NAVER LABS Mapping & Localization Challenge Outdoor - 2nd place

Hyunyoung Jung

gusdud1500@snu.ac.kr

CMALAB, Seoul National University

1. Overview

정확한 **Localization**을 구현하기 위하여 전체적인 과정은 **[1]**의 파이프라인을 따르고 있다. 테스트 시 쿼리 이미지가 주어졌을 때 해당 이미지에 대하여 **Global descriptor**를 계산한 후, 데이터베이스 이미지들을 검색하여 쿼리 이미지와 가장 유사한 **N**개의 이미지를 고른다.**(Section. 2)** 이후에 **Local descriptor**를 사용하여 이전 과정에서 선택된 데이터베이스 이미지와 쿼리 이미지 간의 **correspondence points**를 검출 후 **Semantic & Geometric verification** 단계를 통해 부정확한 매칭점들을 제거한다.**(Section. 3)** 이후 데이터베이스 이미지와 인접한 타임스탬프를 갖는 **Lidar points**들을 **projection**한 후 해당 매칭점들과 가장 가까운 거리에 존재하는 **lidar point**를 해당 **key point**에 대한 **3D** 포인트로 설정을 한다. 이렇게 선택된 **3D Point**들과 매칭된 쿼리 이미지 포인트 간에 **perspective-n-point** 알고리즘을 통해 최종 **pose**를 계산한다.**(Section. 4)**

2. Image Retrieval

먼저 데이터베이스 이미지들에 주어진 camera pose(translation)정보를 기반으로 K-nearest neighbor 알고리즘을 적용하여 2개의 cluster로 분류하였다. 이후 기 학습된 모델 [2]을 통해 데이터베이스 및 쿼리 이미지에 대한 Global descriptor를 계산한 후 각각의 클러스터에서 200장의 가장 유사한 이미지를 선택한다. 이 때 사용한 쿼리 이미지의 경우, 추후 마지막 단계에서 sequence 정보를 이용하기 위하여 test sequence에 주어진 (0, 9, 19, 29, 39, 49) 번째 Right image를 사용하였다.

3. Feature matching & Reranking

앞서 선택된 총 400장(/ 쿼리 이미지)의 이미지에서 [3, 4] 모델을 이용해 key point와 매칭점들을 계산하고 선택된 포인트들 간의 Ransac을 이용한 Fundamental matrix estimation을 통해 outlier를 제거한다. 또한 기 학습된 Semantic segmentation 모델[5]을 이용하여 쿼리와 데이터베이스 이미지에 대한 segmentation 결과를 출력 한 후, 선택된 키포인트들이 속한 물체가 (자동차, 나무, 사람, 하늘, 도로) 일 경우 정확한 Localization을 하는데 악영향을 줄 수 있으므로 제거하였다.

이렇게 최종적으로 남은 포인트들의 개수(inlier)을 이용하여 400장의 데이터베이스 이미지를 정렬하여 가장 inlier수가 많은 순서대로 하나의 쿼리 이미지에 대해 총 20장의 데이터베이스 이미지를 선택한다. Inlier의 개수가 같을 경우에는 global descriptor로 계산한 similarity의 순서를 따른다.

4. 3d Projection & pose estimation

이전 단계에서 선택된 20장의 데이터베이스 이미지에 대해 해당 이미지에 인접한 timestamp를 갖는 Lidar point들을 projection한다(-4 ~ +4 seconds). 이후 projected된 포인트들과 데이터베이스 이미지의 선택된 키포인트 간의 거리가 가장 최소가 되는 점들을 찾아 해당 lidar point들을 해당 데이터베이스 포인트의 3D Point로 가정한다. 이 때, projected lidar point와 데이터베이스 key point간의 거리가 5 pixel 이상 차이가 날 경우에는 제외한다.

이렇게 선택된 3D point들을 Section. 3에서 매칭된 쿼리 이미지 상의 키 포인트와 함께 Ransac을 이용한 PnP 알고리즘을 통해 각각의 쿼리 이미지에 대한 pose 및 inlier들을 계산한다.

이 task의 목적은 최종 프레임(49)에 대한 pose를 예측하는 것이지만, 앞에서의 단계들로 계산된 최종 프레임의 pose가 정확하지 않을 수가 있기 때문에 일부 sequence에 대해서는 이전 프레임에 대해 계산된 결과를 이용하여 최종 pose를 추정하였다. 해당 sequence를 선택하는 과정에서는 아래 조건들을 이용한다.

1. Section. 2에서 이용한 쿼리 이미지(0, 9, 19, 29, 39, 49) 중 49번째 이미지에서 구한 pose가 가장 inlier 수가 많을 경우 이것을 최종 결과로 선택
2. 1을 만족하지 않을 경우 이전 프레임들에서 가장 inlier가 많은 pose를 선택한 후 49번째 프레임으로 주어진 odometry data를 이용하여 transformation한다. 이 때 49번째 프레임을 이용하여 구한 pose와의 거리가 1m 이하로 차이가 날 경우, 49

번째 프레임으로 구한 pose를 선택

3. 1과 2 둘 다 만족하지 않을 경우 앞의 프레임에서 가장 inlier가 많은 pose를 선택한 후 49번째 프레임으로 odometry data를 이용하여 transformation한다. 이렇게 구한 pose를 마지막 프레임의 pose로 선택한다.

Reference

[1] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, Marcin Dymczyk . From Coarse to Fine: Robust Hierarchical Localization at Large Scale

[2] Bingyi Cao, Andre Araujo, Jack Sim, Unifying Deep Local and Global Features for Image Search

[3] Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, SuperPoint: Self-Supervised Interest Point Detection and Description.

[4] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, SuperGlue: Learning Feature Matching with Graph Neural Networks

[5] Yi Zhu1*, Karan Sapra2*, Fitsum A. Reda2, Kevin J. Shih2, Shawn Newsam1, Andrew Tao2, Bryan Catanzaro2, Improving Semantic Segmentation via Video Prediction and Label Relaxation