

흔들림 환경 대응 지면에 정렬된 지역지도 작성 기법

Ground-aligned Local Mapping for Shaking Situations

허동욱¹·김성현¹·황현길²·안사훈³·최성록^{1†}

Dongwook Huh¹, Sung-Hyeon Kim¹, Hyunkil Hwang², Sahoon Ahn¹, Sunglok Choi^{1†}

Abstract: This paper presents a ground-aligned local mapping capable of compensating for shaking robots or sensors. The proposed method enables accurate local mapping using three-dimensional data such as depth images or point clouds, even without external pose information from localization or SLAM. The improved RANSAC is adopted for ground plane detection, incorporating plane filtering and asymmetric loss functions against outliers on the ground plane. Experimental results using a ZED 2i camera in a shaking environment demonstrate that the proposed method generates more accurate local maps compared to conventional approaches.

Keywords: Local mapping, Ground plane detection, RANSAC

1. 서 론

지역지도(local map)는 로봇 주의 공간 정보를 표현하고, 지역경로계획에서 꼭 필요한 기본 정보이다. 지역지도는 상황이나 용도에 따라 다르게 불린다. 자동차 분야에서는 1인칭 차량 시점의 다양한 센서 데이터를 top-view 시점의 지도로 표현하는데, 이를 bird-eye-view(BEV)라고 부른다. 특히 카메라 영상을 이용하는 경우 surround view 또는 around view로 부르기도 한다. 로봇 분야에서는 로봇 주변의 장애물 유무가 아닌 높이 정보로 표현하는 경우, 지형지도(terrain map)[1] 또는 높이지도(elevation map)[2]로 표현하기도 한다. 지역지도의 좌표계는 일반적으로 로봇중심의 지역좌표계를 사용하고, 로봇의 센서가 잘 고정된 상태에서 지역적으로 평탄한 지면을 흔들림 없이 이동할 때, 지역지도 또는 높이지도를 생성하는 과정이 단순하고 그 결과 생성된 지도 또한 깔끔하다.

그러나 로봇이나 센서가 크게 흔들리는 경우, 특히 외부 모듈로부터 로봇의 자세를 획득하여 흔들린 센서 데이터를 보정할 수 없거나 매우 부정확한 경우, 생성된 지역지도가 매우 부정확하고 울퉁불퉁하게 된다. 이러한 로봇이나 센서의 흔들림이 갖

은 대표적인 예가 로봇이 울퉁불퉁한 야지에서 동작하거나, 로봇이 휴머노이드나 사족로봇과 같이 다리형로봇인 경우이다. 로봇이나 센서의 흔들림을 보상하기 위해 짐벌(gimbal)과 같이 흔들림을 상쇄해주는 별도의 하드웨어를 사용하기도 한다. 또 지역지도를 생성하는 과정이 SLAM과 함께 수행되어 흔들리는 로봇의 위치를 보정하며 지도 작성[3]을 생성하기도 한다.

본 연구에서는 깊이영상(depth data) 또는 점군(point cloud)과 같은 3차원 데이터를 이용하여 지역지도를 생성할 때, 지역지도가 지면에 정렬되어 업데이트되도록 하여 로봇 또는 센서의 흔들림에 대응할 수 있는 기법을 제안한다. 특히 흔들림 보정을 위해 외부로부터 로봇의 자세 정보를 전혀 받지 않는 환경에서도 단독(stand-alone)으로 동작할 수 있는 지역지도 생성 기술을 제안한다.

2. 지면에 정렬된 지역지도 작성

지면에 정렬된 지역지도 생성을 위해서는 주어진 3차원 데이터에서 **지면(ground plane)**을 검출하는 단계가 선행되어야 한다. 본 연구에서는 지면이 아닌 부분(이상치; outlier)을 극복하여 지면을 찾기 위해 RANSAC을 이용한다. RANSAC은 랜덤 샘플을 이용한 평면 가설 생성 절차와 생성된 평면 가설을 주어진 데이터에 대해 검증하는 절차의 반복으로 구성되어 있다. 본 연구에서는 빠르고 효율적인 RANSAC 수행을 위해 아래와 평면 가설 생성 전후에 앞서 **지면의 제약조건을 이용한**

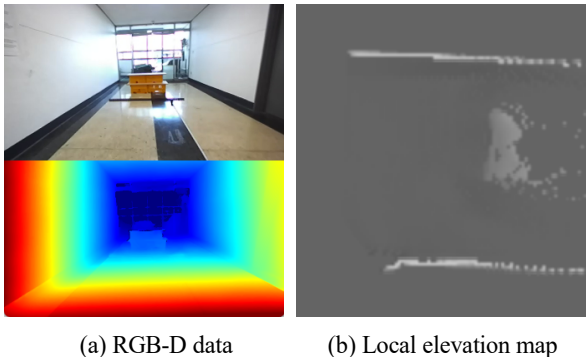
※ This research was supported by MSIT/NRF Grant for Bridge Convergence R&D Program (AI-based Localization and Path Planning on 3D Building Surfaces, 2021M3C1C3096810).

1. SEOULTECH, Seoul, Korea

2. University of Delaware, Newark, DE, USA

3. Hanyang University, Seoul, Korea

† Corresponding author (sunglok@seoultech.ac.kr)



[Fig. 1] Experimental setting: Shaking ZED 2i moved toward objects on a long corridor.

필터링을 통해 불필요한 가설 생성이나 가설 검증 단계를 생략할 수 있도록 하였다. 즉 생성된 평면 가설의 벡터(normal vector)와 평면 가설과 로봇 사이의 거리를 이용한 필터링을 수행한다. 또 평면 가설의 검증 단계 지면이 아닌 3차원 데이터가 대부분(지면 아래가 아닌) 지면 위에만 존재하는 특징을 이용한 비대칭(asymmetric) MSAC 손실함수를 이용한 평가 [4]를 수행하여 정확도를 높였다. 또 3차원 데이터에서 추정된 이상치 비율에 따라 RANSAC의 반복 횟수를 적응적으로 결정하여 RANSAC이 빠르고 효율적으로 동작하도록 하였다.

이후 RANSAC에 얻어진 지면에 해당하는 3차원 점군만을 이용하여 최소사승법(least squares)으로 지면의 파라미터를 최적화하여 더욱 정확한 지면 파라미터를 얻는다.

마지막으로 지면의 법선벡터를 이용해 주어진 3차원 데이터를 지면에 대해 정렬하고, 마지막으로 지면에 정렬된 3차원 데이터를 지역지도에 업데이트한다.

3. 실험 결과

실험에서는 Stereolabs의 ZED 2i 카메라를 이용하여 3차원 RGB-D 데이터를 획득하여 지역지도 작성하였다. [Fig. 1]과 같이 복도 환경에 크기를 알고 있는 물체를 전방에 놓고 ZED 2i 카메라를 손에 들고 해당 물체에 접근하여 이동하였다. 이때 카메라에 흔들림이 발생하였다. 생성된 지역지도의 정확도를 확인하기 위해 복도에 놓인 장애물의 높이를 지역지도에서 획득하여 실제 크기와 비교하였다. 해당 물체의 4개 지점의 높이를 획득하였고, RANSAC의 확실적인 특성을 고려하여 5회 알고리즘을 수행하고 평균 높이에 대한 높이 에러와 계산시간을 [Table 1]과 같이 측정하였다. Open3D[5]의 RANSAC 알고리즘을 대조군으로 비교하였고, 제안하는 기법을 [Table 1]과 같이 세 가지 세팅으로 실험하였다. [Ours w/o MSAC]은 MSAC을 제외한 기본 RANSAC 알고리즘에 필터링만 탑재한 버전이고, [Our w/o asymm.]은 MSAC에서 비대칭(asymmetric) 손실

[Table 1] Ablation study on our proposed method

Methods	Height [m]	Error [m] (↓)	Time [msec] (↓)
Ground truth	0.534	-	-
Open3D [5]	0.523	0.011	(C++) 158
Ours w/o MSAC	0.525	0.009	(Python) 266
Ours w/o asymm.	0.525	0.009	(Python) 272
Ours	0.533	0.001	(Python) 290

함수 부분을 제외한 버전이다. [Ours]는 제안하는 모든 기능을 추가한 버전이다.

실험 결과 모든 기능을 담고 있는 제안하는 기법(Ours)가 가장 낮은 에러 값을 가졌다. 다만 Open3D로 구현된 RANSAC에 비해 약 1.83배 가량 많은 계산시간이 소요되었는데, 이는 C++와 Python 구현에 따른 차이로 생각된다. Open3D와 비슷한 형태의 구현인 기본적인 RANSAC 코드도 약 1.68배 느렸다. Open3D는 C++로 구현됐지만 제안하는 알고리즘은 Python으로 구현하여 이러한 시간의 차이가 발생한 것으로 생각된다.

4. 결론

본 논문에서는 로봇이나 센서에 흔들림이 많은 환경에 대처하여 일관된 좌표계의 정확한 지역지도 작성을 위한 지면에 정렬된 지역지도 생성 기법을 제안한다. 제안하는 기법에서 지면을 정확하게 추정하는 것이 선행되어야 하고, MSAC에서 비대칭 손실커널을 적용하여 지면 위의 물체에 더욱 강인하고 정확하게 지면을 찾을 수 있었다. 복도 환경에서 흔들리는 ZED 2i를 이용한 실험에서 제안하는 기법은 기존 Open3D의 RANSAC을 이용한 기법보다 정확한 지역지도를 생성하는 것을 확인할 수 있었다. 현재는 제안된 알고리즘은 Python으로 구현되어 다소 느리게 동작하고, 추후 이를 C++로 구현할 예정이다.

References

- [1] Fankhauser et al., "Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization," IEEE RA-L, 2018 Github
- [2] Miki et al., "Elevation Mapping for Locomotion and Navigation using GPU," IROS, 2022 Github
- [3] Shan et al., "LeGO-LOAM: Lightweight and Ground-optimized Lidar Odometry and Mapping on Variable Terrain," IROS, 2018 Github
- [4] Choi et al., "Robust Ground Plane Detection from 3D Point Clouds," ICCAS, 2014
- [5] Zhou et al., "Open3D: A Modern Library for 3D Data Processing," arXiv:1801.09847, 2018 Homepage