

## Image Processing & Vision Homework 3: Panorama Stitching

예술공학대학 컴퓨터예술학부

20190807 민정우

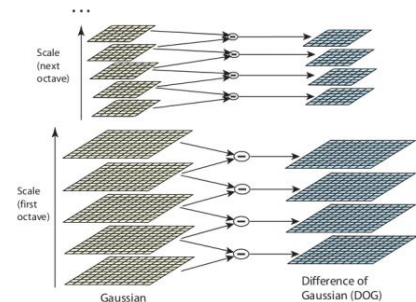
### 1. Image Stitching

이미지 스티칭(Image stitching)은 동일한 장면의 사진을 자연스럽게 붙여 한 장의 파노라마 이미지로 만드는 기법이다. 이미지 스티칭은 각 이미지의 특징점을 찾아 이를 매칭하고, 매칭된 특징점을 통해 이미지 간의 투시 변환 관계를 찾아 이미지를 변환하여 이어 붙이는 과정을 거친다.

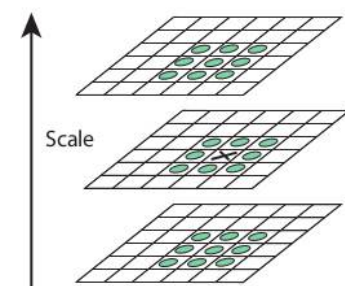
### 2. Find Feature point Matching

두 이미지를 매칭하기 위해 이미지의 특징점(Feature point)을 통해 그 이미지를 분석한다. 특징점은 일반적으로 Corner를 기준으로 검출하며 이동, 회전 및 크기 변화에 불변하는 특징점을 찾기 위해 SIFT 알고리즘을 사용한다.

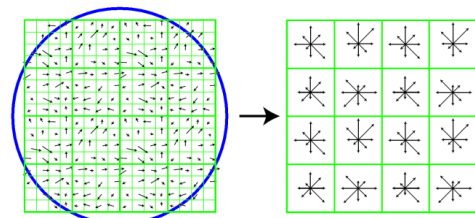
SIFT 알고리즘은 크기 변화에 불변하는 특징점을 찾기 위해 Scale Space를 제작한다. 이미지의 크기를 단계적으로 줄인 각 옥타브마다 가우시안 필터를 표준 편차를 다르게 적용해 여러 개의 블러 이미지들을 만든다. 이후 인접한 두 블러 이미지의 차를 계산하는 DoG(Difference of Gaussian) 연산을 통해 DoG 이미지들을 출력한다.



계산한 DoG 이미지에서 키포인트를 감지한다. 키포인트는 선택한 DoG 이미지와 scale이 다른 인접한 두 DoG 이미지를 통해 계산하며, 검사하는 픽셀과 그 주위 26개의 픽셀과 비교해 극대값이거나 극소값일 때 해당 픽셀을 키포인트로 선정한다. 이후 임계값보다 크기가 작거나 edge에서 검출되지 않은 불안정한 키포인트를 제거한다.



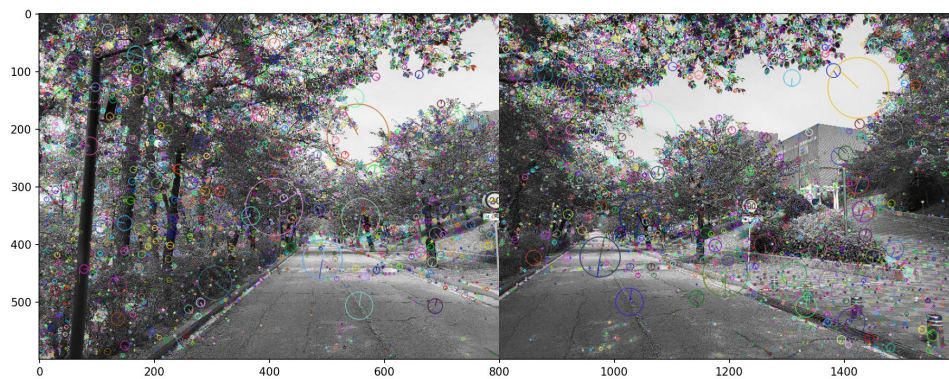
이후 키포인트가 회전 변화에 불변하도록 이미지 그라디언트를 통해 키포인트에 방향을 할당한다. 이후 방향 히스토그램을 제작하는데, 키포인트 주위 영역을 4x4 구역으로 분할하고 각 구역마다 8개 방향의 히스토그램을 구해 키포인트 하나당 128개의 실수 정보를 가진 기술자를 생성한다. 이후 기술자는 정규화를 통해 밝기 의존성을 해결한다.



**SIFT(img) -> kp, des**

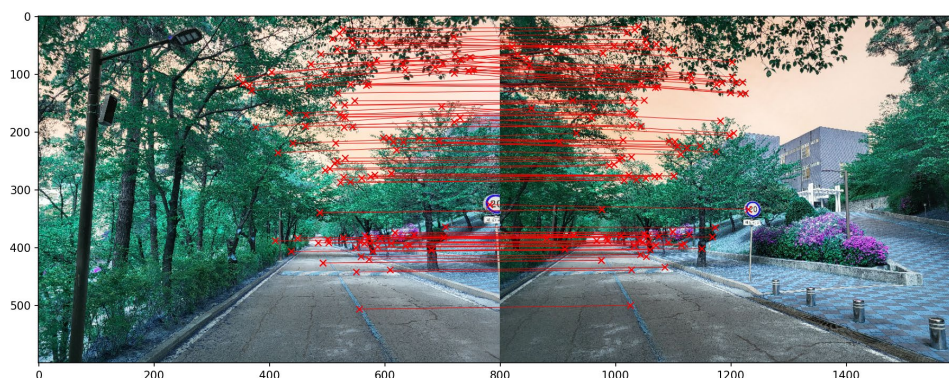
SIFT 알고리즘의 특허는 2020년에 만료됐기 때문에 OpenCV 라이브러리에서 무료로 SIFT 연산을 제공한다. 입력받은 이미지(img)에 OpenCV 라이브러리에서 제공하는 SIFT 연산을 수행하고

그 결과 키포인트(keypoint, kp)와 기술자(descriptor, des) 배열을 반환한다.



**siftMatch(kp1, des1, img1, kp2, des2, img2, threshold) -> matches**

두 이미지의 공통된 부분을 매칭하기 위해 SIFT 연산으로 계산된 두 이미지의 키포인트와 기술자를 비교해 유사한 기술자를 파악한다. cv2.BFMatcher 클래스를 통해 두 이미지의 기술자를 Brute-Force 알고리즘을 통해 하나씩 비교해가며 matcher.knnMatch() 함수를 통해 각 기술자마다 가장 유사한  $k(=2)$ 개의 기술자를 찾는다. 이후 D.Lowe가 고안한 비율 테스트로 적합한 매칭을 파악하는데, 해당 기술자와 가장 가까운 기술자의 거리와 두 번째 가까운 기술자의 거리를 비교해 충분한 차이가 없다면 해당 매칭을 무시하고, 충분한 차이를 가지면 해당 매칭을 적합한 매칭으로 간주한다.



### 3. Find the Perspective Transformation

두 이미지의 변환 관계를 파악하기 위해 RANSAC 알고리즘을 통해 두 이미지에서 매칭된 특징점 쌍에 가장 유사한 변환 모델을 찾는다.

RANSAC 알고리즘은 가장 적합한 모델을 찾기 위해 무작위로 샘플 데이터를 뽑는다. 무작위로 추출한 샘플 데이터들을 만족하는 모델을 계산한 뒤 해당 모델에 일치하는 데이터의 개수를 센다. 이 과정을 여러 번 반복해 가장 많은 데이터가 만족하는 모델을 결과로 반환한다.

**randomPoint(matches, k\_samples) -> rand\_pnt**

입력받은 매칭쌍(matches)에서 무작위로 특정 개수( $k\_sample$ )의 쌍을 뽑아 반환한다.

## homography(pairs) -> H

무작위로 추출한 매칭쌍들이 만족하는 변환 행렬을 계산해 반환한다. 매칭된 두 특징점  $p_1(x_1, y_1)$ ,  $p_2(x_2, y_2)$ 와 두 특징점의 변환 관계를 만족하는 변환 행렬  $H$ 와의 관계를 다음 식으로 나타낼 수 있다.

$$p_2 = Hp_1, \quad \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

다음 행렬을 연립 방정식으로 분할하면 다음과 같다.

$$h_1x_1 + h_2y_1 + h_3 = x_2$$

$$h_4x_1 + h_5y_1 + h_6 = y_2$$

$$h_7x_1 + h_8y_1 + h_9 = 1$$

3번째 식에서  $h_7x_1 + h_8y_1 + h_9$ 가 1이므로 1, 2번째 식의 우변에  $h_7x_1 + h_8y_1 + h_9$ 를 곱한다.

$$h_1x_1 + h_2y_1 + h_3 = x_2(h_7x_1 + h_8y_1 + h_9), \quad x_1h_1 + y_1h_2 + h_3 - x_1x_2h_7 - y_1x_2h_8 - x_2h_9 = 0$$

$$h_4x_1 + h_5y_1 + h_6 = y_2(h_7x_1 + h_8y_1 + h_9), \quad x_1h_4 + y_1h_5 + h_6 - x_1y_2h_7 - y_1y_2h_8 - y_2h_9 = 0$$

다른 모든 매칭쌍에 대해 위 과정을 거쳐 구한 모든 관계식을 모으고 변환 행렬  $H$ 와의 관계식으로 변환한다.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_2 & -y_1x_2 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_2 & -y_1y_2 & -y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_4 & -y_3x_4 & -x_4 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y_4 & -y_3y_4 & -y_4 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_{n+1} & -y_nx_{n+1} & -x_{n+1} \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_{n+1} & -y_ny_{n+1} & -y_{n+1} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

위 식의 왼쪽 행렬에 특이값 분해를 수행한다.

$$M = U\Sigma V^T$$

구하고자 하는 변환 행렬은 특이값 분해의 결과로 얻은 행렬  $V$ 의 맨 오른쪽 열벡터에서 얻을 수 있으며 이 열벡터를 3x3 행렬로 변환한 후 3행 3열의 값이 1이 되도록 표준화한다.

## getError(matches, H) -> errors

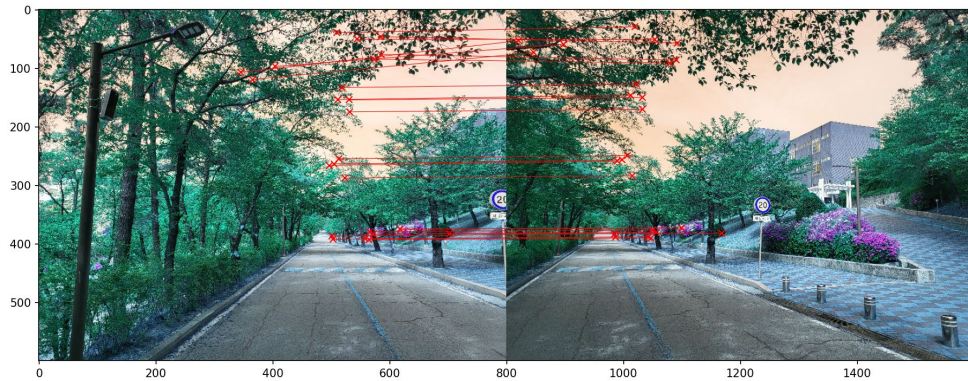
임의의 매칭쌍으로부터 계산된 변환 모델을 모든 매칭쌍에 대입해 실제값과 변환 모델 적응값의 오차를 반환한다.

## ransac(matches, k\_sample, threshold, iters) -> best\_inliers, best\_H

randomPoint() 함수를 통해 매칭된 특징점 쌍(matches) 중에서 임의의 개수(k\_sample)만큼 무작위



로 선별한다. 무작위로 선별된 매칭쌍을 통해 homography() 함수로 변환 모델을 추출한 뒤 getError() 함수로 해당 변환 모델을 적용한 오차를 계산한다. 설정한 임계값(threshold)보다 작은 오차값을 가진 특징점을 세어 inlier를 검출하고 이를 iters만큼 반복해 가장 많은 inlier를 가진 변환 모델(best\_H)을 해당 모델로 추출한 inlier들(best\_inliers)과 함께 반환한다.



#### 4. Stitch Image

SIFT 알고리즘으로 각 이미지의 특징점을 추출한 후 양쪽 이미지의 특징점을 매칭한 다음 RANSAC 알고리즘으로 특정한 원근 변환 모델을 통해 이미지를 원근 변환한 후 서로 이어 붙여 파노라마 이미지를 제작한다.

**stitchImg(left, right, H) -> stitch\_image**

RANSAC 알고리즘으로 추출한 변환 모델(H)을 통해 입력받은 왼쪽 이미지(left)를 원근 변환한 후 오른쪽 이미지(right)와 결합해 파노라마 이미지(stitch\_image)를 제작한다.

