

CSED441 Intro. to Computer Vision : Assn1 Report

October 8, 2025

Student 20240505 Hyunseong Kong

1. Overview

The first assignment is to implement a panorama stitching system.

2. Discussion

2-1. SIFT detection

contrast_th (Contrast Threshold): When this parameter is increased,

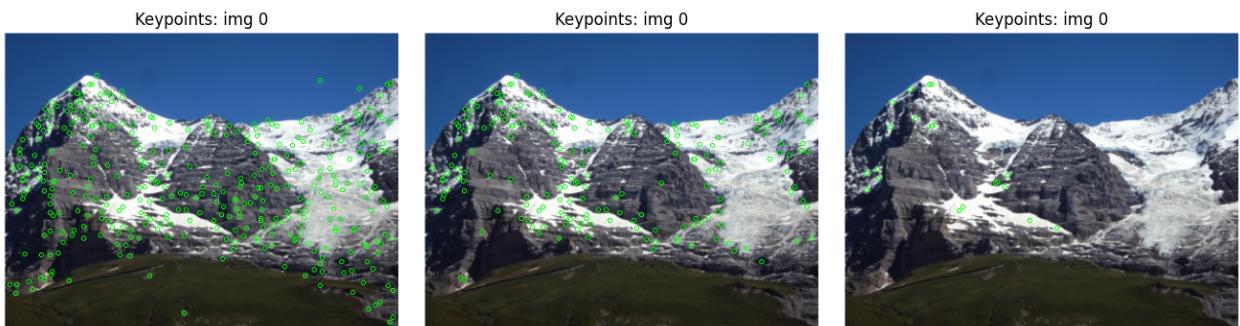
- (1) keypoint distribution: keypoints 수 감소, 대비가 강한 영역에 집중됨.
- (2) matching quality: 증가한다.
- (3) panorama stitching: keypoint 수 감소로 인해, 파노라마 정합 어려워짐.

	Smaller	Baseline	Larger
keypoint distribution	많음	중간	적음
matching quality	낮음	중간	높음
panorama stitching	RANSAC 많이 돌리면 안정적	안정적임	불안정적

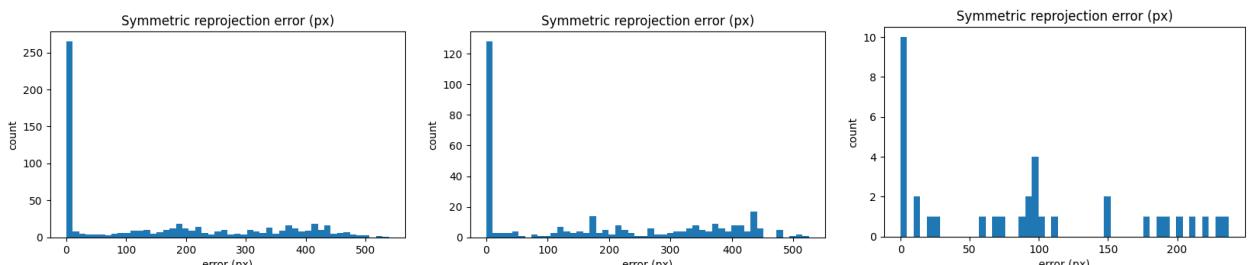
Explanation: 이 파라미터는 non max suppression에 사용된다. 만약, DoG(Difference of Gradient) pyramid의 각 octave, level 내부의 값이 contrast_th보다 작으면, max인지 여부에 상관없이 무시한다. 즉, 이 파라미터가 커질수록 불확실한 정보의 수가 줄어든다. 파노라마 정합을 위해서는 각 이미지 사이에 충분한 대응점 수가 존재해야 하기 때문에, contrast_th를 너무 높이면, 균일한 색상이 분포한 영역에서 특징점이 거의 사라져 두 이미지 간의 매칭점이 부족해진다. 그러면 RANSAC으로 homography를 계산할 때, 정합에 실패하거나 이상하게 겹치는 상황이 발생할 가능성이 있다. 따라서 contrast_th를 적당히 높이면 안정적인 매칭에 기여하지만, 너무 높으면 대응되는 점이 부족하여 이미지 겹치기가 올바르게 수행되지 않는다.

Evidence:

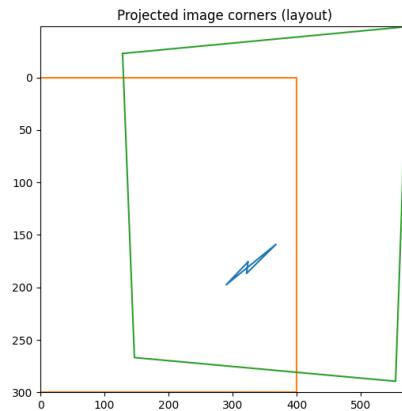
아래는 contrast_th를 순서대로 0.01, 0.03, 0.07로 설정하고 전체 파이프라인을 실행한 결과이다.



contrast_th가 증가하면 전체적으로 keypoint의 개수가 줄어드는 것을 확인할 수 있다.



가장 contrast_th가 큰 오른쪽은 최대 오차가 약 250px 정도에 머무는 것을 확인할 수 있다. 이것은 contrast_th를 높이면 matching quality가 높아진다는 증거이다.



위는 contrast_th == 0.07인 상황에서의 project layout을 보여준다. 여기서 확인할 수 있는 것처럼, 너무 높은 contrast_th 값은 매칭의 개수 자체가 너무 적어 파노라마 정합이 되지 않는다.

2-1. SIFT detection

edge_th (Edge Threshold): When this parameter is increased,

- (1) the types of keypoints selected: edge인 keypoint가 더 많이 선택되게 된다.
- (2) stability of correspondences: 감소한다.
- (3) panorama alignment: 매칭 안정성이 하락하여, 파노라마 정합이 어려워진다.

	Smaller	Baseline	Larger
the types of keypoints selected	corner만 많음	corner와 edge가 섞임	edge-like의 비율이 커짐
stability of correspondences	높음	중간	낮음
panorama alignment	높음	중간	낮음

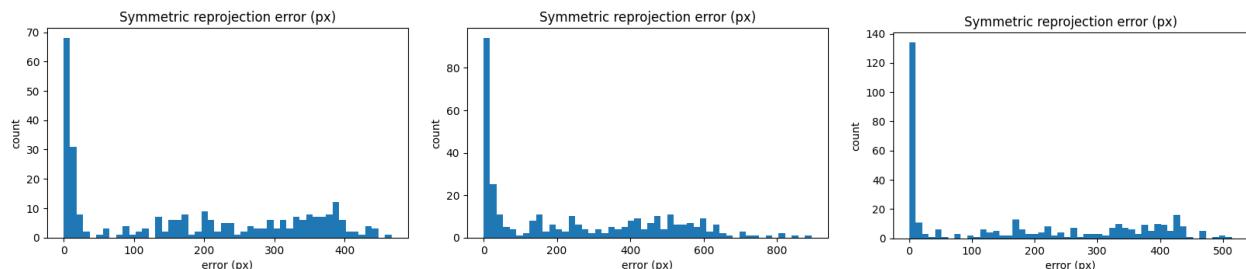
Explanation: 이 파라미터는 hessian edge rejection에 사용된다. 우선, edge rejection은 아래와 같은 식에 의해 이루어진다. 여기서 H 는 hessian matrix이고, e 가 edge_th 이다.

$$\frac{\text{tr}(H)^2}{\det(H)} < \gamma = \frac{(e + 1)^2}{e}$$

즉, 이 파라미터는 작을수록 더 많은 edge-like keypoint가 걸러지게 된다. 달리 말해, 값이 작을수록 corner에 가까운 keypoint만 남게 된다. 반면, edge는 한 쪽 방향으로만 뛰었기 때문에, 이 파라미터가 클수록 매칭 안정성이 하락한다. 이러한 하락은 파노라마 이미지가 제대로 형성되는 것을 방해한다.

Evidence:

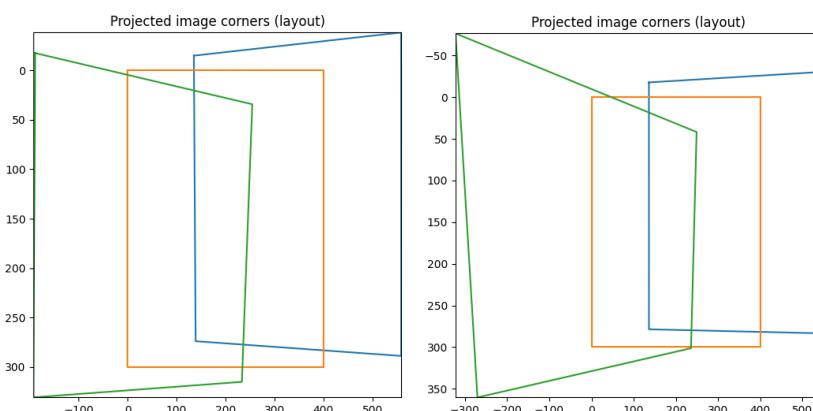
아래는 edge_th 를 순서대로 5.0, 10.0, 50.0으로 설정하고 전체 파이프라인을 실행한 결과이다. 보다 극단적인 결과를 확인하기 위해, $\text{iters} == 150$ 으로 설정하여, 의도적으로 RANSAC의 횟수를 줄였다.



우선 위 결과를 통해 edge_th 가 커질수록, 탐지되는 전체 keypoint가 많아지는 것을 확인할 수 있다. 낮은 edge_th 에서 탐지된 keypoint는 높은 edge_th 에서도 동일하게 탐지될 것이기에, 이러한 증가는 널널한 기준으로 인한 edge-like 데이터의 증가인 것으로 추측된다.

```
edge_th == 5.0: inliers: 61 / 288
edge_th == 10.0: inliers: 23 / 327
edge_th == 50.0: inliers: 62 / 349
```

위 3개 예시에서 edge_th 에 따른 inlier 개수 분포는 위와 같다. 즉, inlier의 상대적 비율이 높은 1번이 2번보다 연결 안정성이 높다. 3번이 2번보다 연결 안정성은 높게 측정되었지만, 아래의 그림에서 projection이 제대로 되지 않은 것을 확인할 수 있는 것처럼, edge_th 가 너무 높으면 오탐지 비율이 높아져 stability가 낮아진다. (아래는 순서대로 2번, 3번의 project layout)



2-1. SIFT detection

max_kp (Maximum Keypoints): When this parameter is increased,

- (1) the number of matches: 증가한다.
- (2) runtime: 증가한다.
- (3) the resulting panorama: 후보군의 증가로 안정적으로 정합된다.

	Smaller	Baseline	Larger
the number of matches	적음	중간	많음
runtime	적음	중간	큽
the resulting panorama	불안정	안정적	매우 안정적

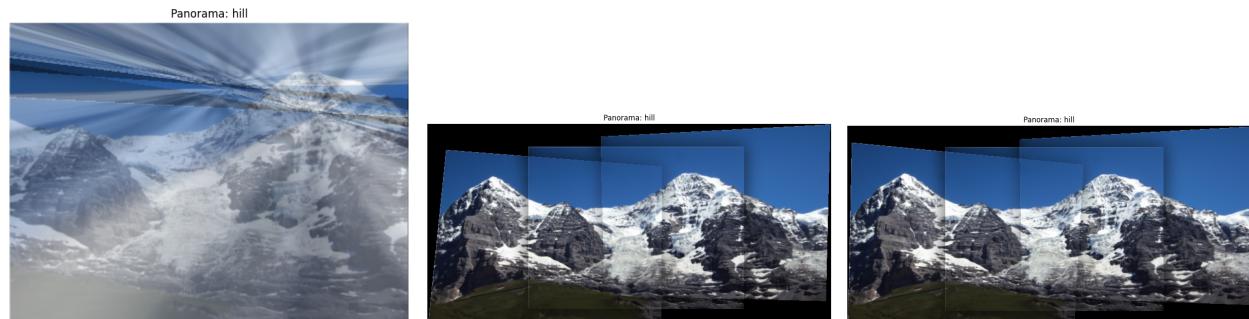
Explanation: 이 파라미터는 선택된 후보군의 최대 개수를 조절하는 역할을 한다. 만약 이 값이 너무 작다면 후보군이 부족해 매칭의 수가 적어지고 이에 정합이 불안정하게 수행된다. 반면에, 이 값이 크다면 후보군은 충분해 매칭의 수가 많아지고 정합이 보다 안정적으로 변하지만, 실행 시간은 늘어나게 된다.

Evidence:

아래는 `max_kp`를 순서대로 10, 100, 1000으로 설정하고 전체 파이프라인을 실행한 결과이다.



위 결과를 통해 `max_kp`가 커질수록 후보군의 개수가 함께 증가하는 것을 확인할 수 있다.



만약 후보군이 너무 적다면 1번 이미지처럼 파노라마가 제대로 만들어지지 않는다. 2번과 3번은 유사한 파노라마 이미지를 보여주는데, 이는 특정 후보군을 넘으면 그보다 많은 후보군들은 큰 영향을 주지 않기 때문이다.

2-2. Matching

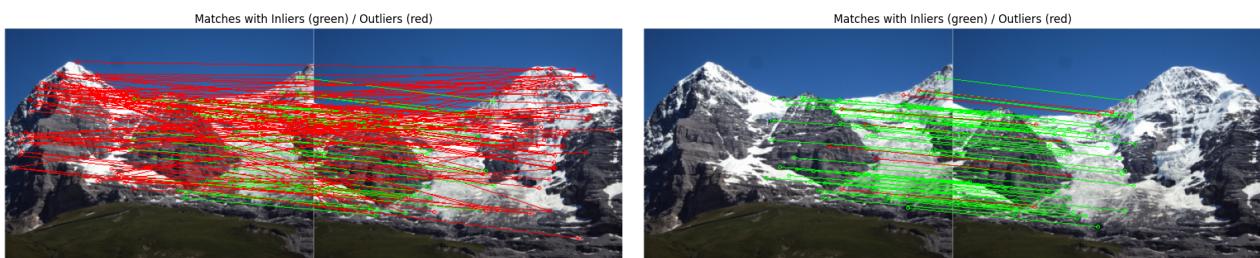
match_method Compare the results when using cross-check matching versus Lowe ratio matching.

- (1) number of matches: crosscheck > ratio
- (2) inlier ratio: crosscheck < ratio
- (3) panorama quality: crosscheck \approx ratio

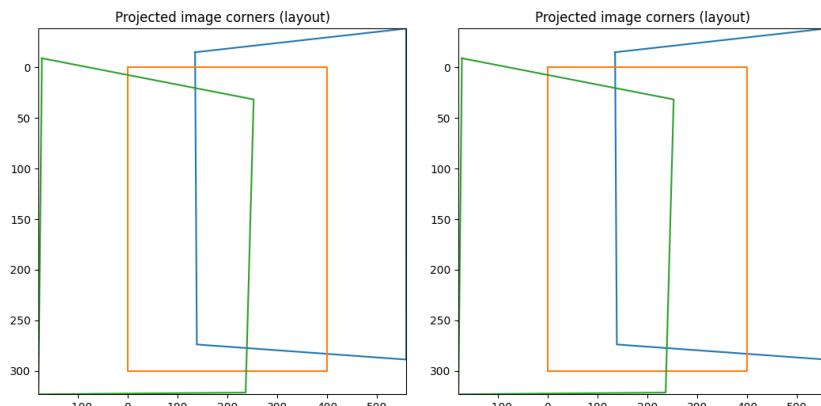
Explanation: lowe ratio 매칭은 가장 가까운 점과 두번째로 가까운 점의 비율을 본다. 그 비율이 1과 크게 차이난다면 가장 가까운 점이 매우 좋은 후보이다. 반면, 그 비율이 1과 유사하다면 매칭 자체가 의미없는 잘못된 매칭일 확률이 높아진다. lowe ratio 매칭은 이러한 비율이 특정 threshold 이상인 것만 취하기 때문에, 개수는 적을지언정 각 matching의 품질은 올라간다. 반면, crosscheck은 상호검증하는 방식이다. 이에, crosscheck 방식은 일관성이 없는 짹은 잘 제거하지만, lowe와 같이 이 매칭이 독보적인지는 확인하지 않는다. 따라서, crosscheck로 확인한 매칭은 개수는 많을지언정 각 matching의 품질이 떨어진다.

Evidence:

아래는 순서대로 crosscheck, ratio 매칭을 수행한 결과이다.



매칭의 수는 crosscheck가 많은 반면, inlier의 비율은 ratio가 많음을 확인해볼 수 있다.



파노라마는 비슷하게 생성됨을 확인해볼 수 있다.

2-3. RANSAC

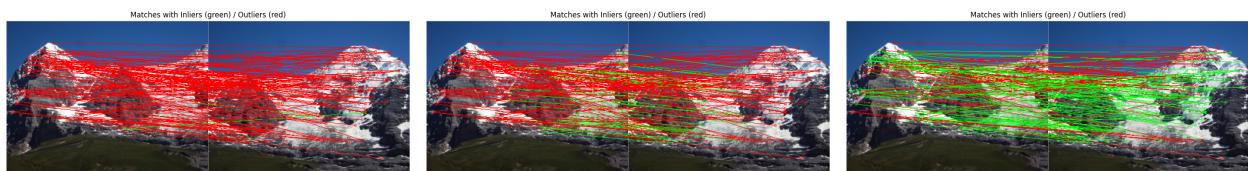
ransac_th (Inlier Threshold in pixels): When this parameter is increased,

- (1) inlier selection: 증가한다.
- (2) model accuracy: 감소한다.
- (3) visual quality of the panorama: 감소한다.

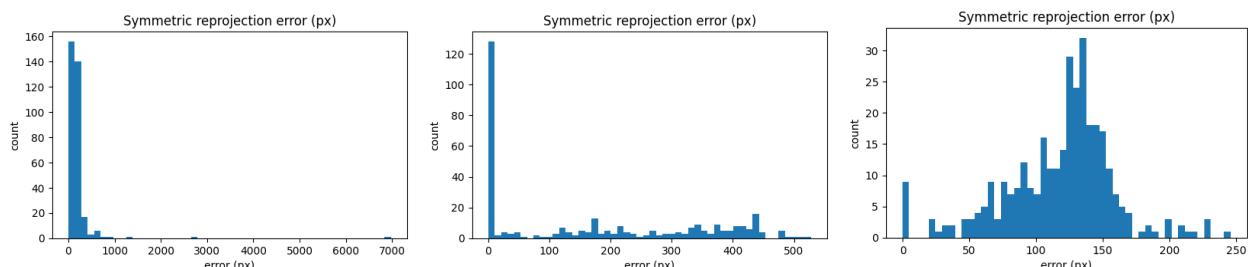
Explanation: 이 파라미터는 랜덤한 hyperplane을 결정한 뒤 그 plane이 얼마나 잘 만들어졌는지 체크하기 위한 값으로, plane으로부터 ransac_th 이내로 떨어진 점의 개수를 세어 얼마나 plane이 잘 골라졌는지를 확인하게 된다. 만약, 이 값이 극단적으로 커진다면, 항상 모든 점들이 선택될 것이고 이에 처음 추정한 plane이 최종 model이 되어 정확도는 감소하게 된다. 추정한 plane으로 homography를 만들기 때문에 panorama의 시각적 퀄리티 또한 떨어진다. 반면에, 이 값이 너무 작다면 어떠한 점들도 선택되지 않을 것이기에 마찬가지로 model의 정확도와 panorama의 시각적 퀄리티가 떨어진다.

Evidence:

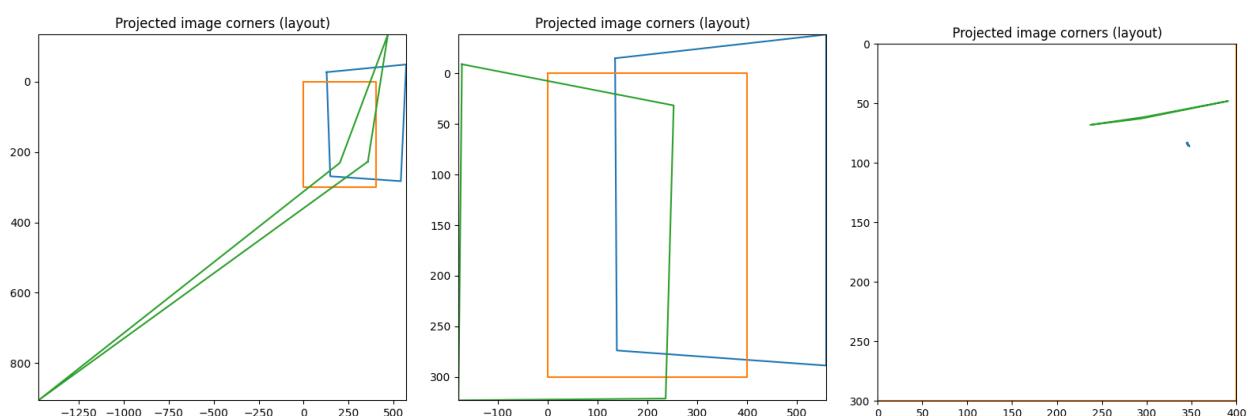
아래는 ransac_th를 순서대로 0.1, 1.5, 300.0으로 설정하고 전체 파이프라인을 실행한 결과이다.



위 결과를 통해 ransac_th가 커질수록 모델이 결정되었을 때 inlier의 수가 증가하는 것을 확인할 수 있다.



또한, 모델의 정확도 분포의 변화를 확인해볼 수 있다. 1번 이미지는 극단적인 값의 존재로 인해 데이터 분포를 정확하게 판단하기는 어렵지만, 2번과 3번을 비교해 평균적인 에러가 점차 증가함을 유추해볼 수 있다.



ransac_th이 너무 작으면 점들이 거의 선택되지 않아 좋지 않은 모델이 만들어진다. 또한, ransac_th이 너무 커도 이상한 점들이 선택되어 좋지 않은 모델이 만들어지게 된다.

2-3. RANSAC

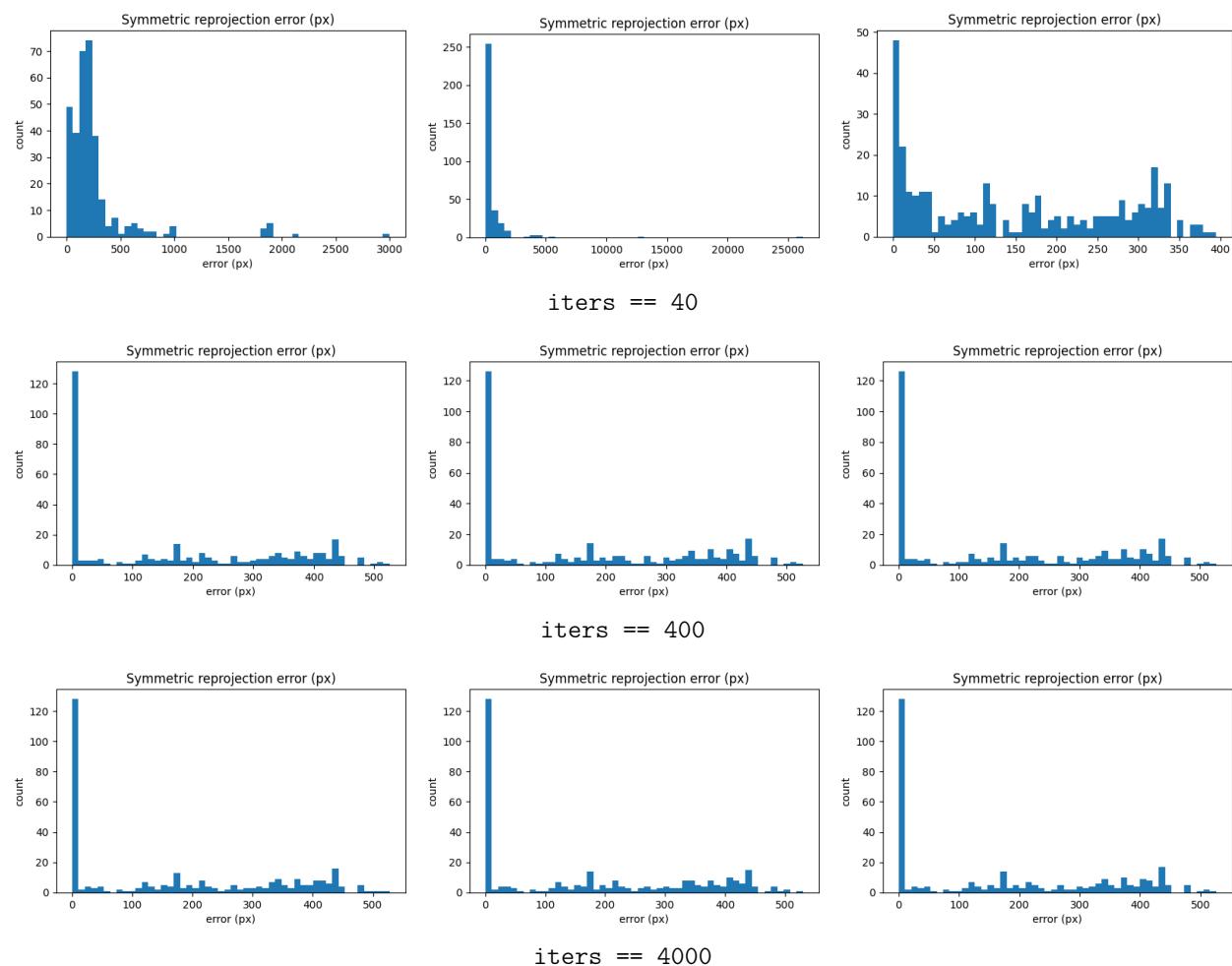
iters (Number of Iterations): When this parameter is increased,

- (1) reproducibility of the result: 증가한다.
- (2) robustness of the homography: 증가한다.
- (3) computational cost: 증가한다.

Explanation: RANSAC은 랜덤한 샘플을 뽑기 때문에 최적의 값을 찾기 위해서는 보다 많은 시행을 거쳐야 한다. 극단적으로 만약, 무한히 많은 시행을 거친다면 항상 최적의 값을 얻을 수 있을 것이고, 반면에 시행을 1번만 거친다면 최적의 값을 얻는 것은 불가능에 가까울 것이다. 즉, 시행 횟수가 늘어날수록, 보다 최적의 결과가 나오며 그 재현율은 높아질 것이다. 이와 더불어 homography의 견고함도 함께 증가하게 된다. 또한, 시행 횟수가 늘어나면 당연하게도 더 많은 연산을 수행하게 될 것이니 컴퓨팅 비용은 증가하게 된다.

Evidence:

아래는 순서대로 seed를 0, 1, 2로 고정한 뒤, iters 값을 변화하여 RANSAC을 수행한 결과이다.



iters가 작은 경우 변동이 크며, 재현성이 낮고, 반면에 iters가 큰 경우는 재현성이 크다는 것을 알 수 있다.

iters == 40: 10, 10, 15

iters == 400: 106, 105, 105

iters == 4000: 108, 105, 107

위는 inlier의 개수이다. iters가 작으면 10개 정도의 데이터만 inlier이지만 iters가 크면 100개 정도의 데이터가 inlier인 것을 확인해볼 수 있다. 이를 통해 homography의 robustness 또한 증가함을 확인해볼 수 있다.

2-4. Homography

DLT Normalization (in geom/dlt.py): Compare results with and without normalization. Which one gives lower reprojection error and better panorama stitching?

Explanation: 정규화를 하는 이유는 만약 좌표 스케일의 차이가 커서 행렬의 항들이 크게 차이나면, 수치적으로 불안정해지기에 이를 해결하기 위함이다. 즉, 정규화를 하게 되면 더 안정적인 파노라마 이미지를 얻을 수 있으며, reprojection error도 낮아진다.

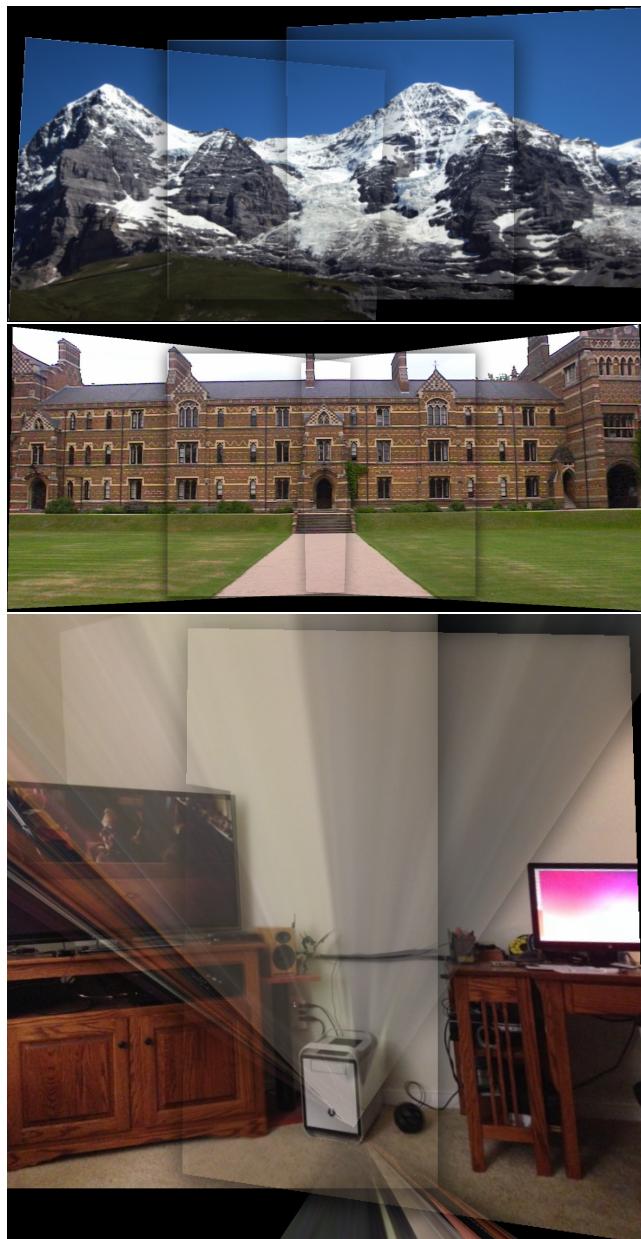
Evidence:

아래는 with vs without normalization된 행렬의 첫번째 행을 나열한 것이다. seed == 2, iters == 1000로 수행되었다.

```
with norm: [ 1.67663252 -1.28293771 1.           ... -2.84918875 2.1801627 -1.69935195]
without norm: [ 1.7300e+02 8.4000e+01 1.0000e+00 ... -5.3111e+04 -2.5788e+04 -3.0700e+02]
```

위 결과를 통해 matrix의 값을 normalize하면 수치적 불안정성이 작아짐을 확인해볼 수 있다. 다만, 수많은 테스트에도 불구하고 error가 0.01 이상으로 작아지는 경우나 파노라마 이미지가 가시적으로 더 안정적인 케이스는 찾지 못했는데, 이는 컴퓨터의 연산 정밀도가 너무 높고 이미지 좌표가 500 이내로 높은 차이가 없기 때문으로 추정된다.

2-5. Failure Case Identification



Explanation: 위의 결과를 보면, indoor 데이터셋의 파노라마 정합이 옳지 않게 수행된 것을 확인해볼 수 있다. 이는 다른 두 이미지셋과는 다르게 indoor 데이터셋은 비교적 가까운 거리의 물체와 배경을 찍은 이미지이기 때문에 단순한 homography 변환 1개로 이것을 매칭하기는 어렵기 때문이다. 사진 자체가 카메라를 제자리에서 회전하며 찍은 것이 아니라 카메라 자체를 약간식 좌우로 이동하면서 찍은 것이기에, 배경과 카메라의 거리 차이가 적은 indoor 데이터셋은 TV나 책상, 벽 같은 서로 다른 깊이에 있는 물체들이 카메라 시점에 따라 상대적인 위치가 달라져 homography 를 찾는데 문제가 발생하게 된다.