

Introduction to Computer Vision Assignment #3

2019-23191 박상욱

Image Stitching

본 프로젝트는 Python 3.6과 OpenCV 3.4.1을 이용하여 진행하였다.

1. Panoramic Pictures

Image Stitching에 사용한 이미지는 아래의 5장을 촬영하여 사용하였다. 5장의 이미지는 한 점에서 카메라를 돌려가며 촬영한 것이다. Stitching에 사용할 때는 이미지를 256 x 256으로 Resize하여 사용하였다.



그림 1. input images

2. Feature Matching

.OpenCV의 SIFT를 이용하여 이미지의 SIFT Feature를 찾고, 두 이미지 사이의 Feature를 Matching 하였다. I1과 I2, I2와 I3, I3와 I4, I4와 I5에 대한 각각의 Feature Matching 이미지는 아래와 같다.

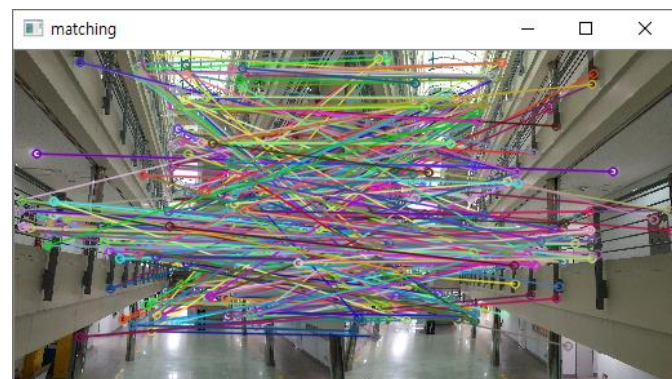
Match I1 & I2



Match I2 & I3



Match I3 & I4



Match I4 & I5

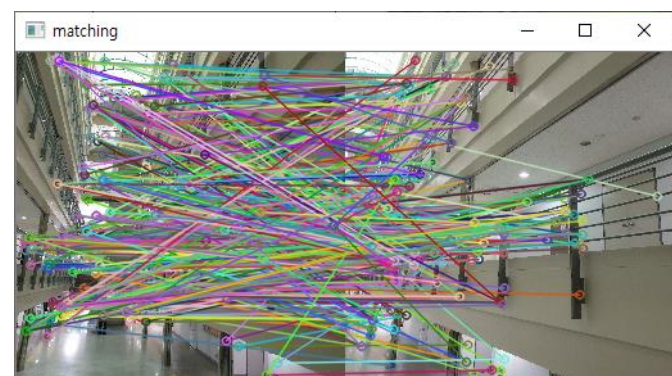


그림 2. Feature Matching between two images

3. Homography Estimation using RANSAC

두 이미지 사이에서 RANSAC과 DLT를 이용해 Homography Matrix를 추정한다. 이를 위해 Feature Matching을 통해 구한 Matching 된 Keypoints에서 임의로 4개의 pair를 Sampling하여 DLT를 이용해 Homography Matrix 후보를 구한다. 구해진 Matrix로 모든 Matching keypoints에 대해 d_{\perp} 를 계산한다. 이 때의 distance는 아래와 같이 Symmetric transfer error를 의미한다.

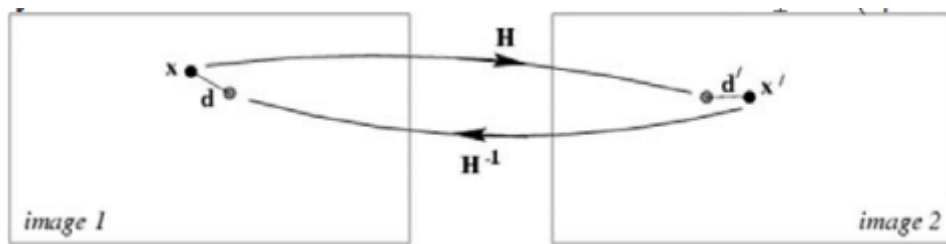


그림 3. Symmetric Transfer Error

d_{\perp} 를 이용해 inliers들의 개수를 세고, n번 동안 RANSAC을 이용해 가장 inlier를 많이 찾은 Homography Matrix 후보를 Homography Matrix로 구한다.

RANSAC과 DLT를 이용해 계산한 이미지 사이의 Homography matrix와 Total Symmetric transfer error는 아래와 같다.

표 1. Homography Matrix & Total Symmetric Transfer Error

| | Homography Matrix | Total Symmetric Transfer Error |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| I1 & I2 H12 | [[8.47739289e-01 5.90844560e-02 4.68581590e+01] [-1.06087139e-01 9.40285233e-01 1.40662913e+01] [-5.87443531e-04 -7.34418068e-06 1.00000000e+00]] | 24428.843635334622 |
| I2 & I3 H23 | [[7.36315252e-01 3.10315259e-02 7.92665021e+01] [-1.49117572e-01 8.97628179e-01 1.68036719e+01] [-9.72706769e-04 -6.38315962e-05 1.00000000e+00]] | 122453.57199838347 |
| I3 & I4 H34 | [[7.50014320e-01 5.48909953e-02 6.93243373e+01] [-1.68459261e-01 8.95407365e-01 1.36807120e+01] [-9.87272058e-04 -1.01681094e-04 1.00000000e+00]] | 5196.374758991344 |
| I4 & I5 H45 | [[7.83715355e-01 8.07396624e-02 6.66561953e+01] [-1.60387404e-01 9.27298484e-01 1.64033027e+01] [-9.19056579e-04 6.53444963e-05 1.00000000e+00]] | 48469.5608083801 |

4. Warping Images

RANSAC과 DLT를 이용해 각 이미지 사이의 Homography Matrix를 찾고, I3를 중심으로 Image Warping을 적용한다. I1, I2는 Matrix Inverse를 이용하여 Warping을 적용하고, I4, I5는 H34, H34*H45를 이용해 Warping을 적용하였다.



그림 4. Final warped result image

5. Math

$Ah = 0$ 인 h 중 $h^* = \min_h ||Ah||$ 를 찾고자 한다.

이 때 $h = 0$ 인 Solution은 우리가 찾고자 하는 것이 아니며, 찾게 되는 h 는 A 의 "null space"에 존재한다. 그리고 h^* 를 찾기 위해 $||Ah||^2$ 를 minimize하는 문제를 생각할 수 있다.

$$||Ah||^2 = (Ah)^T Ah = h^T A^T Ah$$

Largrange multiplier λ 를 이용해 아래의 식을 minimize하는 것과 동일하게 생각할 수 있다.

$$L(h) = h^T A^T Ah - \lambda(h^T h - 1)$$

미분을 통해 아래의 식을 만족하는 값이 minimize 하는 값이 된다.

$$A^T Ah - \lambda h = 0$$

따라서 λ 는 $A^T A$ 의 eigenvalue가 되고 $h = e_\lambda$ 가 대응하는 eigenvector가 된다. 이 때 $L(h) = \lambda$ 가 $\lambda = 0$ 일 때 $h = e_0$, 즉 가장 작은 값의 eigenvalue가 우리가 찾고자하는 h^* 가 된다.