

(1) 코드 설명

<SIFT.cpp>는 두 개의 이미지에서 SIFT를 사용하여 키포인트를 검출하고, 디스크립터를 추출한 다음, 두 이미지간의 매칭된 키포인트를 찾는 과정을 수행하는 코드입니다. 키포인트는 동그라미로 표시하고 매칭된 키포인트들은 빨간색 선으로 연결하여 시각화하였고 이를 창에 띄워지도록 하였습니다. input1, 2의 키포인트, 매칭된 키포인트의 개수 또한 출력됩니다. 세부 설명은 주석으로 넣어두었습니다.

(2) 사용된 개념

- SIFT(Scale-Invariant Feature Transform)

: not robust against scale changes

1) scale space extrema detection

2) key point localization

3) orientation assignment

4) descriptor construction

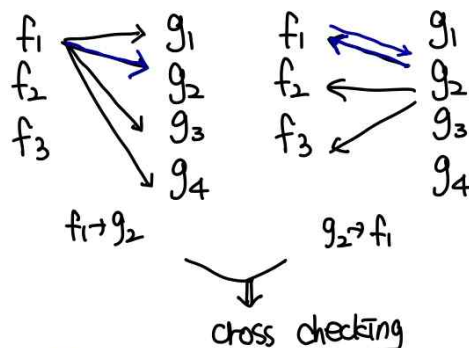
- euclidDistance(): 두 벡터 사이의 유클리드 거리를 계산하는 함수 -> 이 함수는 매칭 절차에서 이웃쌍을 검증하는데 사용됨

- nearestNeighbor(): 주어진 디스크립터와 키포인트들 중에서 가장 가까운 이웃의 인덱스를 찾는 함수 -> 매칭 절차에서 가장 가까운 이웃을 찾는 데 사용됨

- findPairs(): 주어진 두 개의 이미지에 대해 키포인트와 디스크립터를 입력으로 받아 매칭된 키포인트 쌍을 찾는 함수

-> crossCheck가 true이면 이웃쌍을 검증할 때 상호 검증을 수행함

* cross-checking



상호검증하여 서로 매칭되면 reliable

$k = \arg \min \text{dist}(f_i, g_j) \rightarrow \text{NN}(f_i) = g_k$

$l = \arg \min \text{dist}(f_i, g_k) \rightarrow \text{NN}(g_k) = f_l$

$i=l$ 이면 reliable, 아니면 unreliable

-> ratio_threshold가 true이면 이웃쌍을 검증할 때 ratio-based threshold을 사용하여 검증함

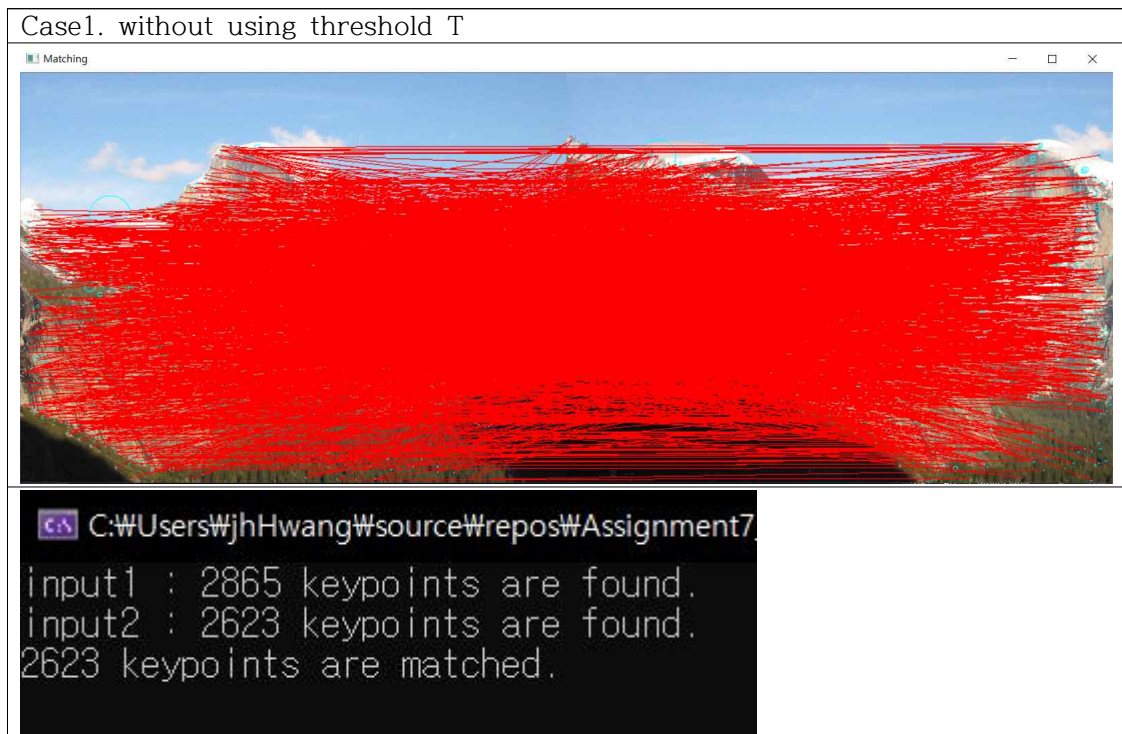
*threshold ratio

$k_1 = \arg \min \text{dist}(f_i, g_j)$

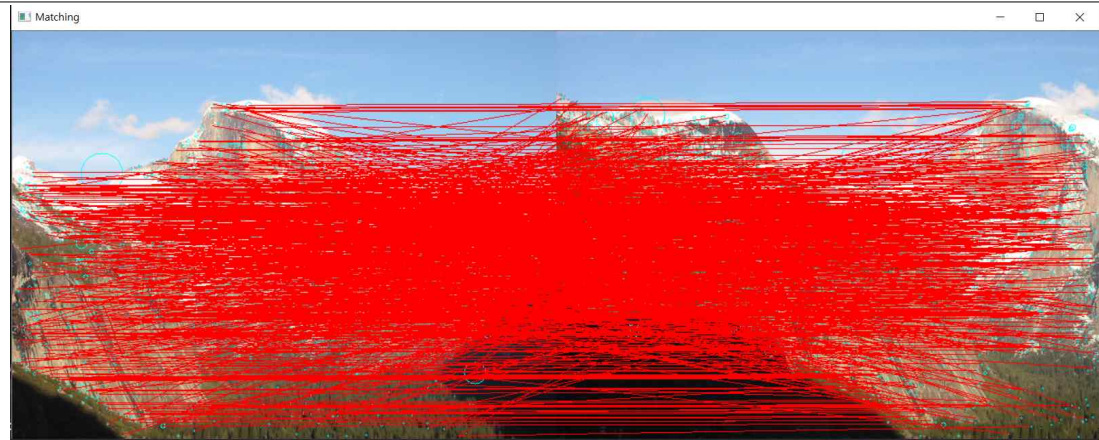
$k_2 = \arg \text{second min dist}(f_i, g_j)$

$$\frac{\text{dist}(f_i, g_{k_1})}{\text{dist}(f_i, g_{k_2})} < T_r \Rightarrow \text{NN}(f_i) = g_{k_1}$$

(3) 수행 결과

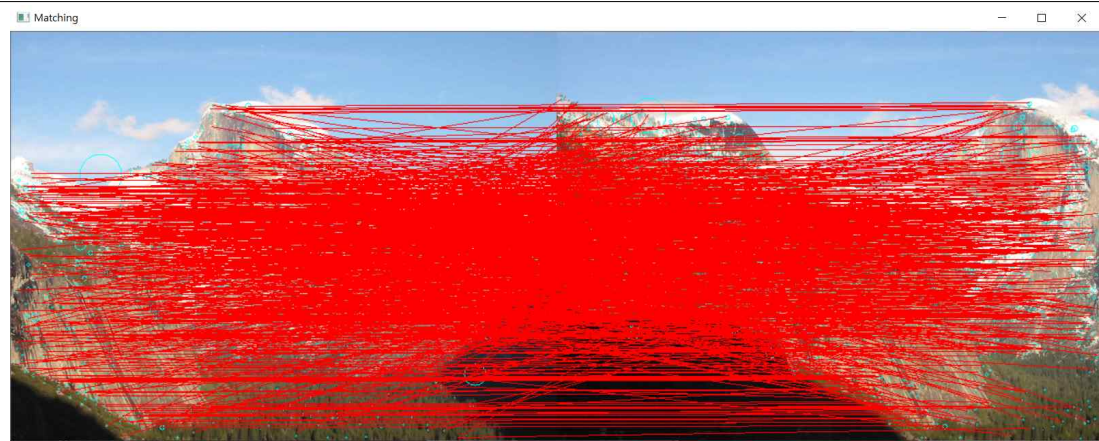


Case2. using cross-checking



```
C:\Users\WjhHwang\source\repos\Assignment7_217  
input1 : 2865 keypoints are found.  
input2 : 2623 keypoints are found.  
1050 keypoints are matched.
```

Case3. using both cross-checking and ratio-based thresholding



```
C:\Users\WjhHwang\source\repos\Assignment7_  
input1 : 2865 keypoints are found.  
input2 : 2623 keypoints are found.  
1050 keypoints are matched.
```