메디치소프트 기술연구소

CH44. OpenCV: FLANN 기반 이미지 특성 매칭

MEDICISOFT

44. FLANN 기반 이미지 특성 매칭

- 44.1 테스트 환경

순 번	제 목	설치 버전
1	운영체제	Windows 10 64bit
2	프로그래밍언어	Python3.6.5

MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

- 이전 chapter 43 에서는 두 이미지의 특성을 매칭하기 위해 Brute-Force 매칭 방법을 활용했다. 방법은 각 이미지에서 검출한 특성들을 전수 조사하여 매칭하는 방법이므로 이미지 사이즈가 크게 되면 성능적인 문제가 발생할 수 있다.

- FLANN은 Fast Library for Approximate Nearest Neighbors의 약자로 큰 이미지에서 특성들을 매칭할 때 성능을 위해 최적화된 라이브러리 모음이다. FLANN 매칭은 큰 이미지에 대해 BF 매칭에 보다 빠르게 동작한다.

MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

- 다음은 44-01 example.py 예제의 code이다.

```
import numpy as numpy
import cv2
def FLANN(factor):
  img1 = cv2.imread('images/mybook1.jpg', cv2.IMREAD_GRAYSCALE)
  img2 = cv2.imread('images/mybook1-1.jpg', cv2.IMREAD_GRAYSCALE)
  res = None
  sift = cv2.xfeatures2d.SIFT_create()
  kp1, des1 = sift.detectAndCompute(img1, None)
  kp2, des2 = sift.detectAndCompute(img2, None)
  FLANN_INDEX_KDTREE = 0
  index_params = dict(altorithm=FLANN_INDEX_KDTREE, trees=5)
  search_params = dict(checks=50)
```


MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

- 다음은 44-01 example.py 예제의 code이다.

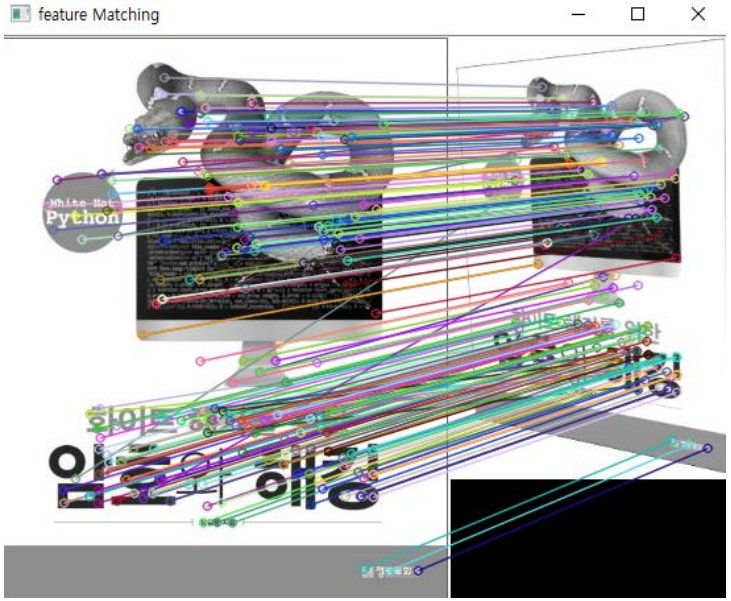
```
flann = cv2.FlannBasedMatcher(index_params, search_params)
matches = flann.knnMatch(des1, des2, k=2)
good =[]
for m, n in matches:
    if m.distance < factor * n.distance:
        good.append(m)
res = cv2. drawMatches(img1, kp1, img2, kp2, good, res, flags = 2)
cv2.imshow('feature Matching', res)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

FLANN(0.7)

MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

- 다음 그림은 44-01 example.py를 실행한 결과 생성되는 image 사진이다.
  뒷 장부터 44-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 44-01 example.py의 실행 결과

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

>> LINE 13~15)

```
FLANN_INDEX_KDTREE = 0
index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
search_params = dict(checks=50)
```

- FLANN 기반 매칭을 위해 두 개의 사전 자료형의 인자가 필요한데, 하나는 indexParams이고, 나머지 하나는 searchParams이다.
- SIFT와 SURF를 활용하는 경우 indexParams는 예제 코드와 같이 사전 자료를 생성하면 된다.

MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

>> LINE 17~18)

```
flann = cv2.FlannBasedMatcher(index_params,search_params)
imatches = flann.knnMatch(des1,des2,k=2)
```

- FLANN 기반 매칭 객체를 앞에서 구성한 사전 자료 형태의 인자를 이용해 생성하고, KNN(K-Nearesst Neighbors) 매칭을 수행한다.
- KNN 매칭은 k=2로 설정된 순위만큼 리턴하므로 2번째로 가까운 매칭 결과까지 리턴한다.
- KNN 매칭을 하는 이유는 리턴한 결과를 사용자가 선택하여 다룰 수 있기 때문이다. k=2로 설정하였으므로 matches는 (1순위 매칭 결과, 2순위 매칭 결과) 가 멤버인 리스트가 된다.

MEDICISOFT

▪ 44.2 [44-01 example.py] - FLANN 기반 이미지 특성 매칭

```
>> LINE 20~23)
good = []
for m,n in matches:
    if m.distance < factor*n.distance:
    good.append(m)
```

- matches의 각 멤버에서 1순위 매칭 결과가 2순위 매칭 결과의 factor로 주어진 비율보다 더 가까운 값만을 취한다.

  factor의 값은 0.7이므로 1순위 매칭 결과가 2순위 매칭 결과의 0.7배보다 더 가까운 값만을 취한다.

MEDICISOFT