**OSP: Lec8-Feature Descriptor**

1871098 손수민

1. SIFT

본 예제는 OpenCv에서 제공하는 함수 SIFT descriptor와 함께 cross-checking, ratio-based thresholding을 구현해 같이 결과를 확인해보는 예제이다.

먼저 두 벡터 사이 유클리디안 거리(dist)를 구해주는 함수이다. 두 벡터의 차이의 제곱을 더해주고 이를 반환해주는 간단한 함수이다.

(에러가 발생해 <float>로 바꿔주었다.)

텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 nn을 구현하는 함수이다. For 문을 돌며 descriptors의 행을 불러와 v에 저장하고 이와 vec의 dist를 구해 최소 distance인 minDist를 구해주고 이의 인덱스인 i를 neighbor를 반환해면 된다. 하지만 밑의 ratio-based thresholding에서 두번째로 작은 값도 필요하므로 bool second 파라미터를 추가해주어 true이면 두번째로 작은 인덱스를 반환하도록 구현하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

만약 두번째로 구한 최소값이 위의 neighbor와 같다면 continue로 무시하도록 하였다.

다음은 매치된 쌍을 찾는 findPairs 함수이다. 마찬가지로 for문을 돌며 행에 접근해준다. 그 후 해당 행과 descriptors와의 최소거리인 인덱스와 두번째로 작은 인덱스를 각각 nn, nn2에 저장해준다. 그리고 만약 ratio\_threshold가 true로 설정되었다면 nn,nn2 와 f\_i 사이의 거리를 구해 나눠주고 이 값이 RATIO\_THR보다 크다면 continue로 무시해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

또한 crossCheck가 true로 되어 있다면 매칭된 값이 역으로도 성립하는지 (L=i)인지 확인해주고 아니라면 continue로 무시해주도록 구현하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이에 대한 feature matching 결과는 case에 따라 다음과 같다.

Case 1: cross-checking false / ratio-based thresholding false

텍스트, 하늘, 빨간색, 일몰이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Case 2: cross-checking true / ratio-based thresholding false

텍스트, 하늘, 빨간색이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Case 3: cross-checking true / ratio-based thresholding true

텍스트, 옥외설치물이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Case1보다 case2가 매칭된 키포인트가 적었고, case2보다 case3가 매칭된 키포인트가 더 적은 것을 볼 수 있었다.

1. Simple Application : Feature2D+Homography to find a known object

https://docs.opencv.org/3.4.1/d7/dff/tutorial\_feature\_homography.html

텍스트이(가) 표시된 사진

자동 생성된 설명

우선 해당 코드는 ./SURF\_descriptor <img1> <img2>로 실행할 수 있다.

따라서 앞의 <img1>로 받아온 이미지의 경로를 imread로 읽어 Mat 형식으로 img\_object에 저장한다. 또한 <img2>로 받아온 이미지의 경로도 마찬가지로 img\_scene으로 저장한다. 만약 두 이미지 중 하나라도 존재하지 않는다면 오류메시지를 출력한다. 여기서 img\_object가 찾을 object가 있는 이미지, img\_scene이 해당 이미지에서 object를 찾을 이미지이다.

첫번째 단계는 이미지에서 키포인트를 탐지하고 SURF에서 descriptors를 추출하는 과정이다. 여기서 minHessian =400으로 설정해 create 하였는데 이는 기본 파라미터 hessianThreshold를 의미한다. 그 후 detector를 이용해 각각의 이미지에서 키포인트를 탐지하고 descriptors를 compute해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

두번째 단계는 FLANN matcher를 사용해 descriptor vector를 매칭시켜주는 과정이다. 결과는 matches에 저장이된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 for문을 돌며 저장된 matches에 접근해 최대, 최소 거리를 구해 프린트시켜준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 다시 for문을 돌며 거리가 3\*min\_dist보다 작은 것만 good\_matches에 push\_back시켜준다. 그 후 drawMatches를 통해 두 이미지에서의 keypoints의 매칭을 찾아 그려준다. 이는 img\_matches에 저장이 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 다시 for문을 돌며 good\_matches에 접근해 각 매칭점을 obj, scene에 저장한 후 findHomography를 사용해 둘 사이의 perspective transformation 을 구해 H에 저장한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 image1에서 corners를 가져와 perspectiveTransfrom를 실행해 해당 벡터에 대해 H를 이용한 perspective Transform을 실행해 이를 scene\_corner에 저장한다.

텍스트, 실내, 은색, 여러개이(가) 표시된 사진

자동 생성된 설명

그 후 이미지 1과 이미지2에 대한 선을 그려 window창에 띄어 보여준다.