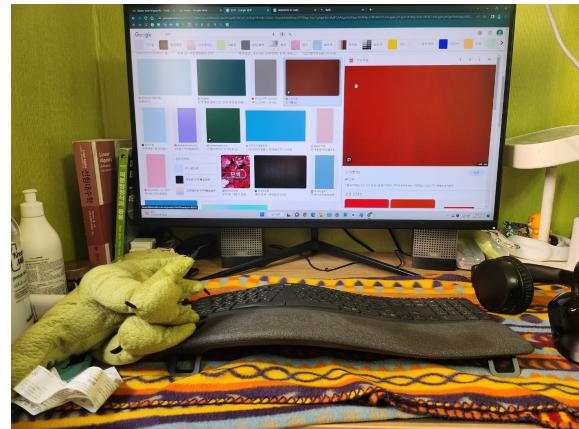
3DDP23 HW02

☰ 유형
☰ 교수님
⌚ 학년/학기

Left 이미지와 Right 이미지



left.png



right.png

해당 이미지를 camera calibrate 값을 알기 위해서 과거의 check 판으로 통해 구했던 것을 이용해서 값을 가져오게 됨

```
ret, intrinsic, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints,  
gray.shape[::-1], None, None)
```

해당 과정으로 구하게 된 Intrinsic matrix와 dist 값을 구함

Intrinsic matrix:

```
[2.94943404e+03 0.00000000e+00 1.51762115e+03]  
[0.00000000e+00 2.95078257e+03 1.99613776e+03]
```

```
[0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

dist:

```
[[ 4.18559859e-02 -2.96803535e-01  5.45439966e-04  
 3.25139790e-05  
 6.89842309e-01]]
```

실제 이미지에 적용되야 하는 회전과 이동벡터 (실제 카메라 촬영의 2개의 이미지에는 회전이 없고, 이동된 것이 없음) (때문에 해당 값을 아래로 초기화함)

rvecs(회전 벡터):

```
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]
```

tvecs(이동 벡터):

```
[1. 0. 0.]
```

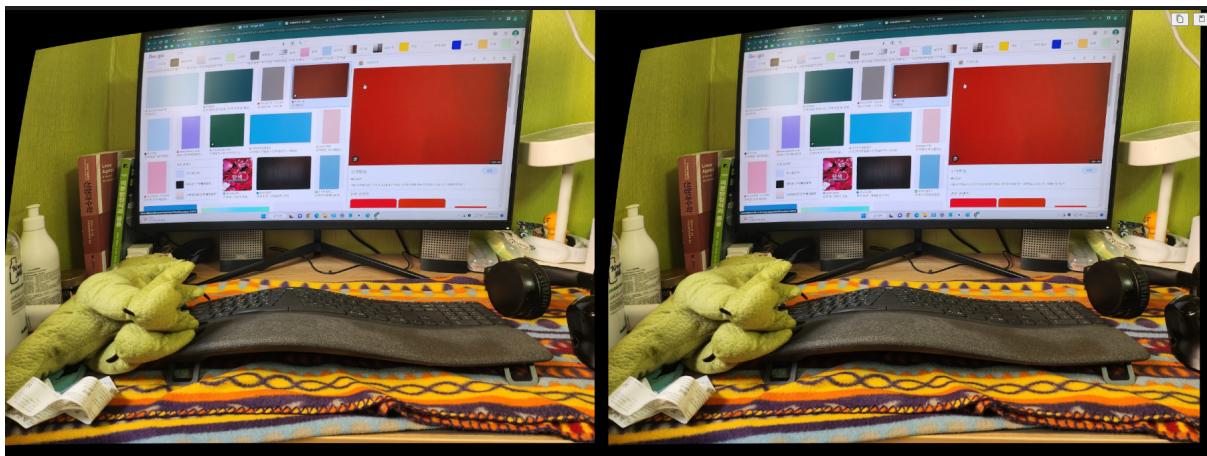
이미지 Rectification

cv2.stereoRectify : 스테레오 이미지의 회전 행렬(Rotation1, Rotation2)과 투영 행렬(Projection1, Projection2) 을 구함.

cv2.initUndistortRectifyMap : 왼쪽 이미지와 오른쪽 이미지의 보정 맵

(rectification_map1_x, rectification_map1_y, rectification_map2_x, rectification_map2_y)을 생성

cv2.remap : remap 함수를 사용하여 왼쪽 이미지(left)와 오른쪽 이미지(right)를 보정된 이미지(left_rectified, right_rectified)로 변환



이미지 DepthMap 추출:

1. 이미지 회색으로 변환함.
2. 스테레오 매처(StereoBM) 개체를 생성함.
3. 스테레오 매처를 사용하여 거리차(disparity)를 계산함.
→ disparity(거리차도) 결과값 도출



3D cloud map 생성

베이스라인 :

카메라간 거리를 의미함. (잘 맞춘 것 같은데 잘 모르겠음)

해당 값 1.3이라고 측정했지만 1.8로 늘리는게 더 정확했음

초점 거리 평균 :

스테레오 카메라 시스템에서 사용되는 두 개의 카메라의 초점 거리(focal length)의 평균값

거리(Z) : `z = baseline * focal_length_mean / disparity[v, u]`

- baseline(기준선 길이)과 평균 초점 거리(focal length)의 곱을 disparity(거리 차)로 나눈 값.
- 이 수식은 disparity(거리 차)가 클수록 더 가까운 점에 해당하는 거리(Z)를 의미함.

X좌표 : `x = (u - principal_point_x) * z / focal_length_x`

- 픽셀의 x 좌표(u)에서 주요 포인트(광학 중심)의 x 좌표(principal_point_x)를 뺀 값에 거리(Z)를 초점 거리(focal length)의 x 성분(focal_length_x)으로 나눈 값임.
- 이 수식은 픽셀의 x 좌표(u)와 주요 포인트의 x 좌표와의 차이가 클수록 점의 실제 좌표(X)도 더 멀어짐을 의미함.

