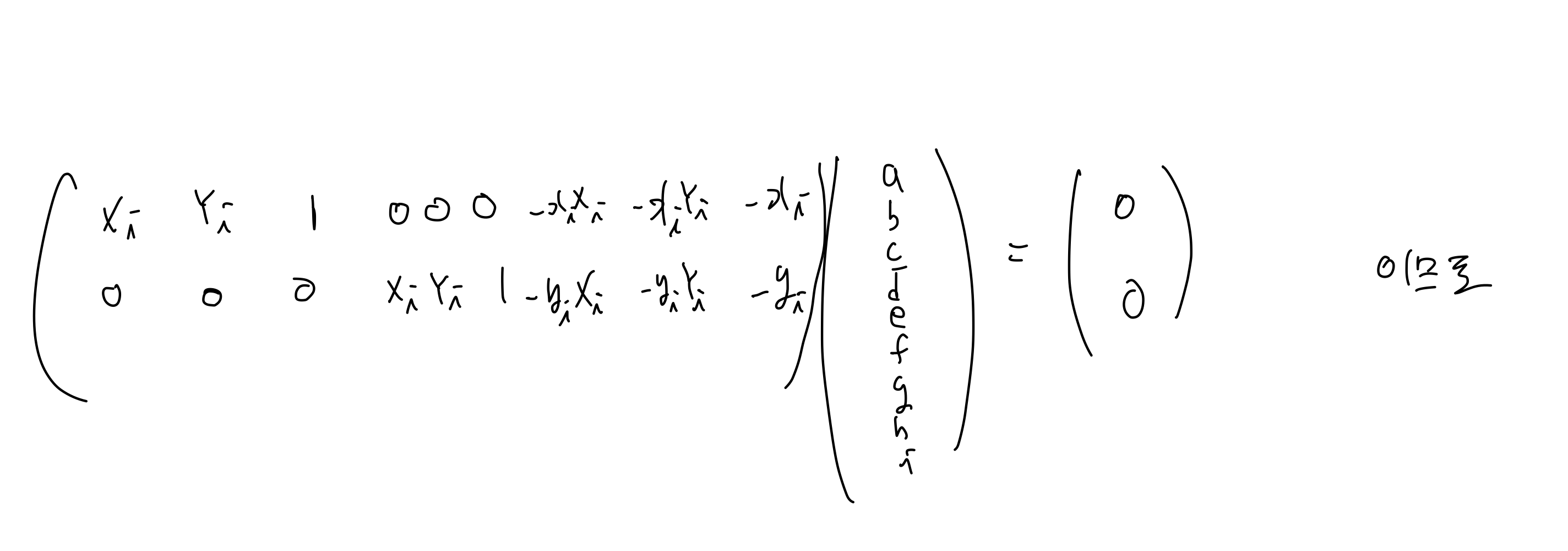
Part 1.

**1) H = compute\_h(p1,p2)**

이번 과제의 이론 부분의 문제 3.(a) 번의 수식

에 따라서 A 행렬 ,h 벡터를 생각하고 Ah = 0 을 Singular Value Decomposition(넘파이의 svd 함수이용)을 A에 적용해서 h 벡터를 찾고, 이를 3 X 3의 H 행렬로 변환하였다. 이때 || h || = 1이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**2) H = compute\_h\_norm(p1, p2)**

각 좌표(p1과 p2)에 각각의 최솟값을 빼고, 최댓값 – 최솟값의 크기를 나누어서 정규화하고, 좌표의 값들이 0 ~ 1 사이의 값을 가지도록 만들었다. 이 과정은 정규화 행렬을 사용했고, homogeneous 좌표계에서 이루어졌다. 이후 앞서 구현한 compute\_h를 호출하였다. 그 다음 compute\_h에서 반환되는 H행렬을 p1좌표를 올바르게 구할 수 있도록 보정해주었다.

Normalization을 포함한 Homograpy 행렬식은

인데, 올바른 p1을 구하려면

의 식이 되어야 한다.

따라서 compute\_h를 통해 받아온 H\_normalized에 을 좌측에 행렬을 곱해주고, 우측에는 행렬을 곱해주어서 p1을 정상적으로 구할 수 있는 H를 만들어 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Part 2.

**1) set\_cor\_mosiac()**

porto1, porto2 사진에서 대응하는 corner에 해당하는 픽셀을 육안으로 확인하고 6개의 쌍을 선택했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**2) warp\_image(igs\_in, igs\_ref, H)**

이 함수는 inverse warping 방법을 사용하여, 구현하였다. 그 과정에서 bilinear interpolation을 사용하였다. igs\_merge는 원래 크기의 4배가 되도록, 즉, 가로 길이 2배, 세로 길이 2배로 설정했다. igs\_warp는 igs\_ref, 즉 proto 2와 같은 크기(1200,1600)이 되도록 생성하였다.

먼저, Inverse warping을 위해 H의 역행렬을 구한 다음, homogeneous 좌표계에서 igs\_merge -> igs\_in로의 좌표를 찾아내었다. 이렇게 output의 좌표를 모두 고려함으로써, aliasing과 sub-sampling 이슈를 해결할 수 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

찾아낸 float 형태의 좌표를 정수 형태로 버림(floor)하여 변환한 뒤, 그 좌표의 근처의 3개의 점(r,r+1,c,c+1)과 이루는 사각형에서 수업시간에 배웠던 bilinear interpolation을 적용하여, igs\_warp와 igs\_merge에 들어갈 픽셀 값을 계산했다. 이 과정에서 igs\_merge에는 warp에 들어가는 이미지가 나중에 덮어쓰기 되므로, 과제 요구 조건처럼 igs\_warp이 우선적으로 merge에 나타난다.

Part 3.

**1) set\_cor\_rec()**

텍스트, 실내이(가) 표시된 사진

자동 생성된 설명

그림과 같이 초록선이 만나는 점 6개를 c\_in으로 고른 다음, c\_ref는 수평선 상의 가장 왼쪽 점의 y좌표를 기준으로 설정하였다.(x좌표는 c\_in과 동일)

텍스트, 스크린샷, 검은색, 화면이(가) 표시된 사진

자동 생성된 설명

**2) rectify(igs, p1, p2)**

함수의 파라미터 p1과 p2는 각각 c\_in(출발지)과 c\_ref(목적지)에 해당한다. 즉 compute\_h\_norm의 p1(목적지),p2(출발지)와 반대이므로 compute\_h\_norm(p2,p1)을 호출한다. 그리고 igs와 동일한 크기의 igs\_rec(output에 해당)을 생성한다.

이후에는 warp\_image함수처럼 igs\_ref -> igs\_in의 좌표를 inverse warping으로 찾아내고, bilinear interpolation을 수행한다.

텍스트이(가) 표시된 사진

자동 생성된 설명