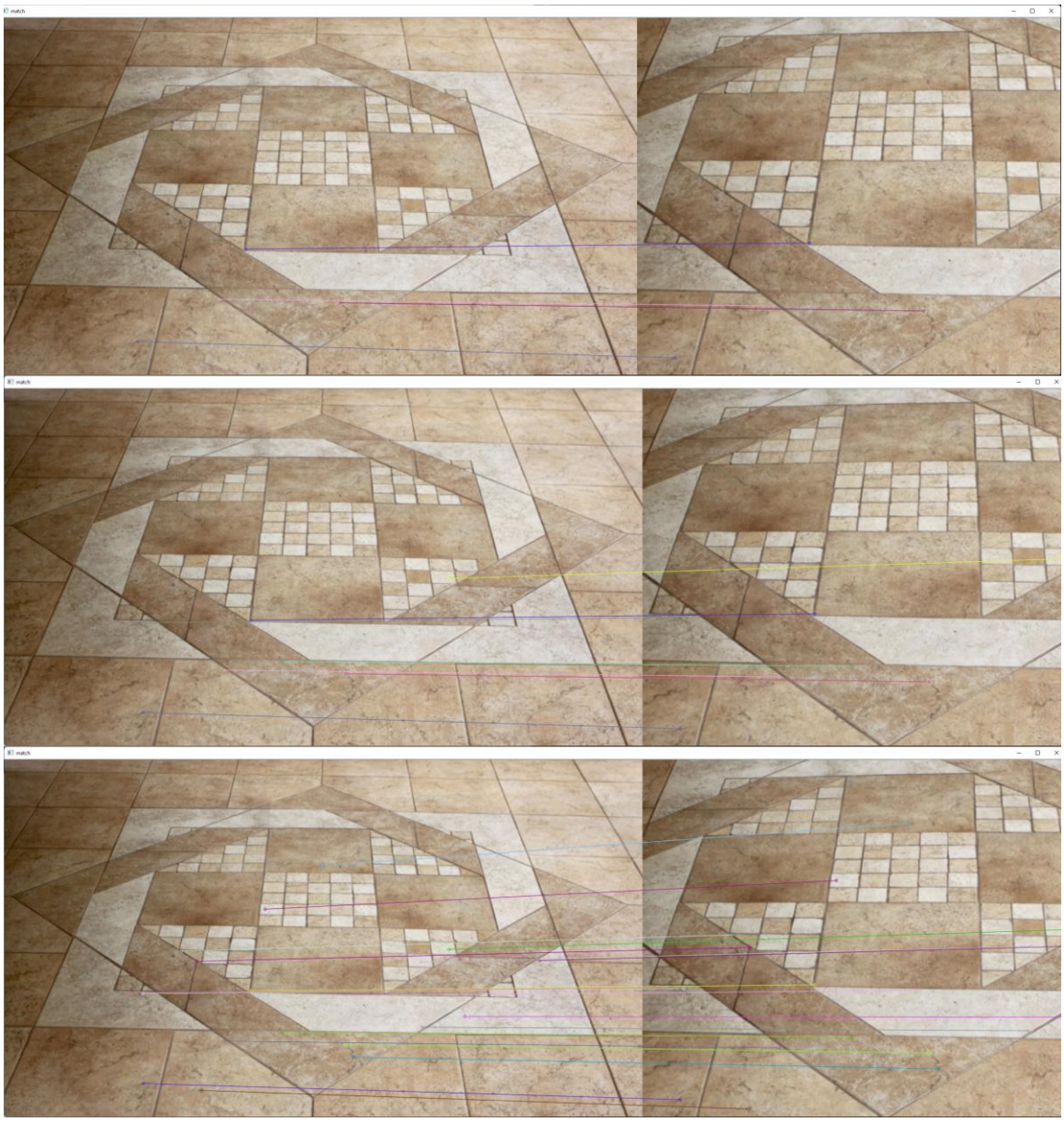
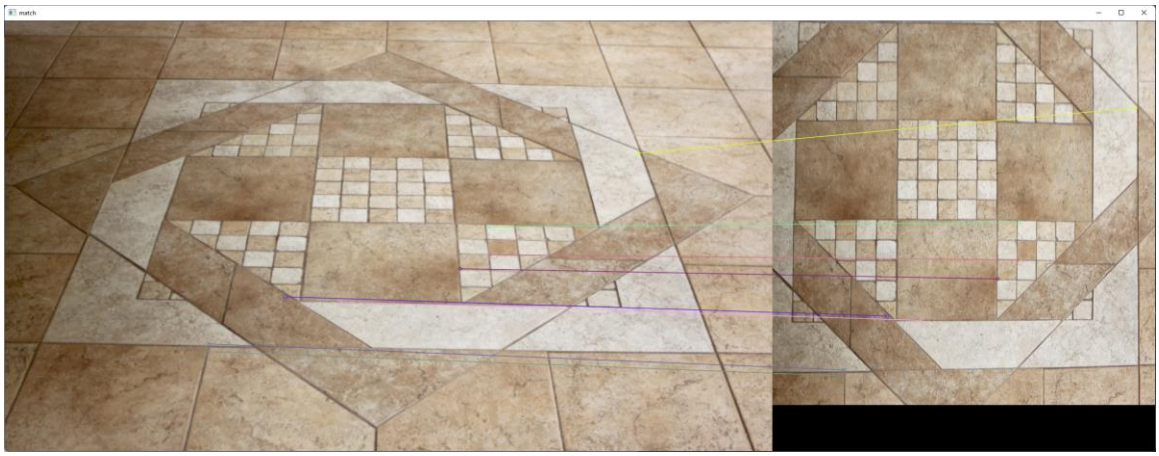
Problem 1: Homography estimation

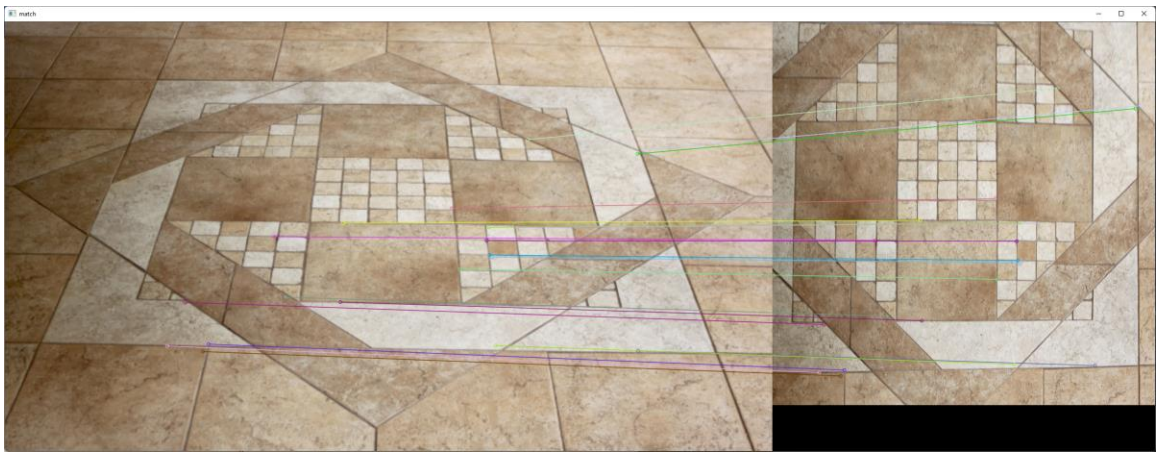
1. Screenshots:

- **Sample k correspondences (k = 4, 8, 20 or even more)**
  - **1-0 vs 1-1**



- **1-0 vs 1-2**





2. Compare the errors:

- I manually removed outliers which k<=20, so if you sample more than 20 correspondences by my command, the error will grow up explosively.

- **1 vs 1-1.png**

	1.py (DLT)	**1-2.py (Normalized DLT)**
Rate = 0.09 / samples = 4	7.596717625272696	**7.5967176224015445** **better**
Rate = 0.1 / samples = 8	0.9872960692534865	**0.9716819371453361** **better**
Rate = 0.1275 / samples = 20	0.45807307899389643	**0.44363343301583236** **better**

- **1 vs 1-2.png**

	1.py (DLT)	**1-2.py (Normalized DLT)**
Rate = 0.09 / samples = 4	**16.53022048005333** **better**	16.530220481168996
Rate = 0.1 / samples = 8	119.35659869293345	**7.8415230227354575** **better**
Rate = 0.1275 / samples = 20	**1.4056141163471134** **better**	2.451722890351554

~~**3. (Bonus)**~~

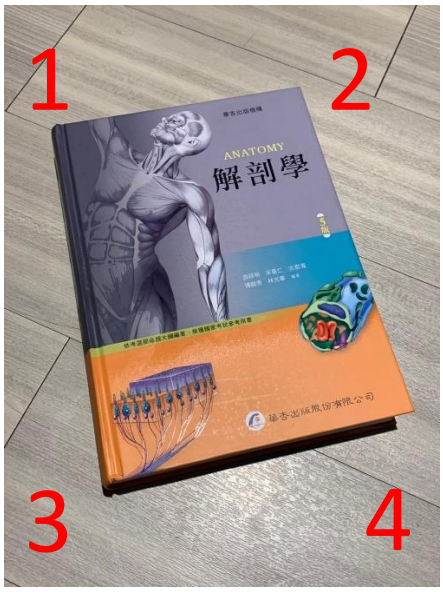
- ~~**My method**~~
- ~~**Screenshot: correspondences of other local features**~~
- ~~**Experimental comparisons**~~

4. Discussion

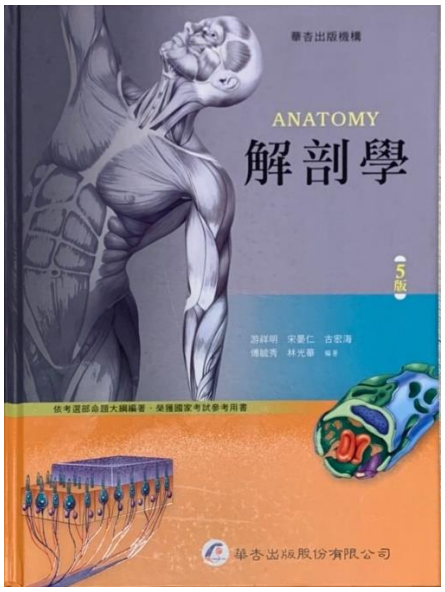
- The Normalized Method generally have better result, and the more feature points we choose the situation is more obvious.
- I found that the second comparation for 1-0 vs 1-2.png, the sift method can't find the accurate and same amount points as 1-0 vs 1-1.png, so when we choose too much point (larger than 20) the error will become bigger.
- However I was really confuse on the result of normal DLT / sample 8 points, I couldn't understand why it had such explosive error.

Problem 2: Document rectification

1. The input document image and Rectified results



2-0.jpg



2-1.jpg

2. Briefly explanation of method

1. I selected the corner by TA's mouse click function, select points order by Left-Up, Right-Up, Left-Down, Right-Down. (If you don't choose by this order the output rectified result image would have problem).
2. Then calculated the Homograph matrix and its inverse matrix which be used when doing the back warping by origin image.
3. And I use a map to memorize the coordinates from news to origins.
4. Finally, I did the warping with bilinear interpolation, which time efficiency is O(cols*rows) because it need to calculate for every pixel on new image and save it as 2-1.jpg.

Misc.

- **Python Environment: 3.6.13**
- **Package: OpenCV, Numpy, skimage, shapely, math**
- **Q1 (DLT):**
    - python 1.py [image path1] [image path2] [correspondence file path] [sample point number (4/8/20)]
- **Q1-2 (Normalize DLT):**
    - python 1-2.py [image path1] [image path2] [correspondence file path] [sample point number (4/8/20)]
    - e.g. python 1-2.py images/1-0.png images/1-2.png groundtruth_correspondences/correspondence_02.npy 4
- **Q2:**
    - python 2.py images/2-0.jpg