

실내 자율 주행을 위한 3D Map 생성 시스템

문성태*, 한상혁**, 엄위섭***, 김연규****

3D Map Generation System for Indoor Autonomous Navigation

SungTae Moon*, Sang-Hyuck Han**, Wesub Eom***, Youn-Kyu Kim****

Abstract

For autonomous navigation, map, pose tracking, and finding the shortest path are required. Because there is no GPS signal in indoor environment, the current position should be recognized in the 3D map by using image processing or something. In this paper, we explain 3D map creation technology by using depth camera like Kinect and pose tracking in 3D map by using 2D image taking from camera. In addition, the mechanism of avoiding obstacles is discussed.

초 록

자율 주행을 위해 주행 지도, 위치 추적 및 목적지까지의 최단 경로 설정 연구가 필요하다. 특히 실내에서는 GPS 신호를 수신 받을 수 없기 때문에 이미지 프로세싱과 같은 방법을 통해 현재 위치를 인식하고 3차원 지도를 생성해야 한다. 본 논문에서는 Depth 카메라인 키넥트를 이용하여 3차원 지도를 생성하고, 일반 카메라로 촬영한 2차원 이미지를 이용하여 3차원 지도에서 현재 위치를 파악하는 방법에 대해 설명한다. 그리고 지도에서 장애물을 확인하고 단순화하는 방법에 대해서도 설명한다.

키워드 : Indoor Autonomous Navigation(실내 자율 주행), Kinect(키넥트), 3D Map(3차원 지도), 위치 추적 (Pose Tracking), Shortest Path Algorithm(최단 경로 알고리즘)

1. 서 론

최근 자율 주행에 대한 관심이 높아짐에 따라 다양한 분야에서 자율 주행을 위한 아이디어가 나오고 있다. 실내에서 자율 주행을 하기 위해서는 실내 지도 정보, 현재 위치 추적, 정해진 목적지까지의 패스 정보가 필요하다. 첫 번째로 실내

에서 자율 주행하기 위해서는 우선 실내 3차원 지도가 필요하다. 주변 환경에 대한 지도 작성을 위해 이미지의 특징점을 추출하고 이를 이용한 지도를 작성하는 방식이 많이 사용되고 있다. 특히 최근 PrimeSense 기술이 개발되고 이미지 정보와 함께 깊이 정보를 동시에 받을 수 있는 센서 장비가 출시되어 정확하고 빠르게 지도를 생

접수일(2012년 9월 14일), 수정일(1차 : 2012년 10월 17일, 게재 확정일 : 2012년 11월 1일)

* 융합기술연구팀/stmoon@kari.re.kr

** 융합기술연구팀/shan@kari.re.kr

*** 우주과학연구팀/wse@kari.re.kr

**** 우주과학연구팀/ykkim@kari.re.kr

성할 수 있게 되었다. 두 번째로 실내에서 현재 위치를 추적해야 한다. 실외와 달리 실내에서는 GPS 정보를 사용할 수 없기 때문에 위치 추적을 위해 기 생성된 3차원 지도를 바탕으로 이미지 특징점을 활용해 현재 위치를 찾는다. 컴퓨터 비전 분야에서는 이를 위치 추정 기술이라고 하며, 지도 작성기술과 함께 연결된 SLAM (Simultaneous Localization And Mapping) 연구들이 활발히 이루어지고 있다[1,2]. SLAM은 위치 인식과 지도를 동시에 작성하는 것을 의미하며 실내 자율 주행 구현을 위해 가장 기본적인면서도 핵심적인 기술이다. 마지막으로 목적지까지 패스 정보를 확보하기 위해 3차원 지도를 분석하고 가장 최단 거리를 찾는 알고리즘이 필요하다.

본 논문에서는 3차원 영상을 받을 수 있는 Kinect 센서를 활용하여 3차원 지도를 생성하고 현재 위치를 추적하며, 목적지까지 장애물을 회피하며 갈 수 있는 방안에 대해 설명한다.

논문의 구성은 다음과 같다 2장에서는 자율 주행과 관련된 연구들을 설명하고, 3장에서는 3차원 영상 센서인 Kinect에 대해 설명한다. 4장에서는 3차원 지도 생성 알고리즘에 대해 설명하고, 5장에서는 현재 위치 추적 방안에 대해 설명하며, 6장에서는 정해진 목적지까지 장애물을 회피하며 갈 수 있는 방안에 대해 설명한다. 7장에서는 시험 결과에 대해 설명하고 마지막으로 8장에서는 결론과 향후 계획에 대해 논의한다.

2. 관련 연구

3차원 지도 생성과 관련하여 다양한 연구가 진행되고 있다. [3]에서는 쿼드로터에 Laser range-finder를 사용하여 지도를 생성하는 시스템을 개발하였다. 하지만 Laser range finder를 사용하는 경우 2차원 상의 깊이 정보만을 알 수 있어서 효율적으로 3차원 지도를 생성하기 위해서는 많은 시간이 소요된다. [4]에서는 스테레오 카메라를 사용하여 3차원 지도를 생성하였다. 하지만, 이 경우 깊이 정보를 파악하기 위해 두 이미지 간의 차이를 가지고 계산하기 때문에 처리 시

간이 많이 필요하고, 정확하지 않다. [5]에서는 RGB-D 카메라를 사용하여 3차원 지도를 생성하였다. 본 논문에서는 기존의 3차원 지도를 생성하는 방법과 동일한 방법으로 생성하면서 메모리 사용을 최소화 시킬 수 있는 방법을 제안하였다.

3. 3차원 영상 획득 센서

3차원 영상을 획득하는 방법은 수동 센서 기반 방법과 능동 센서 기반 방법으로 나눌 수 있다. 수동 센서 기반 방법은 두 대 혹은 그 이상의 카메라로부터 획득한 2차원 영상의 상관관계를 유추함으로써 깊이 정보를 계산하는 방식이고, 능동 센서 기반 방식은 레이저, 자외선 등과 같은 다양한 종류의 센서를 이용해서 3차원 장면으로부터 깊이 정보를 직접적으로 획득하는 방법이다. 능동 센서 기반 방법은 높은 정확도의 깊이 영상을 획득할 수 있는 대신 장비가 워낙 고가이다 보니 접할 수 있는 기회가 많지 않았다. 하지만, 최근 Kinect와 같이 저가임에도 불구하고 높은 성능을 보이는 카메라가 시중에 출시되면서 다양한 형태의 응용 분야에 많이 사용되기 시작했다.

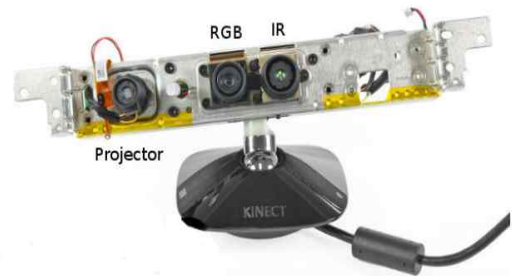


그림 1. 키넥트 내부 구조

Kinect는 마이크로소프트社에서 개발한 센서장비로서 RGB 카메라, 적외선 프로젝터 및 IR 카메라가 장착되어있다[6]. Kinect는 IR 프로젝터와 IR 카메라를 사용하여 깊이 정보를 획득하고 RGB 카메라에서 나온 영상 정보를 동시에 받아들이는 PrimeSense 기술을 사용한다. 초기

Kinect는 동작 인식 및 물체 추적 기능을 활용하여 XBOX 게임에 사용되었지만, 최근에는 낮은 가격과 높은 성능 때문에 다양한 연구 분야에 활용되고 있다. 특히 내부 프로토콜 분석을 통해 OpenNI에서 라이브러리를 제공하고 있다. 전체적인 Kinect의 스펙은 표 1과 같다.

표 1. Kinect 스펙[7]

특징	값
Light Source	Infraed
Field of View	Horizontal 57° Vertical 43°
Tilt Range	54 deg
Measurable Range	0.7~6m
Angular Resolution	0.08 deg
Scan Time	33 msec/scan
Interface	USB
Dimension	307*62*51mm
Weight	460g

4. 3차원 지도 생성

4.1. 알고리즘

3차원 지도를 생성하기 위해서는 그림 2와 같이 이미지 특징점 추출, 특징점 매칭, 포즈 계산이 이루어진다.

이미지 기반 특징점 추출은 SIFT (Scale-Invariant Feature Transform), SURF (Speed Up Robust Feature), ORB (ORiented BRIEF)와 같은 다양한 기법이 있다. 이 방법들은 이미지 내에서 고유한 값을 갖는 포인트를 찾아내는 방법으로 최근에는 GPU와 연관시켜 보다 빠르게 처리를 하고 있다. 본 논문에서는 이미지 특징점을 찾기 위해 GPU-SURF[8] 알고리즘을 사용한다.

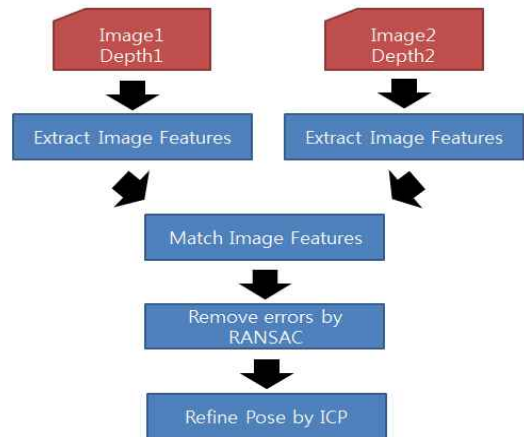


그림 2. 3차원 지도 생성 알고리즘

특징점 매칭은 그림 3과 같이 각 프레임에서 나온 같은 특징점을 매칭시키는 기술이다. 하지만 특이점 매칭이 잘못되는 경우가 발생할 수 있기 때문에 잘못된 매칭이 되는 부분을 찾아 제거해야한다. 이를 위해 확률 기반의 RANSAC (RANdom SAmple Consensus) 알고리즘[9]을 사용하여 불필요한 부분을 제거한다. RANSAC은 측정 노이즈가 심한 원본 데이터로부터 모델 파라미터를 예측하는 방법으로 전체 원본 데이터 중에서 모델 파라미터를 결정하는데 필요한 최소의 데이터를 랜덤하게 샘플하면서 반복적으로 해를 계산함으로써 최적의 해를 찾는 방법이다.

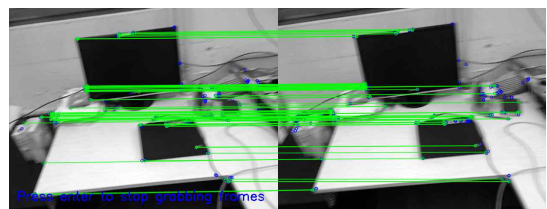


그림 3. 이미지 특징점을 활용한 매칭

마지막으로 Pose 계산은 ICP(Iterative Closest Points) 기술[9]을 사용한다. ICP는 수식 1과 같이 모델과 데이터 사이의 오차(떨어진 거리 정도)를 최소화 시키는 Pose를 계산하게 된다. 여기서 R 과 t 는 각각 회전과 병진 운동을 나타내

고, N_m 과 N_d 는 각각 모델과 데이터의 포인트 개수를 나타낸다. m_i 와 d_j 는 각각 모델과 데이터의 셋을 의미한다. 마지막으로 $w_{i,j}$ 는 매칭 여부를 나타내며 1 혹은 0이다.

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \|m_i - (Rd_j + t)\|^2 \quad (6)$$

하지만, 이미지 기반의 3차원 맵 생성 시 누적 오차가 발생한다. 이 경우를 해결하기 위해 Loop Closure를 수행해야하며 본 논문에서는 g2o (General Framework for Graph Optimization) 알고리즘을 사용한다.

4.2. Pose 생성

3차원 지도 생성에 있어 가장 중요한 부분은 각 프레임을 연결시키기 위한 Pose이다. 앞에서 언급한 이미지 추출, RANSAC 및 GICP는 모두 Pose를 계산하기 위한 알고리즘이다. Pose를 계산하기 위해 동차 변환 행렬 (Homogeneous Transformation Matrix)을 사용한다. 동차변환행렬은 수식 2과 같이 4X4 행렬이고 회전과 병진 운동을 동시에 표현할 수 있다.

$$\begin{bmatrix} & & & t_x \\ & R & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

3차원 지도를 생성할 때 우선 첫 번째 프레임의 pose는 4x4 단위행렬로 두고 두 번째 프레임부터 각 상대 Pose를 수식 3와 같이 누적하여 Pose를 구한다.

$$P_k = \prod_{i=k}^1 H_{i-1}^i P_0 \quad (8)$$

이렇게 계산된 Pose를 가지고 각 프레임들을 연결하게 되면 3차원 지도를 생성할 수 있다. 기존 이미지 처리 방법은 메모리 사용 효율성을 높

이기 위해 그림 4와 같이 저장된 프레임과 새로 들어온 프레임과의 매칭된 특징점의 개수를 계산하고 한계값 이하로 떨어질 때 그 프레임을 3차원 지도에 사용하는 방법을 채택하였다.

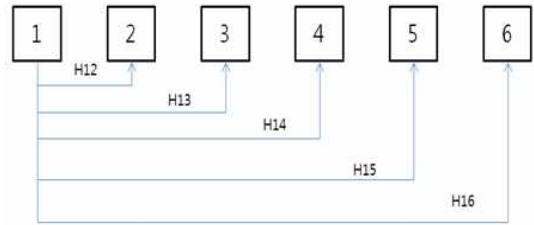


그림 4. 기존 Pose 계산 방식

하지만 이 경우 그림 5와 같이 특징점의 개수가 적어져 매칭이 잘못 되는 경우가 발생하게 되어 정확하지 않은 Pose를 계산하게 되는 문제가 있다.

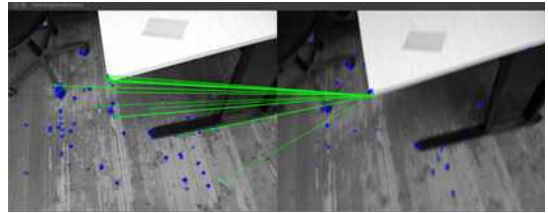


그림 5. 잘못된 매칭 경우

본 논문에서는 잘못된 이미지 매칭을 최소화하고, 메모리 사용 효율성을 높이기 위해 그림 6과 같이 모든 프레임에 상대 변경 내용을 기록하고 있다가 한계값 이하로 매칭이 떨어질 때 상대 변경 내용을 사용하는 방법을 사용한다. 이 경우 기존 방법과 달리 전체 프레임의 Pose값을 사용하기 때문에 하나의 잘못된 값이 나오더라도 전체적인 에러를 줄일 수 있다.

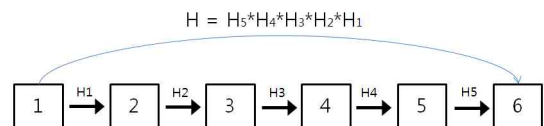


그림 6. Pose 계산 방식

5. 위치 추적

실내에서는 GPS를 사용할 수 없기 때문에 위치 추적을 위해 다른 방법이 필요하다. 3차원 지도를 만드는 방법과 동일하게 현재 위치에서 Kinect를 사용하여 획득한 정보를 가지고 카메라 포즈를 계산할 수 있지만, 무인 비행체와 같이 무게에 민감한 장치에 위치 추적하기 위해 Kinect와 같은 무거운 카메라를 탑재하기는 어렵다. 따라서 본 논문에서는 소형 카메라를 통해 얻은 2차원 이미지를 사용하여 기존에 생성한 3차원 지도에서의 위치를 얻는 방법을 사용하고자 한다.

2차원 이미지와 3차원 지도가 있을 때 카메라 포즈를 계산하는 문제는 Perspective-n-Point Problem (PnP)로 알려져 있다. PnP란 카메라의 intrinsic 값이 주어졌을 때 카메라의 위치 및 촬영 각도를 파악하는 방법을 말한다. 카메라 포즈를 계산하는 원리는 그림 7과 같이 3차원 모델 상의 두 점인 P_i 와 P_j 가 카메라 센터에서 바라볼 때 나오는 각 $\theta_{(i,j)}$ 는 2차원 이미지 상의 같은 두 점인 x_i 와 x_j 에서 나오는 각도가 같다는 점을 이용하여 반복적으로 같은 각도가 같은 부분을 찾아 카메라 포즈(c)를 구할 수 있다. 하지만 실제 데이터는 일부 오류가 있기 때문에 RANSAC 알고리즘을 이용하여 오류 부분을 제거하고, 최적의 위치를 찾아내야 한다.

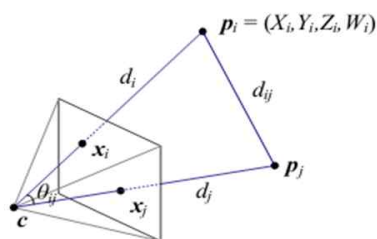


그림 7. 카메라 위치 추정 방법

본 논문에서는 OpenCV에서 제공하는 EPnP (Efficient Perspective-n-Point) 기법을 사용하여

PnP 문제를 해결하였고, 그림 8과 같이 카메라 위치를 파악할 수 있었다.

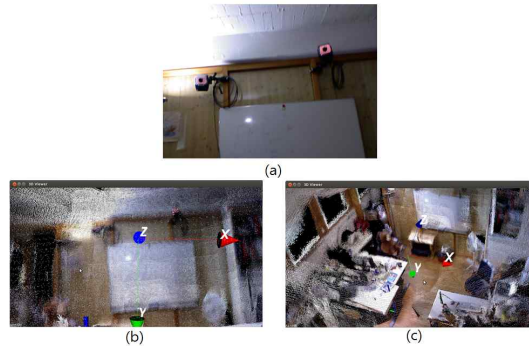


그림 8. 2차원 이미지를 가지고 카메라 포즈 계산 결과 (a) 2차원 이미지 (b) 카메라 포즈에서 바라본 3차원 지도 화면 (c) 다른 각도에서 바라본 3차원 지도 화면

6. 장애물 회피

로버 등을 이용하여 알려지지 않은 행성을 탐사하는 경우, 운행 중 획득한 3차원 지도는 추후 복귀 경로 산출에 이용될 수 있을 뿐만 아니라, 장애물을 파악하고 회피하는데 사용될 수 있다. 우선 복귀 가능한 경로를 파악하기 위해서는 로버에서 3차원 지도에서 경로 정보를 획득하여야 한다. 경로 정보를 확인하기 위해 본 논문에서는 간략화 작업을 수행하였다. 간략화 작업을 위해 우선 장애물이라 판단되는 영역을 결정해야 하는데 시험을 위해 그림 9와 같이 장애물 높이를 10cm 이상으로 정한 후 수행하였다. 실험을 위해 로버 작업장을 3차원 가시화하여 작업하였다.

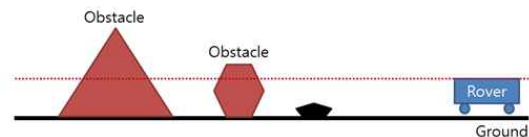


그림 9. 장애물 정의

그 결과 로버 실험장에서 그림 10과 같이 장애물만을 필터링할 수 있었고, 복잡도를 낮추기

위해 Octree를 사용하여 그림 11과 같이 간략화 작업을 수행하였다.

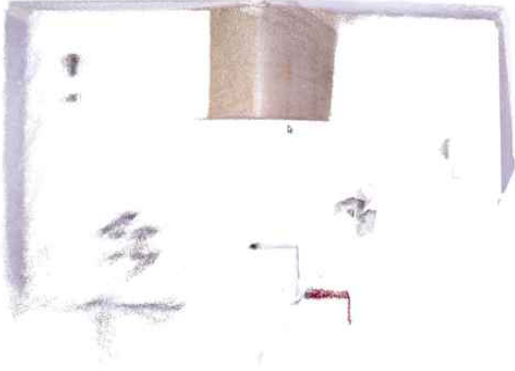


그림 10. 장애물 필터링 작업

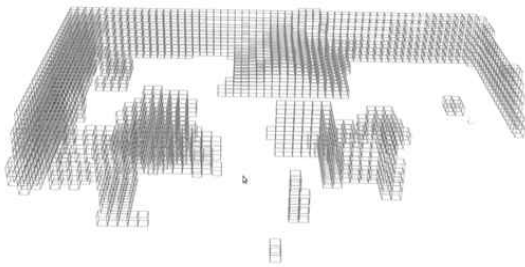


그림 11. Octree를 사용한 간략화 작업 수행

그 결과 그림 12와 같이 간략화 작업을 수행하여 최단 경로를 쉽게 얻을 수 있었다.

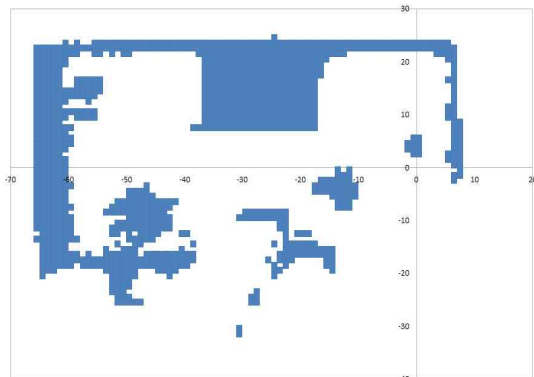


그림 12. 간략화 작업 수행

7. 시험 결과

7.1. 개발 환경

본 논문에서 개발한 3차원 지도 생성 프로그램은 Linux (Ubuntu 12.04) 환경에서 C++로 개발되었고, 최대한 OpenSource를 사용하여 개발시간을 단축시켰다. 프레임워크로 MRPT (Mobile Robot Programming Toolkit)[11]가 사용되었고, 3차원 지도 처리를 위해 PCL 라이브러리[12]를 사용하였다.

7.2. 시험 결과

실제 실험에 앞서 우선 뮌헨 TUM 대학의 Computer Vision Group[13]에서 제공하는 Data Set을 가지고 실험해 보았다. 이 그룹에서는 Kinect로 촬영한 영상 정보와 함께 카메라 위치를 파악하기 위해 절대 경로 (Ground Truth)를 제공하고 있다. 실험 결과 그림 13와 같이 평균 10Cm 정도 오차 범위에서 처리됨을 볼 수 있었다.

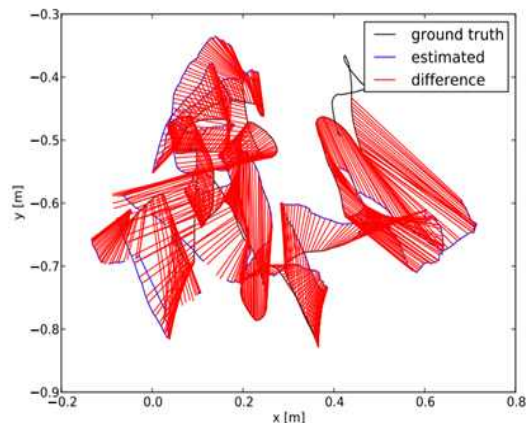


그림 13 생성 정보 비교

3차원 지도 생성을 위해 Kinect를 활용해 시험 대상을 촬영하였다. 수행 결과 그림 14와 같이 전체 지도를 생성할 수 있었다. 카메라 위치를 추적해 본 결과 그림 15와 같았다.



그림 14. 생성된 3D 지도

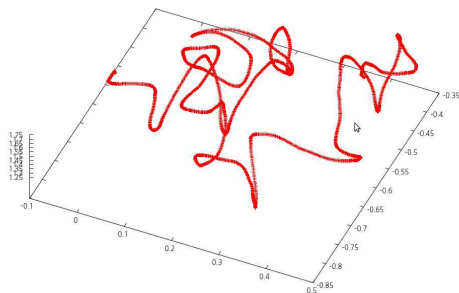


그림 15. Camera Trajectory

7.3. 한계점 및 제안

지금까지 개발된 버전에는 몇 가지 문제점이 존재한다. 첫째 넓은 범위의 Map을 생성 하는 경우 Tracking을 놓치는 경우가 발생하였다. 실험을 위해 10m 이상의 복도와 방이 있는 공간을 촬영한 결과 그림 16과 같이 간헐적으로 Tracking을 놓쳐 지도가 망가지는 현상을 볼 수 있었다. 분석 결과, 복도에서 방을 촬영하기 위해 좌우로 움직일 때 이미지 매칭 처리가 늦어져 Tracking을 놓치는 경우가 발생하였다.

둘째, 시간이 지날수록 그림 17과 같이 누적 오차가 발생하였다. 누적 오차를 줄이기 위해 Loop Closure 방식을 사용하고 있지만, 누적 오차는 여전히 남아있다.



그림 16. Tracking을 놓치는 경우

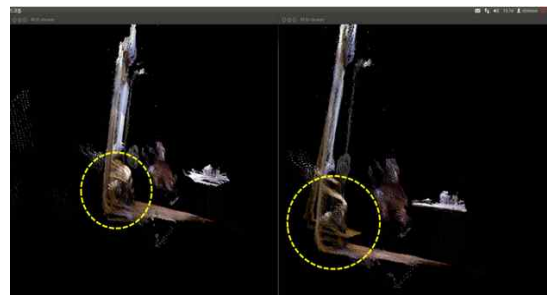


그림 17. 누적 오차 발생한 경우

원인을 찾기 위해 Ground Truth 정보를 갖는 샘플을 사용하여 분석한 결과 그림 18과 같이 x,y,z에 대한 오차는 yaw, raw, pitch에 비해 상당히 크다는 것을 알 수 있었다. 분석 결과 오차는 ICP 수행 중 발생하며 정확하지 않은 RawData로 인해 발생하는 것으로 추정하고 있다. 특히 IR 데이터의 경우 오차가 많이 발생하고 있다.

8. 결론 및 향후 계획

본 논문에서는 자율 주행을 위해 3차원 지도를 생성하고, 현재 위치에서 카메라로 촬영한 2차원 이미지와 3차원 지도상에 저장된 특징점들을 활용하여 위치를 파악하는 방법에 대해 설명하였다. 그리고 3차원 지도 단순화 작업을 통해 정해진 목적지까지 최단 거리를 파악하는 방법에 대해서도 논의하였다.

하지만 넓은 범위 촬영 시 누적 오차로 인해

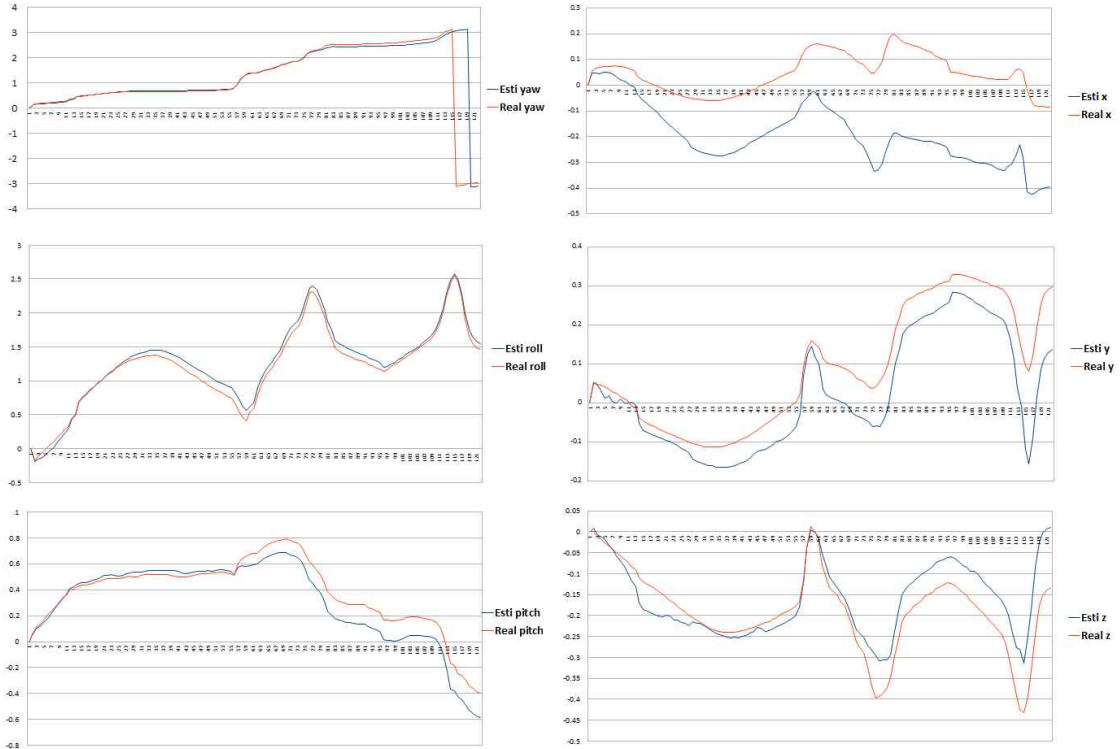


그림 18. 수행 결과 6-DOF 오차

정확한 3차원 지도를 얻지 못하는 경우가 발생하므로 생성된 3차원 지도를 이용한 운항체의 위치 추적 및 장애물 회피는 향후 추가 연구가 필요하다.

약 어 정 리

GPS	Global Positioning System
3D	3 Dimension
SLAM	Simultaneous Localization And Mapping
SIFT	Scale-Invariant Feature Transform
SURF	Speed Up Robust Feature
ORB	ORiented BRIEF
BRIEF	Binary Robust Independent Elementary
RANSAC	RANdom SAMple Consensus
ICP	Iterative Closest Points
GICP	Generalized-ICP
g2o	General Framework for Graph Optimization

OpenCV	Open Computer Vision
PnP	Perspective-n-Point
EPnP	Efficient Perspective-n-Point
MRPT	Moblie Robot Programming Toolkit
PCL	Point Cloud Library

참 고 문 헌

1. Hugh D., Tim B., "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithm," Vol. 13, Issue 2, 2006, pp. 99~110
2. Hugh D., Tim B., "Simultaneous Localization and Mapping (SLAM): Part II State Of The Art," Vol. 13, Issue 3, 2006, pp. 108~117
3. William M., Ivan D. Jizhong X., "3D Indoor Mapping for Micro-UAVs Using Hybrid

- Range Finders and Multi-Volume Occupancy Grids," Robotics Science and System, 2010
4. K. Kongolige, M. Agrawal, "Outdoor mapping and navaigation using setero vision," In Experimental Robotics, Vol. 39, ,2008, pp. 179~190.
 5. P. Henny, M. Krainin, "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments," International Symposium on Experimental Robotics (ISER), 2010
 6. Jam S., Michal J., "3D with kinect," IEEE ICCV, 2011, pp. 1154~1160.
 7. Chan-Soo Park, Sung-Wan Kim, "Comparison of Plane Extraction Performance using Laser Scanner and Kinect," Ubiquitous Robots and Ambient Intelligence, 2011, pp. 153~155.
 8. H. Bay, A. Ess, "SURF: Speeded Up Robust Features,", Computer Vision and Image Understanding, Vol. 110, No. 3, 2008, pp. 346~359.
 9. M. Fischeler, R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Graphics and Image Processing, Vol. 24, No. 6, 1981, pp. 381~395.
 10. Paul J., Neil M., "A Method for Registration of 3-D Shapes," Trans. of the PAAMI, Vol. 14, No. 2. pp. 239~256.
 11. MRPT, "<http://mrpt.org>"
 12. PCL, "<http://pointclouds.org>"
 13. RGB-Dbenchmark, "<http://vision.in.tum.de/datasets/rgbd-dataset/tools>"