

Project 2: Local Feature (Harris Corner) Detector

연결되는 Local Feature Matching 과제 2-3 중 1 부

요약

- 마감: 2020.05.07
- Part 1: Code – Implement function `get_interest_points()` in file `Student.py`
 - `Main.py`, `Student.py` 파일의 주석 설명문을 자세히 읽고 작업하세요.
- Part 2: Report
 - 구현된 각 세부 과정에 대한 관련 이론을 요약하여 구현된 코드와 관련지어 설명하십시오.
 - 기본 제공된 Data 폴더의 예시 영상들에 대해 도출된 결과를 분석하십시오.

개요

지역적 특징점 local feature 정합 matching 알고리즘(강의자료 6,7,8 장)을 구현하여 실제 장면에 대한 다시점 영상 multiple views of real-world scenes 을 정합하고자 한다. 해당 문제의 각 세부 과정에 대해 각각 지금까지 수백편의 컴퓨터비전 computer vision 논문이 발표되었다. 우리는 과제 2,3 에 걸쳐 Harris corner detector 와 단순화된 버전의 SIFT simplified version of SIFT 알고리즘을 구현하고자 한다. 하지만, 원한다면 얼마든지 다른 알고리즘을 구현하셔도 된다.(추가 점수 부여 가능)

주어진 과제 Task: 지역적 특징점 정합 local feature matching 의 세 가지 주요 단계 the three major steps 를 구현 implement (coding)할 것:

- **검출 Detection** in the `get_interest_points` function in `student.py`. Please implement the Harris corner detector (Szeliski 4.1.1, Algorithm 4.1). (이번 과제 2)
- **기술자 생성 Description** in the `get_features` function, also in `student.py`. 유사-SIFT (SIFT-like) 특징점 기술자 local feature descriptor (Szeliski 4.1.2)를 구현하십시오. 완전-SIFT까지 구현하지는 않아도 됨! 요구 기준을 충족시킬 때까지 디테일을 추가하여 보시오. 정합 파이프라인 matching pipeline 을 빠르게 테스트하기 위해서는 정규화된 패치 normalized patch 를 기술자로 우선 사용해 보세요. (다음 과제 3)
- **정합 Matching** in the `match_features` function of `student.py`. 특징점 정합 matching local features 방법론 중 "비율 테스트 ratio test" 또는 "최소거리 비율테스트 nearest neighbor distance ratio test"을 구현해 보시오 (Szeliski 4.1.3; equation 4.18 in particular). (다음 과제 3)

유용할 수 있는 함수들 Potentially useful functions: 기존의 필터링 함수들 및, `zip()`, `skimage.measure.regionprops()`, `skimage.feature.peak_local_max()`, `numpy.arctan2()` 등.

유용할 수 있는 함수들 라이브러리들 Potentially useful libraries: 다양한 필터링 함수를 제공하는 `skimage.filters.x()` 또는 `scipy.ndimage.filters.x()`, 영상의 미분을 연산하는 `np.gradient()`, 원소별 *element-wise binning* 기능을 제공하는 `np.digitize()` 등. 그리고 이전 과제에서 각자 구현한 함수 등.

사용 불가 함수들 Forbidden functions: `skimage.feature.daisy()`, `skimage.feature.ORB()` 등 특징점 기술자를 바로 도출해주는 함수들, `skimage.feature.corner_harris()` 등 특징점을 바로 검출해 주는 함수들, 그리고 히스토그램을 연산 *computes histograms* 해 주는 함수들, `sklearn.neighbors.NearestNeighbors()` 등 가장 가까운 이웃 비율 *nearest neighbor ratios* 을 계산해 주는 함수들, 그리고 `scipy.spatial.distance.cdist()` 등 벡터의 배열 간 거리를 계산해 주는 함수들 등 (벡터 간 거리를 계산하는 함수는 추후 제공될 가이드를 통해 직접 계산해 보세요.). 불확실한 경우 반드시 질문을 해 주세요.

코드 실행하기 Running the Code

`main.py` 파일을 실행하면 됨. 터미널/커멘드 창에서 `python` 키워드로 `python interpreter` 를 실행시키고, 그 뒤에 실행시킬 `python` 파일명을 제공하면 됨. 아울러 커멘드 라인 인자 *command-line argument* 에 '-p'을 추가하면 데이터셋을 로드 할 수 있음. 예를 들어, `$ python main.py -p notre_dame.` `main.py` 에 더 상세한 내용을 참고하십시오.