



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2022년 8월

석사학위논문

# 시각 관성 주행 거리 측정을 사용한 위치, 자세 추정 및 3D 지도 작성

조선대학교 대학원

전자공학과

이 보 은

# 시각 관성 주행 거리 측정을 사용한 위치, 자세 추정 및 3D 지도 작성

**Pose Estimation and 3D Mapping Using Visual Inertial Odometry**

2022년 8월 26일

조선대학교 대학원

전자공학과

이 보 은

# 시각 관성 주행 거리 측정을 사용한 위치, 자세 추정 및 3D 지도 작성

지도교수    고    낙    용

이 논문을 공학석사학위신청 논문으로 제출함.

2022년 4월

조 선 대 학 교    대 학 원

전자공학과

이    보    은

# 이보은의 공학석사학위논문을 인준함

위원장 조선대학교 교수 유성현 (인)

위원 조선대학교 교수 고낙용 (인)

위원 조선대학교 교수 문지윤 (인)

2022년 5월

조선대학교 대학원

## 목 차

|   |    |
|---|----|
| 제1장 서론 .....                                    | 1  |
| 제1절 연구 배경 .....                                 | 1  |
| 제2절 연구 목적 .....                                 | 2  |
| 제3절 연구 내용 .....                                 | 3  |
| 제4절 기존 연구 .....                                 | 4  |
| <br>  |    |
| 제2장 센서 보정 .....                                 | 7  |
| 제1절 스테레오 카메라의 보정 .....                          | 7  |
| 제2절 스테레오 카메라와 IMU의 보정 .....                     | 10 |
| 제3절 Allan Variance .....                        | 16 |
| <br>  |    |
| 제3장 다중 상태 제약 칼만 필터를 사용한 시각 관성<br>주행 거리 측정 ..... | 24 |
| 제1절 상태 벡터 .....                                 | 25 |
| 제2절 Process model .....                         | 26 |
| 제3절 Measurement model .....                     | 29 |
| 제4절 확장 칼만 필터 업데이트 .....                         | 32 |
| 제5절 이미지 프로세싱 .....                              | 34 |

|  |    |
|--|----|
| 제4장 지도 작성 .....                        | 45 |
| 제1절 지도의 종류 .....                       | 45 |
| 제2절 RTAB-Map을 이용한 지도 작성 .....          | 47 |
| 제5장 실험 및 결과분석 .....                    | 49 |
| 제1절 센서 보정 .....                        | 49 |
| 제2절 시각 관성 주행 거리 측정 및 RTAB-Map 결과 ..... | 61 |
| 제3절 시각 관성 주행 거리 측정 성능 비교 .....         | 76 |
| 제6장 결론 .....                           | 95 |
| 참고문헌 .....                             | 96 |

## 도 목 차

|   |    |
|---|----|
| 그림 2.1 카메라 보정을 위한 체커보드 패턴 .....                     | 8  |
| 그림 2.2 카메라 초점 거리 .....                              | 9  |
| 그림 2.3 스테레오 카메라와 IMU 사이의 보정 흐름도 .....               | 11 |
| 그림 2.4 카메라-IMU 보정 프레임 .....                         | 12 |
| 그림 2.5 Allan variance 도출에 사용되는 데이터 구조 .....         | 17 |
| 그림 2.6 Bias instability에 대한 그래프 .....               | 20 |
| 그림 2.7 ARW/VRW 노이즈에 대한 그래프 .....                    | 22 |
| 그림 2.8 RRW 노이즈에 대한 그래프 .....                        | 23 |
| 그림 3.1. 일반적인 스테레오 다중 상태 제약 칼만 필터의 흐름도 .....         | 25 |
| 그림 3.2. 선형 및 원형 픽셀 간 위치를 보여주는 핵 $C$ 의 1차 인접 .....   | 35 |
| 그림 3.3 일치시킬 두 곡선 .....                              | 40 |
| 그림 3.4 스테레오 카메라 비전 .....                            | 43 |
| 그림 4.1 그리드 지도 예시 .....                              | 45 |
| 그림 4.2 특징 기반 지도 예시 .....                            | 46 |
| 그림 4.3 토폴로지컬 지도 예시 .....                            | 47 |
| 그림 4.4 메트릭 지도 예시 .....                              | 47 |
| 그림 4.5 RTAB-Map의 메모리 관리 모델 .....                    | 48 |
| 그림 5.1 oCamS-1CGN-U 외관 .....                        | 50 |
| 그림 5.2 스테레오 카메라 보정에 사용한 April grid 패턴 .....         | 51 |
| 그림 5.3 bag 파일에 저장된 카메라 image raw 데이터 .....          | 52 |
| 그림 5.4 스테레오 카메라의 polar error와 azimuthal error ..... | 53 |
| 그림 5.5 스테레오 카메라의 재투영 오차 .....                       | 53 |
| 그림 5.6 Allan Deviation 결과 .....                     | 55 |
| 그림 5.7 가속도계와 자이로스코프에 대한 Allan Deviation 그래프 .....   | 56 |
| 그림 5.8 가속도 비교 및 오차 .....                            | 58 |

|         |   |    |
|---------|---|----|
| 그림 5.9  | 각속도 비교 및 오차 .....                               | 59 |
| 그림 5.10 | 카메라에 대한 재투영 오차 .....                            | 60 |
| 그림 5.11 | 실험 환경 .....                                     | 62 |
| 그림 5.12 | 바닥을 향한 경우 시각 관성 주행 거리 측정 시각화 .....              | 63 |
| 그림 5.13 | 바닥을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프 ..... | 65 |
| 그림 5.14 | 바닥을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과 .....       | 66 |
| 그림 5.15 | 바닥을 향했을 경우 카메라 화면 .....                         | 67 |
| 그림 5.16 | 정면을 향한 경우 시각 관성 주행 거리 측정 시각화 .....              | 68 |
| 그림 5.17 | 정면을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프 ..... | 69 |
| 그림 5.18 | 정면을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과 .....       | 71 |
| 그림 5.19 | 천장을 향한 경우 시각 관성 주행 거리 측정 시각화 .....              | 72 |
| 그림 5.20 | 천장을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프 ..... | 73 |
| 그림 5.21 | 천장을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과 .....       | 74 |
| 그림 5.22 | 루프 클로저가 작동하지 않은 천장 지도 작성 결과 .....               | 75 |
| 그림 5.23 | 바닥을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프 .....         | 78 |
| 그림 5.24 | 바닥을 향한 경우 7번의 실험에 대한 z축 통계 .....                | 78 |
| 그림 5.25 | 바닥을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프 .....         | 81 |
| 그림 5.26 | 정면을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프 .....         | 83 |
| 그림 5.27 | 정면을 향한 경우 7번의 실험에 대한 z축 통계 .....                | 83 |
| 그림 5.28 | 정면을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프 .....         | 86 |
| 그림 5.29 | 천장을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프 .....         | 88 |
| 그림 5.30 | 천장을 향한 경우 7번의 실험에 대한 z축 통계 .....                | 88 |
| 그림 5.31 | 천장을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프 .....         | 91 |
| 그림 5.32 | 특징점 추출 과정에서의 Data Association .....             | 94 |

## 표 목 차

|  |    |
|--|----|
| 표 5-1. oCamS-1CGN-U 모델 카메라 사양 .....      | 49 |
| 표 5-2. 보정 결과 구해진 카메라의 내부 및 외부 파라미터 ..... | 54 |
| 표 5-3. IMU에 대한 yaml 파일 .....             | 55 |
| 표 5-4. Allan variance 결과 .....           | 57 |
| 표 5-5. 카메라와 IMU 보정 결과 .....              | 61 |
| 표 5-6. 각 방향에 대한 z축 기준 평균 .....           | 92 |
| 표 5-7. 각 방향에 대한 z축 기준 최댓값 및 최솟값 .....    | 92 |
| 표 5-8. 각 방향에 대한 z축 기준 표준편차 .....         | 93 |

# ABSTRACT

## **Pose Estimation and 3D Mapping Using Visual Inertial Odometry**

Lee, Boeun

Advisor : Prof. Ko, Nak Yong, Ph. D.

Dept. of Electronic Engineering,

Graduate School of Chosun University

Map building and real-time localization are essential for the autonomous navigation of mobile robots. This paper shows an implementation of mapping and localization algorithm and analysis of the performance of the algorithm. The implementation uses multi-state constraint Kalman filter(MSCKF) for localization and RTAB-Map for mapping. This study uses ceiling-view image and floor-view image, whereas previous applications of MSCKF and RTAB-Map generally use front-view image for localization. The implementation calibrates the camera and inertial measurement unit(IMU). The calibration includes transformations among the cameras and IMU. It also uses Allan variance approach to adjust the covariance of measurement errors. The trajectories by the MSCKF were compared to investigate the effect of viewing direction to the localization performance of VIO and mapping performance of the RTAB-Map. The analysis show that number and characteristics of feature affect the performance. Even if there are large number of featured points recognized, the performance degrades if correspondence of features are not correctly identified. The results of the study can be used for adequate implementation of navigation algorithms in 3-Dimensional space.

# 제1장 서론

## 제1절 연구 배경

최근 자율주행 기술 발전으로 자율주행 로봇에 관한 관심이 높아지고, 연구 및 개발도 활발하게 이루어지고 있다. 국내 기업들뿐만 아니라 해외 소프트웨어 기업인 구글, 엔비디아, 인텔 등에서도 자율주행에 대한 연구 개발이 꾸준히 이루어지고 있다[1]. 이러한 자율주행 기술은 특히 로봇 산업 시장 확대로 로봇 분야의 수요가 높아지고 있다.

자율주행 로봇은 재난 및 재해 지역, 군사 지역, 우주 또는 해저 지역 등의 인간이 직접 하기 어렵고 위험하고 부적합한 곳에서 업무를 대신 수행하기 위해 개발 및 연구되었다. 초기의 로봇들은 인간이 직접 조작하여 업무 및 목적 등을 수행해야 했다. 하지만 현재는 카메라, 관성 측정장치(IMU, Inertial Measurement Unit), GPS(Global Positioning System) 등 로봇에 설치할 수 있는 센서들의 개발로 인간의 개입 없이도 로봇 스스로가 판단 가능하고, 업무를 수행할 수 있게 되었다[2].

자율주행 기술이 많이 적용되는 로봇 분야에는 모바일 로봇(mobile robot), 무인 항공기(UAV, Unmanned Aerial Vehicle) 등이 있다. 자율주행 기술이 적용된 모바일 로봇은 주로 농약 살포, 골프장 잔디 깎기, 무인 군사 작전 등에 사용된다. 최근에는 작물 분배, 과일 따기와 같은 농업 응용 분야에 적용하는 연구가 이루어지고 있다[3]. 모바일 로봇의 자율주행을 위해서는 대부분 레이저 센서나 초음파 센서 등과 같은 거리 측정 센서를 사용한다[4].

무인 항공기는 지난 몇 년 동안 민간 및 국방 분야에 광범위하게 사용되었다. 무인 항공기는 지역 모니터링, 지도 작성, 구조 조사, 소방과 같은 다양한 민간 업무와 국경 감시, 정찰 등과 같은 국방 업무를 진행하는 데에 사용되었다. 이런 무인 항공기에는 대부분 GPS와 IMU 센서 등이 사용된다[5]. 이러한 로봇의 자율주행을 위해서는 로봇의 현재 위치 파악과 지도 작성 과정이 필수적이다.

로봇이 자율주행을 하기 위해서는 자신의 위치 파악뿐만 아니라 장애물을 회피하면서 경로를 생성하여 목표 지점까지 스스로 주행해야 한다. 이때 지도 작성 과정이 필요하다. 지도 작성은 안전하고 성공적인 자율주행을 가능하게 하는 데 중요한 역할을 한다. 일반적으로 지도는 실제 환경의 특징과 위치를 객체로써 표현

한 것이다[6].

지도 작성 방법은 센서와 데이터 저장 방법에 따라 나누어진다. 종류에는 특이점 지도(feature based map), 점유 격자 지도(occupancy grid map), 토폴로지컬 지도(topological map) 및 스캔 매칭 지도(scan matching map) 등이 있다[7]. 현재 실내외, 2D 및 3D 지도 작성을 위한 연구들이 많이 이루어지고 있다. 지도 작성을 위해서 라이다(LiDAR, Light Detection And Ranging) 또는 깊이 카메라(depth camera)를 이용할 수 있다[8].

라이다는 레이저 펄스를 쏘 목표물에 맞고 다시 돌아오는 시간을 측정한다. 측정된 시간을 통해 장애물까지의 거리, 형태 및 크기 등을 파악하여 주변을 인식함으로써 지도를 작성할 수 있다[9]. 스테레오 깊이 카메라는 스테레오 정합(stereo matching)을 통해 주변 환경의 깊이 즉, 3차원 위치 정보를 계산함으로써 지도를 작성할 수 있다[10].

비전 연구와 관련하여 일반적으로 점, 선, 혹은 점과 선을 모두 사용한 특징 추출을 사용함으로써 위치 및 자세를 추정하고 대부분 정면을 보고 실험 및 시뮬레이션을 진행한다[11-13]. 비전 관련 연구는 시각 관성 주행 거리 측정(VIO, Visual Inertial Odometry), 시각 관성 항법 시스템(VINS, Visual Inertial Navigation System) 등이 있다. 또한, 3D 지도 작성은 일반적으로 라이다, 카메라, 혹은 라이다와 카메라를 동시에 사용하여 진행하고, 마찬가지로 대부분 정면을 보고 실험 및 시뮬레이션을 진행한다[14-16].

## 제2절 연구 목적

앞에서 언급한 기존의 제시된 시각 관성 주행 거리 측정과 같은 비전 관련 연구에서 사용한 위치 및 자세 추정과 3D 지도 작성은 대부분이 카메라를 정면을 향하게 하고 실험 및 시뮬레이션을 진행한다. 하지만, 이러한 일반적으로 사용하고 있는 카메라를 정면으로 향하게 한 시각 관성 주행 거리 측정과 3D 지도 작성의 경우 상황에 따라 수중에서 작업하는 지형 탐지 로봇 등에는 적합하지 않을 수 있다는 한계점이 존재한다. 수중에 떠서 지형을 탐지하는 로봇의 경우 카메라를 정면을 향하도록 하면, 깊은 물 속은 어두워 작업을 할 수 없게 된다. 따라서 이 경우에는 바닥을 향해 탐지해야 하므로 카메라를 정면이 아닌 바닥을 향하도록 하여

이동체의 위치 및 자세를 추정하고, 3D 지도 작성을 진행해야 한다.

본 논문은 스테레오 카메라와 IMU를 통해 자율주행을 위한 시각 관성 주행 거리 측정 시스템을 Ubuntu 환경에서 ROS를 사용하여 구현하고, 3D 지도 작성을 진행함으로써 이동체의 위치 및 자세 추정과 3D 지도를 작성하는 것에 목적을 두고 있다. 시각 관성 주행 거리 측정 알고리즘을 구현할 때 확장 칼만 필터(EKF, Extended Kalman Filter)를 기반으로 하는 다중 상태 제약 칼만 필터(MSCKF, Multi-State Constraint Kalman Filter)를 사용하였고, RTAB-Map을 사용하여 3D 지도 작성을 진행하였다. 시각 관성 주행 거리 측정은 관성 측정과 카메라의 특징을 모두 사용하여 이동체의 궤적을 추정하는 프로세스이다. 이는 GPS를 사용하지 않기 때문에, GPS가 거부된 실내 환경 또는 신호가 잘 잡히지 않는 환경에서도 사용할 수 있다.

기존의 시각 관성 주행 거리 측정과 3D 지도 작성의 경우 일반적으로 정면만을 바라보고 실험을 진행하였고, 카메라의 방향은 중요한 것이 아니었다. 하지만, 수중에서 작업하는 로봇의 경우에는 지형 탐사를 위해 바닥을 봐야 하는 등 카메라가 정면을 향하는 것이 부적합한 상황이 생길 수 있다. 이를 위해 본 논문에서는 어느 상황에도 적용할 수 있도록 바닥, 정면, 그리고 천장을 향한 경우에 대해 모두 실험하였고, MATLAB을 통해 결과 및 성능을 비교하였다.

### 제3절 연구 내용

본 논문에서는 이동체의 위치 및 자세 추정을 위해 Ubuntu 환경에서 ROS를 사용하여 시각 관성 주행 거리 측정 알고리즘을 구현하고 RTAB-Map을 사용하여 3D 지도 작성을 진행한다. 연구를 위해 IMU가 내장되어있는 스테레오 카메라를 사용하였고, 성능을 평가하기 위해 실내에서 실험을 진행하였다. 실험에 앞서, 스테레오 카메라 사이의 보정(calibration)과 스테레오 카메라와 IMU 사이의 보정을 통해 왜곡을 줄여 정확도를 높였다.

스테레오 카메라 보정은 두 인접한 카메라의 시야가 겹치도록 하여 두 카메라 시스템의 내부 및 외부 파라미터를 추정하는 것이다. 또한, 스테레오 카메라와 IMU 사이의 보정을 위해 Allan variance를 통하여 IMU의 noise 파라미터를 얻는다. IMU noise 파라미터에는 자이로스코프에 대한 Angle Random Walk(ARW), bias

instability, random walk와 가속도계에 대한 Velocity Random Walk(VRW), bias instability, random walk 값이 포함된다. 미리 만들어놓은 IMU 데이터가 담긴 bag 파일을 사용하여 Allan variance를 계산하고 IMU noise 파라미터를 얻는다. 얻은 IMU의 noise 파라미터를 사용하여 카메라와 IMU 사이의 보정을 진행한다.

보정된 스테레오 카메라와 IMU를 사용하여 시각 관성 주행 거리 측정을 진행한다. 시각 관성 주행 거리 측정을 위해 확장 칼만 필터를 기반으로 하는 다중 상태 제약 칼만 필터를 사용하였다. State 벡터에는 IMU state와 camera state가 포함된다. 시각 관성 주행 거리 측정의 결과로 추정된 위치와 자세가 출력된다. 또한, RTAB-Map을 이용한 3D 지도 작성을 진행한다. 이는 외관 기반 루프 클로저 디텍터를 기반으로 하는 RGB-D, 스테레오 및 라이다 그래프 기반의 SLAM 접근 방식이며, 카메라 데이터를 사용하여 3D 지도와 추정된 위치를 출력한다. 시각 관성 주행 거리 측정과 3D 지도 작성 실험 모두 카메라가 바닥, 정면 그리고 천장을 향하는 3가지 경우로 나누어 진행하였고, MATLAB을 통해 각각의 결과 및 성능을 확인하여 시각 관성 주행 거리 측정의 경우 카메라가 천장을 향하는 경우 성능이 가장 좋고 바닥을 향하는 경우 가장 좋지 않았으며, 3D 지도 작성의 경우 카메라가 천장을 향하는 경우가 가장 좋지 않은 것으로 나타났다. 이는 특징점이 추출되지 않아 시각 관성 주행 거리 측정의 알고리즘에서 다중 상태 제약 칼만 필터의 time update만 진행되고 measurement update는 진행되지 않아 공분산이 계속해서 증가하기 때문에 오차가 커져 성능이 낮아지게 된다. RTAB-Map의 경우 비슷한 환경에서 3D 지도 작성을 진행하면 들어온 이미지가 기존의 이미지인지 새로운 이미지인지 판단할 수 없어 루프 클로저가 이루어지지 않기 때문이다. 실험 결과, 추출된 특징점의 개수가 많다고 해서 성능에 좋은 영향을 끼치는 것이 아님을 증명하였다.

## 제4절 기존 연구

### 1. 정면의 특징 추출을 통한 위치 및 자세 추정

시각 관성 주행 거리 측정은 시각 관성 항법 시스템 등의 비전과 관련된 알고리즘으로 많은 연구가 지속적으로 진행되고 있다. 비전과 관련된 위치 및 자세 추정에 대한 연구는 특징 추출을 위해 점을 사용하는 경우, 선을 사용하는 경우, 그

리고 점과 선을 모두 사용하는 경우 3가지 유형으로 나눌 수 있다. 먼저 Leutenegger[11]는 실시간 Vision-aided 관성 항법을 위한 확장 칼만 필터 기반 알고리즘을 제시한다. 이 논문에서는 여러 카메라 포즈에서 정적 점에 대한 특징이 관찰될 때 발생하는 기하학적 제약을 표현할 수 있는 측정 모델을 도출하였다. 이는 특징 수에서 선형으로만 계산 복잡성을 가지며, 대규모 실제 환경에서 고정밀 포즈 추정이 가능하다. 알고리즘의 성능은 KITTI 데이터를 통해 확인되었고, KITTI 데이터셋은 정면 주행 장면을 기록해 놓은 데이터이다.

G. Kottas[12]는 관성 측정과 선의 시각 관찰을 바탕으로 3D의 움직이는 이동체의 상태를 추정하는 알고리즘을 제안한다. 이 논문에서는 관찰 가능성 특성을 조사하고 하나(2개 이상)의 선이 감지될 때 관측 불가능한 자유도를 5개(4개) 가지고 있음을 증명하였다. 또한, 이 결과를 활용하여 슬라이딩 시간대에 걸쳐 선 measurement를 효율적으로 처리하기 위해 도입된 확장 칼만 필터 추정기의 일관성을 개선하였다. 마지막으로, 이를 정면을 바라본 실험을 통해 검증하였다.

Y. Yang[13]은 3D 라인 삼각측량에 대한 degenerate 모션 분석과 점과 선을 이용한 모노 카메라 시각 관성 주행 거리 측정 알고리즘을 제안한다. 이미지의 세그먼트 측정을 바탕으로 슬라이딩 윈도우 기반의 3D 세그먼트 삼각측량 알고리즘 2가지를 제안하고 성능을 비교한다. 이를 실제 실험과 Euroc Mav 데이터셋으로 검증하였다. Euroc Mav 데이터셋은 드론을 사용하여 기록한 데이터로 공중에서 아래를 비스듬히 보는 이미지로 이루어져 있다.

## 2. 정면을 보고 진행한 3D 지도 작성

3D 지도 작성은 이동체의 자율주행을 위해 필수적인 것으로 주로 라이다 혹은 카메라를 사용한다. 이는 라이다를 사용하는 경우, 카메라를 사용하는 경우, 그리고 라이다와 카메라를 모두 사용하는 경우 3가지로 나눌 수 있다. 먼저 K. Lenac[14]은 3D LiDAR가 제공하는 360° FOV(field of view) 포인트 클라우드에서 추출한 평면 표면 세그먼트 세트를 등록하여 이동체의 경로를 추정하는 빠른 3D 포즈 기반 SLAM 시스템을 제안한다. 이는 실내 실험과 실외 실험을 통해 검증되었으며, 실내 실험은 연구용 이동식 로봇 허스키 A200을 사용하였고, 실외 실험은 자동차에 3D LiDAR를 장착하여 진행하였다. 두 실험 모두 LiDAR가 정면을 향해 바

로 장착되어있다.

J. M. Sáez[15]는 3D 지도 작성을 위한 스테레오 기반 접근법을 제안한다. 연속된 포인트 클라우드 간의 최적의 로컬 정렬은 에너지 최소화에 따른 증분 일치 필터링 프로세스에 의존하는 고속 자기 운동 및 동작 추정 알고리즘에 의해 계산된다. 그 후, 다중 뷰 ICP의 일종인 quasi-random 업데이트 알고리즘은 지도의 전역 불일치를 최소화한다. 이 알고리즘을 검증하기 위해 로봇에 스테레오 카메라를 고정하고 실내 실험을 진행하였다. 카메라는 정면을 향하도록 고정되어있다.

J. Li[16]는 실시간 도로에서의 지도 작성을 위한 LiDAR와 카메라 정보가 크고 정확하게 통합된 3D 시맨틱(semantic) 지도 작성을 제안한다. LiDAR 및 IMU의 다중 센서 융합으로 로봇 위치를 찾기 위해 SLAM이 수행되며, 로봇이 이동하는 동안 주변 장면의 지도가 생성된다. 또한, 환경의 시맨틱 지도를 만들기 위해 합성곱 신경망(CNN, Convolution Neural Network) 기반 시맨틱 세그먼트 이미지가 사용된다. LiDAR와 카메라의 센서 융합을 사용하여 포인트 클라우드의 시맨틱 라벨 프레임워크를 생성한 다음 자세 측면에서 시맨틱 지도를 만든다. 마지막으로, 분류 성능을 향상시켜 시맨틱 지도를 최적화하기 위해 고차 3D 완전 연결 CRF(Conditional Random Field)를 활용하였다. 제안한 알고리즘을 검증하기 위해 KITTI 데이터셋이 사용되었다.

## 제2장 센서 보정

센서 측정에 대한 수많은 요인이 오차의 원인이 된다. 센서를 알맞게 사용하기 위해서는 모든 센서 장치를 보정하여 예상 측정 스케일에 맞도록 정확도를 조정해야 한다[17]. 서로 다른 센서에서 추출한 측정값을 융합하는 경우, 사진의 고정밀 보정이 필요하다[18]. 본 논문에서는 스테레오 카메라와 IMU를 사용하기 때문에 두 카메라 사이의 보정과 스테레오 카메라와 IMU 사이의 보정이 필요하다.

### 제1절 스테레오 카메라의 보정

카메라 보정은 사진 커뮤니티에서 오랫동안 중요한 이슈였다. 컴퓨터 비전 분야에서 더 높은 정확도 측정의 필요성이 증가함에 따라, 컴퓨터 비전 커뮤니티의 연구가 지속적으로 진행되어왔다[19]. 카메라의 정확한 보정은 차원 측정, 입체 투시 깊이 또는 영상의 움직임과 같은 정량적 측정이 수반되는 분야에 특히 중요하다. 또한, 서로 다른 카메라 간의 기하학적 관계를 추정하는 것도 보정의 중요한 측면이다. 이러한 기하학적 관계를 특징짓는 해당 매개변수를 외부 매개변수라고 한다. 실제 카메라는 완벽하지 않고 다양한 문제가 지속적으로 나타난다고 잘 알려져 있다. 기하학적인 측정의 경우 주요 문제는 카메라의 왜곡(distortion)이며, 이는 영상 평면에서 영상 포인트의 위치와 관련 있지만, 영상 품질과는 직접적인 관련이 없다. 예를 들어, 약간 흐린 영상에서의 점의 위치는 여전히 흐린 점의 중심으로 측정할 수 있다. 그러나 점의 위치가 정확하지 않으면 점의 좌표에 따라 달라지는 결과가 정확하지 않을 수 있다[20]. 따라서, 이미지를 이용한 보정은 중요한 임무가 되었다.

스테레오 카메라를 보정하는 목적은 각 카메라의 내부 및 외부 파라미터를 추정하는 것이다. 내부 파라미터는 초점 거리, 이미지 중심, 이미지 스케일링 계수 및 렌즈 왜곡 계수를 포함한 카메라 광학 내부의 특성을 나타내고, 외부 파라미터는 좌표계에 대한 카메라의 위치와 방향을 나타낸다[21]. 이러한 파라미터는 카메라와 관련된 점의 공간 위치를 고려하여 점의 영상 좌표를 도출하는 방법을 결정한다.

카메라 파라미터를 추정하기 위해 주로 그림 2.1과 같은 체커보드 패턴 또는

april grid 패턴을 사용한다. 체커보드 패턴을 사용한 보정은 지난 10년 동안 가장 많이 쓰이는 보정 방법이다[22]. 스테레오 카메라를 실행하여 상하좌우로 움직이고 회전하면서 데이터를 파일로 저장하고, Kalibr 툴박스를 사용하여 저장한 bag 파일로 카메라 보정을 진행한다.

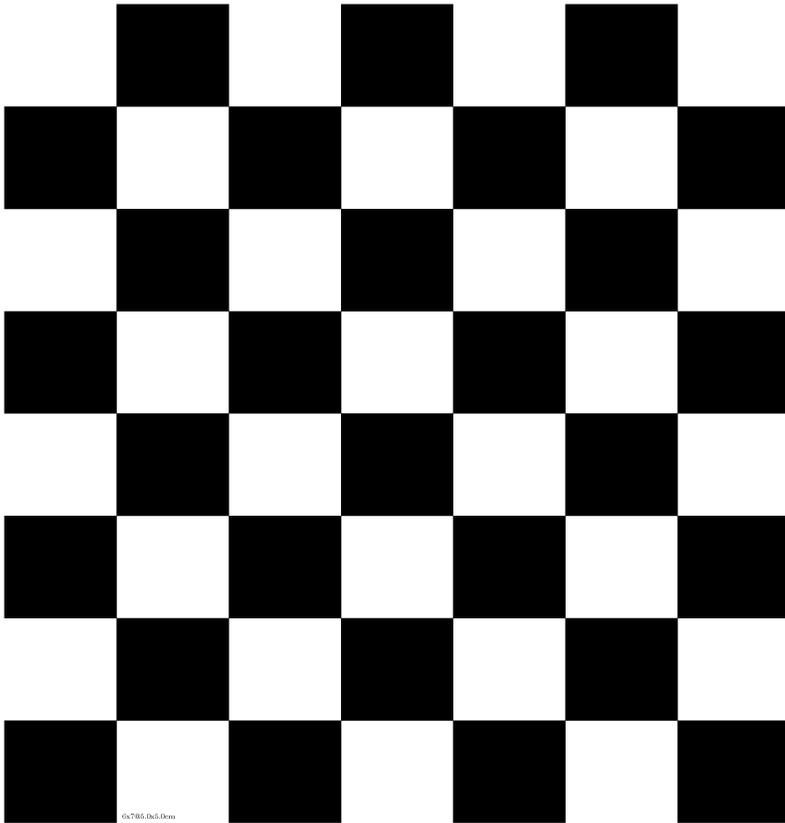


그림 2.1 카메라 보정을 위한 체커보드 패턴

Kalibr 툴박스를 사용하여 그림 2.2의 초점 거리(focal length) ( $f_x, f_y$ ), 주점(principal point) ( $c_x, c_y$ ) 및 왜곡(distortion) 파라미터를 포함하는 카메라 내부 파라미터를 추정함으로써 카메라 사이의 보정을 할 수 있다. 또한, 카메라 외부 파라미터와 각 카메라 프레임 간의 변환도 추정하여 보정 가능하다.

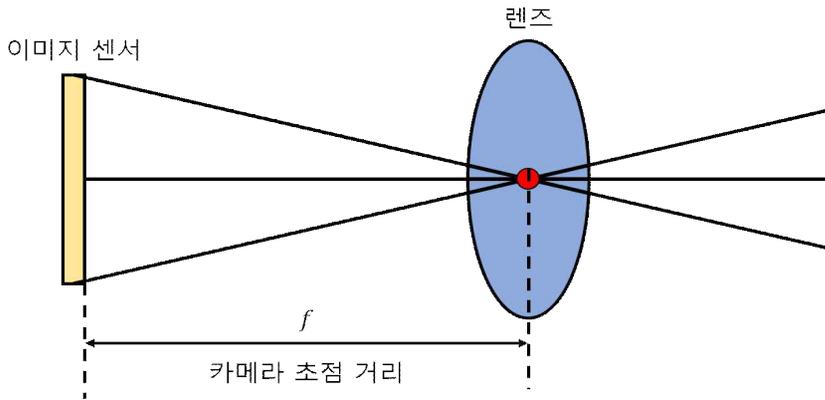


그림 2.2 카메라 초점 거리

Kalibr를 통해 스테레오 카메라를 보정하기 위해서는 다음과 같은 입력이 필요하다.

**[입력 1]** 카메라 데이터를 기록한 bag file : 스테레오 카메라의 각각의 raw 이미지 데이터가 포함된 bag 파일을 만든다. 이는 두 카메라가 동시에 target을 응시하도록 하고, 움직이면서 수집한 데이터이다. 데이터는 raw 데이터이다.

**[입력 2]** target yaml file : bag 파일을 만들 때 사용한 target에 대한 yaml 파일이 필요하다. 이 파일에는 target의 행, 열의 개수 및 tag size(정사각형 하나의 길이), tag와 space에 대한 비율이 적혀있다.

**[입력 3]** 입력 1의 bag 파일에서 사용된 각 카메라의 topic

**[입력 4]** 카메라와 왜곡 모델, 예를 들어 카메라 모델의 경우 pinhole, omnidirectional, double sphere, extended 등이 있고, 왜곡 모델에는 radial-tangential, equidistant, fov 등이 있다.

스테레오 카메라의 보정 후, 카메라의 내부 및 외부 파라미터가 포함된 camchain.yaml 파일을 얻을 수 있다. 이러한 결과를 통해 카메라와 IMU 사이의 보정할 수 있고, 카메라와 IMU 사이의 상대적인 회전, 변환 및 시간 오프셋을 추정

할 수 있다.

## 제2절 스테레오 카메라와 IMU의 보정

여러 센서의 데이터를 융합하는 상태 추정에 대한 방법 중 대부분은 단일 클럭에 대해 모든 측정 타임 스탬프가 정확하게 알려져 있다고 가정한다. 결과적으로, 센서의 시간 동기화는 로봇 시스템을 구축하는 데 있어 중요한 측면이다. 예를 들어, IMU는 동적 모션에 대한 스케일 정보와 견고성을 제공하는 반면, 카메라는 최대 스케일 포즈까지 추정할 수 있다. 이처럼 서로 다른 센서에서 추출한 측정값을 융합하는 경우, 보정 과정이 필요하다.

카메라와 IMU 사이의 회전은 카메라의 시각적 target과 IMU의 중력을 사용하여 두 센서가 모두 수직 방향을 관찰하도록 하여 추정할 수 있다. 스테레오 카메라 보정은 앞 절에서처럼 체커보드와 같은 동일한 시각적 target을 사용하여 동일한 영상을 저장하여 진행할 수 있다. 그림 2.3은 스테레오 카메라와 IMU의 보정 과정을 보여준다.

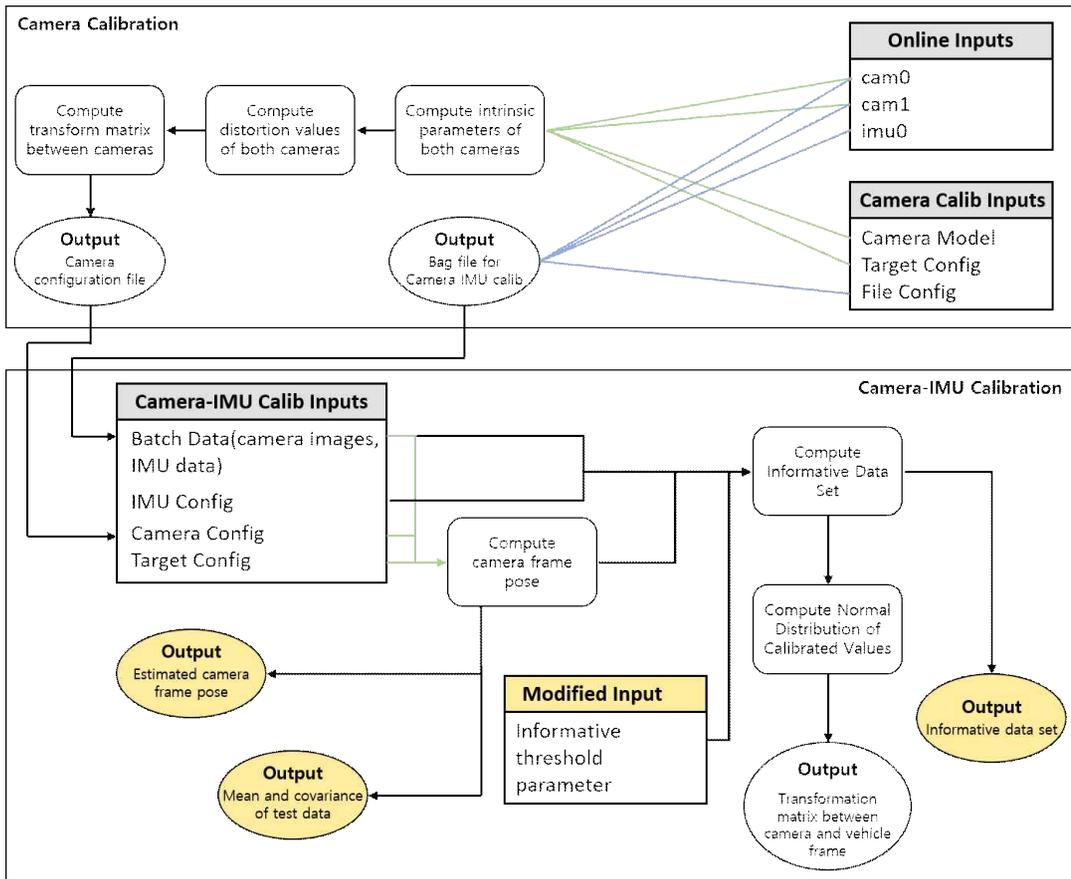


그림 2.3 스테레오 카메라와 IMU 사이의 보정 흐름도 [23]

카메라와 IMU를 보정의 목적은 두 센서 사이의 상대적인 회전, 변환 및 시간 오프셋을 결정하는 것이다. 보정을 수행하기 위해 센서의 헤드가 정적 보정 패턴 앞에서 흔들릴 수 있으므로 일반적으로 1~2분 정도 짧은 시간 간격인  $T = [t_1, t_k]$  동안 데이터를 수집한다. 그림 2.4는 일반적인 좌표계 설정을 나타낸다.

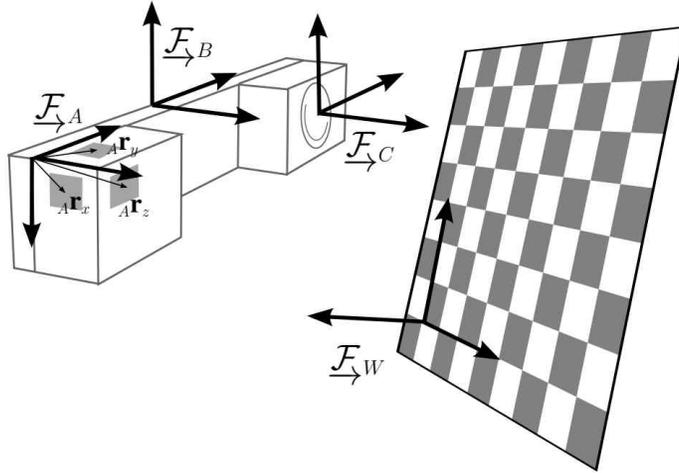


그림 2.4 카메라-IMU 보정 프레임 [23]

$F_W$ 는 기준 좌표계를 나타내고 이는 보정 target의 평면에 설정되고,  $F_A$ 와  $F_C$ 는 각각 IMU 프레임, 카메라 프레임을 나타낸다. 추정은 기준 좌표계인  $F_W$ 에 대해 수행되고, 선형 가속도 및 각속도는 IMU 프레임  $F_A$ 에서 측정된다. 또한, 카메라 프레임  $F_C$ 은 z축이 광축을 아래로 향하도록 하고 카메라 광학 렌즈 중심부에 위치한다[23]. 카메라와 IMU 사이의 보정에 대한 수식은 P. Furgale[24]의 논문을 참고하였다.

## 1. 추정되는 값과 시간 변동 상태의 파라미터화

카메라-IMU 보정 알고리즘은  $F_W$ 에서 표현된 중력 방향  $g_w$ , 카메라와 IMU 사이의 변환( $T_{c,i}$ ) 및 카메라 시간과 IMU 시간 사이의 오프셋( $d$ )를 추정한다. 또한, 시간 변동(time-varying) 상태인 IMU의 포즈( $T_{w,i}(t)$ ), 가속도계 바이어스( $b_a(t)$ ) 및 자이로스코프 바이어스( $b_w(t)$ )도 추정한다.

IMU 포즈는  $6 \times 1$ 의 열벡터로 파라미터화 되고, orientation과 변환에 대해 3 자유도를 갖는다. IMU 프레임에서 world 프레임으로 포인트를 가져오는 변환은 수식 (2.1)로 표현된다.

$$\mathbf{T}_{w,i}(t) = \begin{bmatrix} \mathbf{C}(\psi(t)) & \mathbf{t}(t) \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.1)$$

여기서,  $\psi(t) = \Phi_\psi(t)\mathbf{c}_\psi$ 는 orientation 파라미터를 인코딩하고,  $\mathbf{C}$ 는 파라미터로부터 회전 행렬(rotation matrix)을 만들어주는 함수이다. 그리고  $\mathbf{t}(t) = \Phi_t(t)\mathbf{c}_t$ 는 변환을 인코딩한다. World 프레임에 대한 속도  $\mathbf{v}(t)$ 와 가속도  $\mathbf{a}(t)$ 는 수식 (2.2)와 수식 (2.3)과 같이 나타낼 수 있다.

$$\mathbf{v}(t) = \dot{\mathbf{t}}(t) = \dot{\Phi}_t(t)\mathbf{c}_t \quad (2.2)$$

$$\mathbf{a}(t) = \ddot{\mathbf{t}}(t) = \ddot{\Phi}_t(t)\mathbf{c}_t \quad (2.3)$$

주어진 회전 파라미터에 대해 각속도에 대한 관계에 대한 수식은 다음과 같다. 이때,  $\mathbf{S}$ 는 각속도에 대한 파라미터 비율과 관련된 표준 행렬(standard matrix)이다.

$$\boldsymbol{\omega}(t) = \mathbf{S}(\psi(t))\dot{\psi}(t) = \mathbf{S}(\Phi(t)\mathbf{c}_\psi)\dot{\Phi}(t)\mathbf{c}_\psi \quad (2.4)$$

## 2. Measurement 모델과 Process 모델

시간  $t_j + d$ 에서 표시되는 랜드마크  $\mathbf{p}_w^m$ 의 픽셀 위치는  $\mathbf{y}_{mj}$ 로 표시된다. 여기서  $t_j$ 는 이미지의 타임 스탬프이고  $d$ 는 미지의 시간 오프셋,  $j = 1, 2, \dots, J$ 는 영상의 인덱스이다. 표준 이산 시간 IMU와 카메라 measurement 수식은 다음과 같다.

$$\boldsymbol{\alpha}_k = \mathbf{C}(\psi(t_k))^T(\mathbf{a}(t_k) - \mathbf{g}_w) + \mathbf{b}_a(t_k) + \mathbf{n}_{\alpha_k} \quad (2.5)$$

$$\bar{\mathbf{w}}_k = \mathbf{C}(\psi(t_k))^T\mathbf{w}(t_k) + \mathbf{b}_w(t_k) + \mathbf{n}_{w_k} \quad (2.6)$$

$$\mathbf{y}_{mj} = \mathbf{h}(\mathbf{T}_{c,i}\mathbf{T}_{w,i}(t_j + d)^{-1}\mathbf{p}_w^m) + \mathbf{n}_{ymj} \quad (2.7)$$

앞의 수식에서  $\alpha_k$ 는 가속도계의 measurement,  $\bar{\mathbf{w}}_k$ 는 자이로스코프의 measurement이고 각각  $k=1,2,\dots,K$  일 때 시간  $t_k$ 에서 샘플링된다. 여기서 노이즈  $\mathbf{n}$ 은  $N(0,R)$ 로 정의되고, 다른 노이즈와 통계적으로 독립적이라고 가정할 수 있고,  $\mathbf{h}$ 는 비선형 카메라 모델일 수 있다. 또한  $M$ 개의 랜드마크가 있다 ( $\{p_w^m | m = 1, 2, \dots, M\}$ )[24].

IMU 바이어스는 zero-mean white Gaussian process에 의해 모델링된다. 이는  $E[\mathbf{w}_a(t)\mathbf{w}_w(t')^T] = 0$ 을 따른다.

$$\dot{\mathbf{b}}_a(t) = \mathbf{w}_a(t), \quad \mathbf{w}_a(t) \sim GP(0, \mathbf{Q}_a\delta(t-t')) \quad (2.8)$$

$$\dot{\mathbf{b}}_w(t) = \mathbf{w}_w(t), \quad \mathbf{w}_w(t) \sim GP(0, \mathbf{Q}_w\delta(t-t')) \quad (2.9)$$

### 3. 추정

앞의 1에서 정의된 5가지 값을 추정한다. Measurement와 관련된 오차항은 현재 상태 추정값이 주어졌을 때 측정값과 예측된 측정값의 차로 구성된다. IMU 바이어스에 대한 연속 시간 프로세스 모델은 정수 오차항을 만든다. 따라서, 목적 함수는 다음의 요소로부터 만들어진다[24].

$$\mathbf{e}_{ymj} = \mathbf{y}_{mj} - \mathbf{h}(\mathbf{T}_{c,i}\mathbf{T}_{w,i}(t_j + d)^{-1}\mathbf{p}_w^m) \quad (2.10)$$

$$J_y = \frac{1}{2} \sum_{j=1}^J \sum_{m=1}^M \mathbf{e}_{ymj}^T \mathbf{R}_{ymj}^{-1} \mathbf{e}_{ymj} \quad (2.11)$$

$$\mathbf{e}_{\alpha_k} = \alpha_k - \mathbf{C}(\psi(t_k))^T(\mathbf{a}(t_k) - \mathbf{g}_w) + \mathbf{b}_a(t_k) \quad (2.12)$$

$$J_\alpha = \frac{1}{2} \sum_{k=1}^K \mathbf{e}_{\alpha_k}^T \mathbf{R}_{\alpha_k}^{-1} \mathbf{e}_{\alpha_k} \quad (2.13)$$

$$\mathbf{e}_{w_k} = \bar{\mathbf{w}}_k - \mathbf{C}(\psi(t_k))^T \mathbf{w}(t_k) + \mathbf{b}_w(t_k) \quad (2.14)$$

$$J_w = \frac{1}{2} \sum_{k=1}^K \mathbf{e}_{w_k}^T \mathbf{R}_{w_k}^{-1} \mathbf{e}_{w_k} \quad (2.15)$$

$$\mathbf{e}_{b_a}(t) = \dot{\mathbf{b}}_a(t) \quad (2.16)$$

$$J_{b_a} = \frac{1}{2} \int_{t_1}^{t_K} \mathbf{e}_{b_a}(\tau)^T \mathbf{Q}_a^{-1} \mathbf{e}_{b_a}(\tau) d\tau \quad (2.17)$$

$$\mathbf{e}_{b_w}(t) = \dot{\mathbf{b}}_w(t) \quad (2.18)$$

$$J_{b_w} = \frac{1}{2} \int_{t_1}^{t_K} \mathbf{e}_{b_w}(\tau)^T \mathbf{Q}_w^{-1} \mathbf{e}_{b_w}(\tau) d\tau \quad (2.19)$$

추정을 위해 몇 가지 가정이 필요하다. 카메라와 IMU의 보정 이전에 카메라의 보정이 되어있어야 하고, IMU의 노이즈 및 바이어스 파라미터를 알고 있다고 가정한다. 카메라 보정에 대해서는 앞 절에서 설명했고, IMU의 노이즈 및 바이어스 파라미터에 대해서는 다음 절에서 설명할 것이다.

## 4. 보정

Kalibr를 사용한 카메라와 IMU의 보정에 필요한 입력은 다음과 같다.

**[입력 1]** 두 카메라와 IMU 데이터가 기록된 bag file : 스테레오 카메라의 각 카메라에 target에 대한 이미지와 IMU 데이터를 포함한 ROS bag 파일을 만든다. 카메라의 데이터는 raw 데이터를 사용한다.

**[입력 2]** target yaml file : 카메라를 보정할 때 사용한 target에 대한 yaml 파

일이 필요하다. 이 파일에는 target의 행, 열의 개수 및 tag size(정사각형 하나의 길이), tag와 space에 대한 비율이 적혀있다.

**[입력 3]** camchain file (camchain.yaml) : 카메라 보정을 통해 얻은 카메라의 내부 및 외부 파라미터가 적혀있다.

**[입력 4]** IMU에 대한 noise 파라미터

**[입력 5]** bag 파일에 저장된 카메라와 IMU의 topic

카메라와 IMU의 보정이 끝나면 카메라와 IMU 사이의 최종 추정 변환을 포함한 텍스트 파일과 yaml 파일이 생성된다. 또한, 보정 중에 발견된 추정값 및 오차에 대한 그래프가 출력된다.

### 제3절 Allan Variance

노이즈 분석 방법 중, PSD(Power Spectral Density)와 Allan variance 방법은 확률적 모델보다 일반적인 적용을 위해 관성 시스템 커뮤니티에서 선호하였다. PSD를 사용해서 전달함수를 추정하여 노이즈를 모델링하는 주파수 영역 접근법은 간단하지만, 비전문가에게는 이해하기 어려울 수 있다. 반면 Allan variance는 계산하기 쉽고 해석 및 이해가 비교적 간단할 뿐만 아니라 노이즈 모델링에 있어서 매우 정확하다[25].

Allan Variance 방법은 1966년 David Allan[26]에 의해 제시되었으며, 단순한 분산 분석 방법으로 발전기의 위상 및 주파수 불안정성의 특성화를 위해 널리 쓰이게 되었다. 이는 기본 노이즈 프로세스의 특성을 결정할 수 있으며, 오차 특성을 분석하는 데 적용될 수 있다. Allan variance는 평균 시간 함수로서 RMS(Root Mean Square) 랜덤 드리프트 오차를 나타내고, 데이터 노이즈를 발생시키는 기본 랜덤 프로세스의 특성을 결정하는 데 사용할 수 있다[27]. 또한, 전체 데이터 길이에 대해 특정 연산을 수행함으로써 관성 센서 데이터에서 다양한 유형의 노이즈 향을 특성화하는 데 사용된다. 이 방법의 주요 특징은 오차를 보다 쉽게 특성화하

고 식별하여 전체 노이즈 통계에 대한 기여도를 높일 수 있다는 것이다[25]. Allan Variance 관련 수식은 L. B. Pupo[25]의 논문을 참고하였다.

## 1. Allan Variance의 원리

Allan variance는 클러스터 분석 방법을 기반으로 한다. 데이터는 특정 길이의 그룹으로 분할되고, 연속된  $N$ 개의 데이터 포인트가 주어지면, 각각은  $t_0$ 의 샘플링 시간을 갖는다.  $n(n < N/2)$ 개의 연속된 데이터 포인트의 그룹이 형성되면, 각각의 그룹은 그림 2.5과 같은 클러스터이다[25].

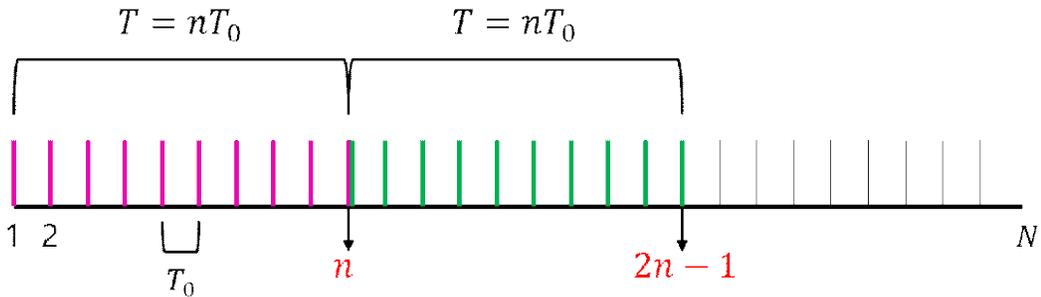


그림 2.5 Allan variance 도출에 사용되는 데이터 구조[25]

각 그룹의 주기는  $T = nT_0$ 이다. 관성 센서의 순간 출력 속도가  $\Omega(t)$ 인 클러스터 평균은 수식 (2.20)과 같다[27].

$$\bar{\Omega}_k(T) = \frac{1}{T} \int_{t_k}^{t_k+T} a(t) dt \quad (2.20)$$

여기서,  $\bar{\Omega}_k(T)$ 는  $k$ 번째 데이터 포인트에서 시작하여  $n$ 개의 데이터 포인트를 포함하는 클러스터의 평균 출력 속도를 나타낸다. 다음 시간의 클러스터 평균은 수식 (2.21)과 같다.

$$\bar{\Omega}_{k+T}(T) = \frac{1}{T} \int_{t_k+T}^{t_k+2T} \Omega(t) dt \quad (2.21)$$

인접한 두 클러스터로부터 계산된 길이  $T$ 를 갖는 Allan variance는 다음과 같고,  $\langle \rangle$ 는 앙상블 평균을 의미한다[28].

$$\sigma^2(T) = \frac{1}{2} \langle [\bar{\Omega}_{k+T}(T) - \bar{\Omega}_k(T)]^2 \rangle = \frac{1}{2(K-1)} \sum_{k=1}^{K-1} [\bar{\Omega}_{k+T}(T) - \bar{\Omega}_k(T)]^2 \quad (2.22)$$

Allan variance의 각 계산에 대한 서로 다른 클러스터 길이 또는 상관 시간 (correlation time)을 선택하면, 상관 시간의 함수인 Allan variance를 얻을 수 있다. log-log plot에서 이 방법은 일반적으로 Allan variance 대  $T$ 의 제곱근으로 나타나는 그래프의 다양한 기울기를 조사하여 오차의 다른 기여도를 구별할 수 있다. 상관 시간의 적절한 값을 선택하면, angle random walk, quantization noise, bias instability, rate random walk 등의 정보를 얻을 수 있다. 다양한 random 프로세스는 각 프로세스가 적절한 기울기를 가지며 일반적으로 곡선의 다른 위치에 나타나므로 쉽게 식별할 수 있다[25][28].

Allan variance는 출력 각도 또는 속도 측면에서 다음과 같이 정의할 수 있다. 여기서, 적분의 하한은 정의에서 각도 또는 속도만 사용되므로 명시되지 않는다.

$$\theta(t) = \int^t \Omega(t) dt \quad (2.23)$$

각도 또는 속도 측정은  $t = kt_0, k = 1, 2, \dots, N$ 으로 주어진 이산 시간으로 수행되어 표기가  $\theta_k = \theta(kt_0)$ 로 간단하다. 수식 (2.20)과 수식 (2.21)을 다음과 같이 다시 쓸 수 있다.

$$\bar{\Omega}_k(T) = \frac{\theta_{k+n} - \theta_k}{T} \quad (3.24)$$

$$\bar{\Omega}_{k+T}(T) = \frac{\theta_{k+2n} - \theta_{k+n}}{T} \quad (2.25)$$

따라서, Allan variance는 수식 (2.26)과 같이 표현된다.

$$\sigma^2(T) = \frac{1}{2T^2(K-1)} \sum_{k=1}^{K-1} (\theta_{k+2n} - 2\theta_{k+n} + \theta_k)^2 \quad (2.26)$$

또한, Allan variance와 PSD 사이의 동치관계(equivalent relation)는 수식 (2.27)과 같이 나타낼 수 있다.

$$\sigma^2(T) = 4 \int_0^\infty df \cdot S_\Omega(f) \frac{\sin^4(\pi f T)}{(\pi f T)^2} \quad (2.27)$$

여기서,  $S_\Omega(f)$ 는  $\Omega(T)$  프로세스의 PSD이고, 이는 시간 내에 정지해 있는 것으로 가정한다.

## 2. 관성 센서 노이즈의 Allan Variance 분석

관성 센서의 가장 일반적인 오차 및 노이즈 소스에는 constant bias error, bias instability, angle random walk(ARW), velocity random walk(VRW), rate angle walk(RRW), quantization noise, rate ramp 그리고 sinusoidal component가 있다. 그중에서 카메라와 IMU의 보정 과정에서 필요한 bias instability, ARW, VRW 및 RRW를 소개할 것이다.

### 가. Bias Instability

센서 바이어스를 설명하는 데 사용되는 두 가지 특성이 있다. 이 두 특성은 양과 음의 입력에 대한 바이어스의 차이인 bias asymmetry와 시간의 유한한 샘플에 걸쳐 계산된 바이어스의 랜덤한 변화인 bias instability이다. 이 효과는 자이로스코프와 가속도계에 영향을 미친다.

$1/f$  노이즈라고도 하는 bias instability와 관련된 PSD는 다음과 같고, 여기서  $B$ 는 bias instability 계수이고  $f_0$ 는  $3dB$  차단 주파수이다.

$$S_{\Omega}(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)\frac{1}{f} & f \leq f_0 \\ 0 & f > f_0 \end{cases} \quad (2.28)$$

Allan variance인 수식 (2.28)에서 bias instability와 관련된 PSD를 대입하고 적분하면 다음과 같은 수식이 얻어진다.

$$\sigma^2(T) = \frac{2B^2}{\pi} \left\{ \ln 2 - \frac{\sin^3 x}{2x^2} (\sin x + 4x \cos x) + C_i(2x) - C_i(4x) \right\} \quad (2.29)$$

여기서,  $x = \pi f_0 T$ 와  $C_i$ 는 코사인-적분 함수이고, 이 수식은 다음과 같이  $T \gg 1/f_0$ 에 대해 단순화할 수 있다[28].

$$\sigma(T) \rightarrow \sqrt{\frac{2\ln 2}{\pi}} B = 0.664B \quad (2.30)$$

Bias instability 값은 그림 2.6과 같이 기울기가 0인 영역의 root Allan variance 그림에서 읽을 수 있다.

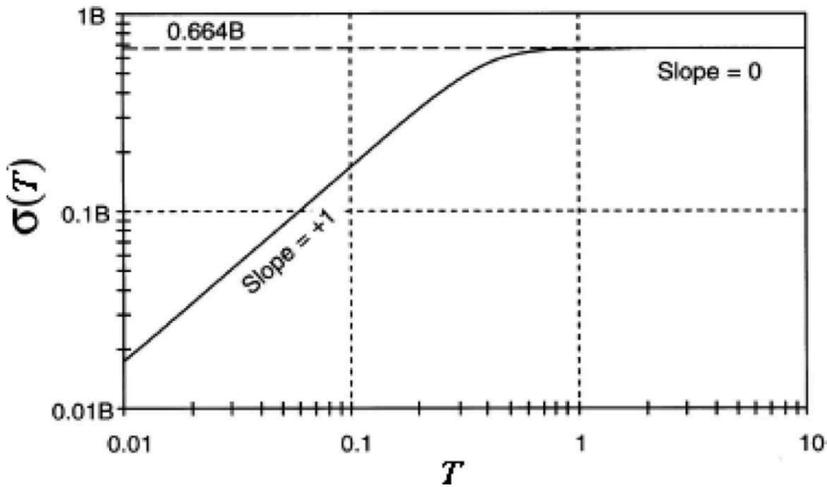


그림 2.6 Bias instability에 대한 그래프[25]

## 나. Angle Random Walk 및 Velocity Random Walk

각도 및 속도 센서는 각각 민감하게 회전 및 이동 속도를 측정한다. 센서 출력 신호는 센서의 샘플링 속도보다 더 큰 속도로 변하는 열역학적 노이즈 유형에 의해 교란된다. 이를 angle random walk (ARW) 및 velocity random walk (VRW)라고 한다. ARW와 VRW는 자이로스코프에 대해  $^{\circ}/\sqrt{h}$ 의 단위, 가속도계에 대해  $m/s/\sqrt{h}$ 의 단위인 노이즈로 각도 및 속도 계산에 직접적으로 사용된다.

ARW와 VRW는 신호가 통합될 때 발생하는 평균 편차 또는 오차를 말한다. 이 오차는 통합 시간에 따라 증가하며, 속도 적분에 기반한 각도 및 속도 측정에 근본적인 한계를 제공한다[29]. 상관 시간이 샘플링 시간보다 훨씬 짧은 고주파 노이즈 항은 자이로스코프 ARM 및 가속도계 VRW에 영향을 미친다. 하지만, 이러한 소스들은 설계로 제거될 수 있다. 노이즈 항은 모두 자이로스코프 또는 가속도계 속도 출력에서 백색 잡음 스펙트럼으로 특징지어진다. PSD와 관련된 수식은 다음과 같다[28].

$$S_{\Omega}(f) = Q^2 \quad (2.31)$$

여기서,  $Q$ 는 단위가  $^{\circ}/h/\sqrt{Hz}$ 인 ARW/VRW 계수이다. ARW/VRW의 PSD 수식 (3.31)을 Allan variance 수식 (2.27)에 대입하고 통합하면 다음과 같이 나타낼 수 있다.

$$\sigma^2(T) = \frac{Q^2}{T} \quad (2.32)$$

$T$ 대  $\sigma(T)$ 의 log-log 그래프에서 ARW/VRW는 그림 2.7과 같이  $-1/2$ 의 기울기를 갖는다.

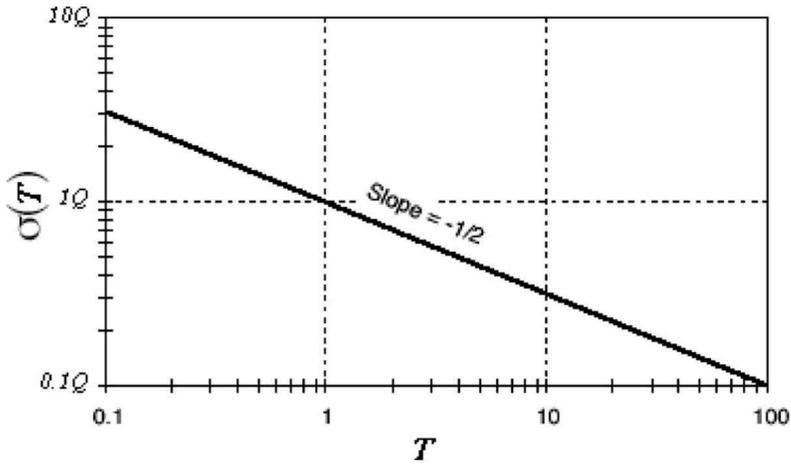


그림 2.7 ARW/VRW 노이즈에 대한 그래프[25]

#### 다. Rate Random Walk

Rate random walk(RRW)는 속도 PSD와 관련된 속도가 수식 (3.33)인 출처를 알 수 없는 오류이며, 여기서  $K$ 는 RRW 계수이다.

$$S_{\Omega}(f) = \left(\frac{K}{2\pi}\right)^2 \frac{1}{f^2} \quad (2.33)$$

수식 (2.27)에 RRW의 PSD 수식 (2.33)을 대입하고 통합하면 다음과 같다.

$$\sigma^2(T) = \frac{K^2 T}{3} \quad (2.34)$$

이는 그림 2.8과 같이  $T$ 대  $\sigma(T)$ 의 log-log 그래프에서 비율 RRW는 1/2의 기울기를 갖는다. 이 노이즈의 크기  $K$ 는  $T=3$ 에 읽을 수 있다.

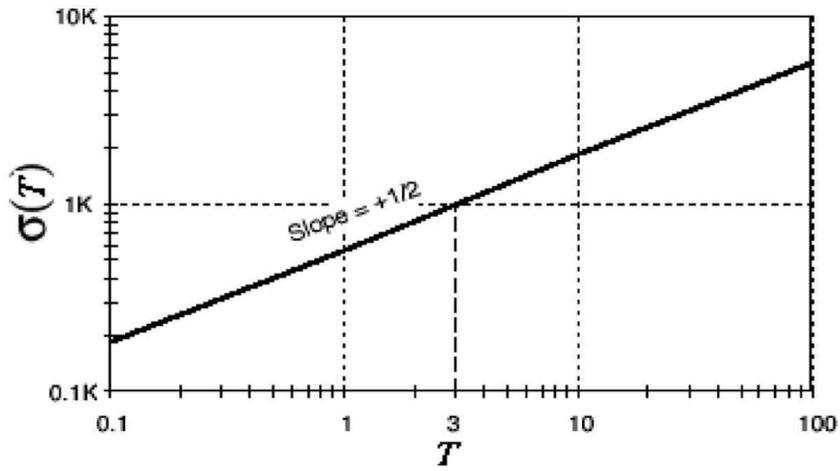


그림 2.8 RRW 노이즈에 대한 그래프[25]

## 제3장 다중 상태 제약 칼만 필터를 사용한 시각 관성 주행 거리 측정

다중 상태 제약 칼만 필터(MSCKF, Multi-State Constraint Kalman Filter)는 고정 슬라이딩 윈도우를 사용하여 3D 움직임을 추적하는 확장 칼만 필터(EKF, Extended Kalman Filter)를 기본으로 하는 알고리즘이다. 이 알고리즘은 뛰어난 성능으로 전문가들로부터 상당한 관심을 받아왔다. 특히, 다중 상태 제약 칼만 필터는 SLAM(Simultaneous Localization And Mapping) 기반 알고리즘에 비해 정밀도가 높고 일관성 있으며 계산의 복잡도가 낮은 것으로 나타난다[30].

다중 상태 제약 칼만 필터는 IMU의 바이어스를 포함한 일반적인 IMU 상태 변수에 이전 시간의 영상이 얻어진 시점에서의 위치 및 자세 즉, 슬라이딩 윈도우(sliding window)를 증가시킨다. 그에 맞춰 시간 전파 모델의 단위행렬이 증가한다. 다중 상태 제약 칼만 필터는 측정치로서 특징점의 2차원 위치를 사용한다. 측정치를 업데이트하는 과정에서는 특징점 전부가 아닌, 특징이 카메라의 시야를 벗어나는 등 추적에 실패하거나 최대 추적 횟수를 초과한 특징점만을 사용한다. 또한, 다중 상태 제약 칼만 필터는 특징점 위치에 대한 측정치 자코비언의 왼쪽 영공간(left null space)을 양변에 곱하여 제거하는데, 이는 다중 상태 제약 칼만 필터의 상태 변수가 특징점의 위치를 포함하지 않기 때문이다[31-32].

일반적인 스테레오 다중 상태 제약 칼만 필터의 알고리즘은 그림 3.1과 같이 나타낼 수 있다. 특징이 추출되고 왼쪽 오른쪽 카메라 이미지 사이에 매칭을 진행한다. 상태는 왼쪽 카메라의 포즈 파라미터로 증강된다. 두 카메라 사이의 보정 파라미터를 고려하여 오른쪽 카메라의 포즈가 왼쪽 카메라에서 계산된다. 마지막으로, 상태는 각 시간 단계에서 두 카메라에서 얻은 측정치에 의해 업데이트된다[33].

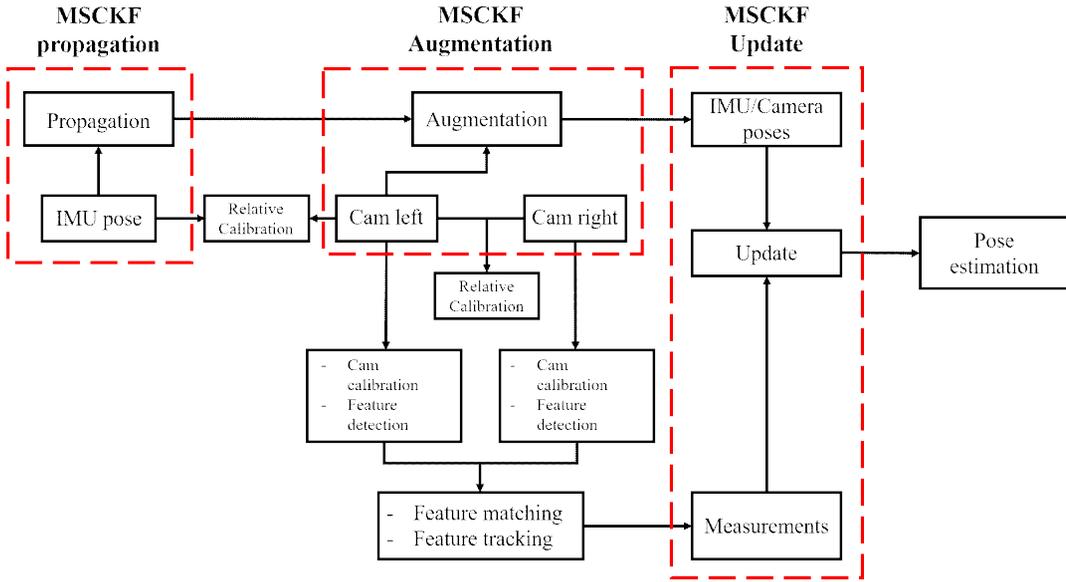


그림 3.1 일반적인 스테레오 다중 상태 제약 칼만 필터의 흐름도

### 제1절 상태 벡터

다중 상태 제약 칼만 필터에 대한 수식은 K. Sun[34]의 논문을 참고하였다. IMU 상태는 다음과 같이 정의할 수 있다.

$$\mathbf{x}_{IMU} = [{}^I_G \mathbf{q}^T \quad \mathbf{b}_g^T \quad {}^G \mathbf{v}_I^T \quad \mathbf{b}_a^T \quad {}^G \mathbf{p}_I^T \quad {}^I_C \mathbf{q}^T \quad {}^I_C \mathbf{p}_C^T]^T \quad (3.1)$$

여기서,  ${}^I_G \mathbf{q}^T$ 는 global 프레임인  $\{G\}$ 에서 IMU 프레임인  $\{I\}$ 로의 회전을 나타내고,  ${}^G \mathbf{v}_I$ 와  ${}^G \mathbf{p}_I$ 는 각각  $\{G\}$ 에 대한 IMU의 속도와 위치를,  $\mathbf{b}_g$ 와  $\mathbf{b}_a$ 는 각각 IMU의 자이로스코프와 가속도계에서 측정된 각속도와 선가속도의 바이어스를 나타낸다. 또한,  ${}^I_C \mathbf{q}$ 와  ${}^I_C \mathbf{p}_C$ 는 카메라 프레임인  $\{C\}$ 에서  $\{I\}$ 로의 회전과 이동을 나타낸다. 본 알고리즘에서는 왼쪽 및 오른쪽 카메라와 관련된 외부 파라미터를 알고 있다고 가정하고, 왼쪽 카메라 프레임을 사용한다. 실제 IMU 상태를 사용하면 상태 벡터의 쿼터니언에 대한 추가 단위 제약 때문에 공분산 행렬에 문제가 발생할 수 있다. 대신 수식 (3.2)로 정의된 오차 IMU 상태는 위치, 속도 및 바이어스에 사용되는 표준 가법 오차(additive error)와 사용된다(예,  ${}^G \tilde{\mathbf{p}}_I = {}^G \mathbf{p}_I - {}^G \hat{\mathbf{p}}_I$ )[34].

$$\tilde{\mathbf{x}}_I = \left[ {}^I_G \tilde{\boldsymbol{\theta}}^T \quad \tilde{\mathbf{b}}_g^T \quad {}^G \tilde{\mathbf{v}}_I^T \quad \tilde{\mathbf{b}}_a^T \quad {}^G \tilde{\mathbf{p}}_I^T \quad {}^I_C \tilde{\boldsymbol{\theta}}^T \quad {}^I_C \tilde{\mathbf{p}}_C^T \right]^T \quad (3.2)$$

쿼터니언의 경우 오차 쿼터니언인  $\delta \mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}$ 은 수식 (3.3)과 같은 오차 상태와 관련 있다.

$$\delta \mathbf{q} \approx \left[ \frac{1}{2} {}^G \hat{\boldsymbol{\theta}}^T \quad 1 \right]^T \quad (3.3)$$

이러한 표현을 사용하면 orientation 오차의 차수가 3으로 감소하여 불확실성을 나타낼 수 있다. 따라서  $N$ 개의 카메라 상태는 상태 벡터에서 함께 고려되기 때문에 각 카메라 오차 벡터가 수식 (3.4)인 전체 오차 상태 벡터는 수식 (3.5)가 된다.

$$\tilde{\mathbf{x}}_{C_i} = \left[ {}^{C_i} \tilde{\boldsymbol{\theta}}^T \quad {}^G \tilde{\mathbf{p}}_{C_i}^T \right]^T \quad (3.4)$$

$$\tilde{\mathbf{x}} = \left[ \tilde{\mathbf{x}}_{IMU}^T \quad \tilde{\mathbf{x}}_{C_1}^T \quad \tilde{\mathbf{x}}_{C_2}^T \quad \cdots \quad \tilde{\mathbf{x}}_{C_N}^T \right]^T \quad (3.5)$$

## 제2절 Process Model

연속 시간에서 IMU 상태의 process model은 다음과 같다.

$$\begin{aligned} {}^I_G \dot{\hat{\mathbf{q}}} &= \frac{1}{2} \Omega(\hat{\boldsymbol{\omega}}) {}^I_G \hat{\mathbf{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3 \times 1}, \\ {}^G \dot{\hat{\mathbf{v}}} &= C({}^I_G \hat{\mathbf{q}})^T \hat{\mathbf{a}} + {}^G \hat{\mathbf{g}}, \\ \dot{\hat{\mathbf{b}}}_a &= \mathbf{0}_{3 \times 1}, \quad {}^G \dot{\hat{\mathbf{p}}}_I = {}^G \hat{\mathbf{v}}, \\ {}^I_C \dot{\hat{\mathbf{q}}} &= \mathbf{0}_{3 \times 1}, \quad {}^I_C \dot{\hat{\mathbf{p}}}_C = \mathbf{0}_{3 \times 1} \end{aligned} \quad (3.6)$$

여기서,  $C$ 는 쿼터니언을 해당 회전 행렬로 변환하는 함수이고,  $\hat{\omega}$ 와  $\hat{a}$ 는 각각  $3 \times 1$ 의 각속도와 가속도에 대한 IMU 측정값이며 이는 바이어스가 제거되었다.

$$\hat{\omega} = \omega_m - \hat{\mathbf{b}}_g \quad (3.7)$$

$$\hat{a} = \mathbf{a}_m - \hat{\mathbf{b}}_a \quad (3.8)$$

또한,  $\Omega(\hat{\omega})$ 은 수식 (3.9)와 같다. 이때,  $[\hat{\omega}_\times]$ 는  $\hat{\omega}$ 의 skew symmetric matrix를 말하고 이는 수식 (3.10)과 같이 표현된다.

$$\Omega(\hat{\omega}) = \begin{bmatrix} -[\hat{\omega}_\times] & \omega \\ -\omega^T & 0 \end{bmatrix} \quad (3.9)$$

$$[\hat{\omega}_\times] = \begin{bmatrix} 0 & -\hat{\omega}_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3.10)$$

수식 (3.6)에 기반한 IMU 오차 상태에 대한 선형 연속 시간 process model은 수식 (3.11)과 같이 나타낼 수 있고,  $\mathbf{n}_{IMU}^T = [\mathbf{n}_g^T \quad \mathbf{n}_{wg}^T \quad \mathbf{n}_a^T \quad \mathbf{n}_{wa}^T]^T$ 이다.  $\mathbf{n}_{IMU}^T$ 에서 벡터  $\mathbf{n}_g$ 와  $\mathbf{n}_a$ 는 자이로스코프 및 가속도계 측정값의 가우시안 노이즈를 나타내며,  $\mathbf{n}_{wg}$ 와  $\mathbf{n}_{wa}$ 는 자이로스코프 및 가속도계 측정값의 바이어스의 random walk rate를 나타낸다.

$$\dot{\tilde{\mathbf{x}}}_{IMU} = \mathbf{F}\tilde{\mathbf{x}}_{IMU} + \mathbf{G}\mathbf{n}_{IMU} \quad (3.11)$$

$$\mathbf{F} = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}_{\times}] & -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ -\mathcal{C}^{(I_G \hat{\mathbf{q}})^T} [\hat{\mathbf{a}}_{\times}] & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathcal{C}^{(I_G \hat{\mathbf{q}})^T} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (3.12)$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathcal{C}^{(I_G \hat{\mathbf{q}})^T} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (3.13)$$

IMU의 이산 시간 measurement를 처리하기 위해서는, IMU 상태를 전파(propagation)하기 위해 수식 (3.6)에 4차 Runge-Kutta 적분을 적용한다. 상태의 불확실성을 전파하기 위해 수식 (3.11)의 이산 시간 상태 전이 행렬과 이산 시간 노이즈 공분산 행렬을 먼저 계산한다.

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau) d\tau\right) \quad (3.14)$$

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau) \mathbf{G} \mathbf{Q} \mathbf{G}^T \boldsymbol{\Phi}^T(t_{k+1}, \tau) d\tau \quad (3.15)$$

여기서,  $\mathbf{Q} = \mathbb{E}[\mathbf{n}_{IMU} \mathbf{n}_{IMU}^T]$ 는 시스템의 연속 시간 노이즈 공분산 행렬이다. 이때 IMU 상태의 전파된 공분산은 다음과 같다.

$$\mathbf{P}_{II_{k+1k}} = \boldsymbol{\Phi}_k \mathbf{P}_{II_{kk}} \boldsymbol{\Phi}_k^T + \mathbf{Q}_k \quad (3.16)$$

전체 상태의 공분산은 수식 (3.17)과 같이 분리함으로써 전체 불확실성 전파는

수식 (3.18)로 표현될 수 있다.

$$P_{k|k} = \begin{bmatrix} P_{II_{k|k}} & P_{IC_{k|k}} \\ P_{IC_{k|k}}^T & P_{CC_{k|k}} \end{bmatrix} \quad (3.17)$$

$$P_{k+1|k} = \begin{bmatrix} P_{II_{k+1|k}} & \Phi_k P_{IC_{k|k}} \\ P_{IC_{k+1|k}}^T & P_{CC_{k|k}} \end{bmatrix} \quad (3.18)$$

새로운 이미지가 들어오면, 상태는 새로운 카메라 상태로 증분되어야 한다. 새로운 카메라 상태의 포즈는 최신 IMU 상태로부터 수식 (3.19)로 계산할 수 있다.

$${}^G\hat{\mathbf{q}} = {}^C\hat{\mathbf{q}} \otimes {}^I\hat{\mathbf{q}}, \quad {}^G\hat{\mathbf{p}}_C = {}^G\hat{\mathbf{p}}_C + C({}^I\hat{\mathbf{q}})^T {}^I\hat{\mathbf{p}}_C \quad (3.19)$$

이 카메라 포즈는 상태 벡터에 추가되며, 이에 따라 공분산 행렬이 수식 (3.20)으로 증분된다. 상태 증분 자코비언  $\mathbf{J}$ 는 다음 수식 (3.21)과 같다[34-35].

$$P_{k|k} = \begin{bmatrix} \mathbf{I}_{21+6N} \\ \mathbf{J} \end{bmatrix} P_{k|k} \begin{bmatrix} \mathbf{I}_{21+6N} \\ \mathbf{J} \end{bmatrix}^T \quad (3.20)$$

$$\mathbf{J} = [\mathbf{J}_I \quad \mathbf{0}_{6 \times 6N}] \quad (3.21)$$

$$\mathbf{J}_I = \begin{bmatrix} C({}^I\hat{\mathbf{q}}) & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ -C({}^I\hat{\mathbf{q}})^T [{}^I\hat{\mathbf{p}}_{C \times}] & \mathbf{0}_{3 \times 9} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix} \quad (3.22)$$

### 제3절 Measurement Model

포즈  $({}^C_i\mathbf{q}, {}^G\mathbf{p}_{C_i})$ 가 갖는 스테레오 카메라에서 관찰되는 단일 특징  $f_j$ 의 경우를 살펴본다. 스테레오 카메라는 좌측 및 우측 카메라에 대해 각각  $({}^C_{i,1}\mathbf{q}, {}^G\mathbf{p}_{C_{i,1}})$  및  $({}^C_{i,2}\mathbf{q}, {}^G\mathbf{p}_{C_{i,2}})$ 로 동시에 표시되는 포즈가 다르다. 상태 벡터는 왼쪽 카메라의 포즈만

포함하지만, 오른쪽 카메라의 포즈는 보정의 외부 파라미터를 사용하여 쉽게 얻을 수 있다. 스테레오 measurement는  $\mathbf{z}_i^j$ 로 수식 (3.23)과 같다.

$$\mathbf{z}_i^j = \begin{bmatrix} u_{i,1}^j \\ v_{i,1}^j \\ u_{i,2}^j \\ v_{i,2}^j \end{bmatrix} = \begin{bmatrix} \frac{1}{C_{i,1}Z_j} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \frac{1}{C_{i,2}Z_j} \end{bmatrix} \begin{bmatrix} C_{i,1}X_j \\ C_{i,1}Y_j \\ C_{i,2}X_j \\ C_{i,2}Y_j \end{bmatrix} \quad (3.23)$$

스테레오 이미지가 적절하게 보정되었다고 가정하면,  $\mathbf{z}_i^j$ 의 차수는  $\mathbb{R}^3$ 으로 줄어들 수 있다. 하지만  $\mathbb{R}^4$ 에서  $\mathbf{z}_i^j$ 를 표현하면, 스테레오 이미지에서 동일한 특징의 observation이 더 이상 동일한 이미지 평면에 있을 필요가 없으므로 스테레오 보정(rectification)의 필요성이 없어진다. 수식 (3.23)에서  $k \in \{1, 2\}$ 일 때,  $(C_{i,k}X_j, C_{i,k}Y_j, C_{i,k}Z_j)^T$ 는 카메라 포즈와 관련 있는 왼쪽 및 오른쪽 카메라 프레임  $C_{i,1}$ 과  $C_{i,2}$ 에서 특징  $f_j$ 의 위치를 나타낸다. 각 카메라에서 특징의 위치는 다음과 같다.

$${}^{C_{i,1}}\mathbf{p}_j = \begin{bmatrix} C_{i,1}X_j \\ C_{i,1}Y_j \\ C_{i,1}Z_j \end{bmatrix} = \mathcal{C}({}^{C_{i,1}}\mathbf{q})({}^G\mathbf{p}_j - {}^G\mathbf{p}_{C_{i,1}}) \quad (3.24)$$

$${}^{C_{i,2}}\mathbf{p}_j = \begin{bmatrix} C_{i,2}X_j \\ C_{i,2}Y_j \\ C_{i,2}Z_j \end{bmatrix} = \mathcal{C}({}^{C_{i,2}}\mathbf{q})({}^G\mathbf{p}_j - {}^G\mathbf{p}_{C_{i,2}}) \quad (3.25)$$

World 프레임에서 특징 위치  ${}^G\mathbf{p}_j$ 는 현재 추정된 카메라 포즈를 기반으로 [11]의 최소 제곱 방법을 사용하여 계산한다. 현재 추정치에서 측정 모델을 선형화하면 측정의 잔차(residual)는 수식 (3.26)으로 근사화할 수 있다.

$$\mathbf{r}_i^j = \mathbf{z}_i^j - \hat{\mathbf{z}}_i^j = \mathbf{H}_{C_i}^j \tilde{\mathbf{x}}_{C_i} + \mathbf{H}_{f_i}^j {}^G\hat{\mathbf{p}}_j + \mathbf{n}_i^j \quad (3.26)$$

$$\mathbf{H}_{C_i}^j = \frac{\partial \mathbf{z}_i^j}{\partial C_{i,1} \mathbf{p}_j} \cdot \frac{\partial C_{i,1} \mathbf{p}_j}{\partial \mathbf{x}_{C_{i,1}}} + \frac{\partial \mathbf{z}_i^j}{\partial C_{i,2} \mathbf{p}_j} \cdot \frac{\partial C_{i,2} \mathbf{p}_j}{\partial \mathbf{x}_{C_{2,1}}} \quad (3.27)$$

$$\mathbf{H}_{f_i}^j = \frac{\partial \mathbf{z}_i^j}{\partial C_{i,1} \mathbf{p}_j} \cdot \frac{\partial C_{i,1} \mathbf{p}_j}{\partial G \mathbf{p}_j} + \frac{\partial \mathbf{z}_i^j}{\partial C_{i,2} \mathbf{p}_j} \cdot \frac{\partial C_{i,2} \mathbf{p}_j}{\partial G \mathbf{p}_j} \quad (3.28)$$

where

$$\frac{\partial \mathbf{z}_i^j}{\partial C_{i,1} \mathbf{p}_j} = \frac{1}{C_{i,1} \hat{Z}_j} \begin{bmatrix} 1 & 0 & -\frac{C_{i,1} \hat{X}_j}{C_{i,1} \hat{Z}_j} \\ 0 & 1 & -\frac{C_{i,1} \hat{Y}_j}{C_{i,1} \hat{Z}_j} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.29)$$

$$\frac{\partial \mathbf{z}_i^j}{\partial C_{i,2} \mathbf{p}_j} = \frac{1}{C_{i,2} \hat{Z}_j} \begin{bmatrix} 1 & 0 & -\frac{C_{i,2} \hat{X}_j}{C_{i,1} \hat{Z}_j} \\ 0 & 1 & -\frac{C_{i,2} \hat{Y}_j}{C_{i,1} \hat{Z}_j} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.30)$$

$$\frac{\partial C_{i,1} \mathbf{p}_j}{\partial \mathbf{x}_{C_{i,1}}} = ([C_{i,1} \hat{\mathbf{p}}_{j \times}] - C(G^{C_{i,1}} \hat{\mathbf{q}})) \quad (3.31)$$

$$\frac{\partial C_{i,1} \mathbf{p}_j}{\partial G \mathbf{p}_j} = C(G^{C_{i,1}} \hat{\mathbf{q}}) \quad (3.32)$$

$$\frac{\partial C_{i,2} \mathbf{p}_j}{\partial \mathbf{x}_{C_{i,1}}} = C(G^{C_{i,2}} \hat{\mathbf{q}})^T ([C_{i,1} \hat{\mathbf{p}}_{j \times}] - C(G^{C_{i,1}} \hat{\mathbf{q}})) \quad (3.33)$$

$$\frac{\partial C_{i,2} \mathbf{p}_j}{\partial G \mathbf{p}_j} = C(G^{C_{i,1}} \hat{\mathbf{q}})^T C(G^{C_{i,2}} \hat{\mathbf{q}}) \quad (3.34)$$

여기서,  $\mathbf{n}_i^j$ 는 measurement 노이즈이고,  $\mathbf{H}_{C_i}^j$ 와  $\mathbf{H}_{f_i}^j$ 는 measurement 자코비언이다. 이는 chain rule에 의해 구해진다. 수식 (3.29)에서 (3.34)까지는 수식 (3.27)과 (3.28)에 포함된 항의 값이다. [11]에서 설명한 바와 같이,  ${}^G\mathbf{p}_j$ 는 카메라 포즈를 사용하여 계산되기 때문에  ${}^G\mathbf{p}_j$ 의 불확실성은 그 상태에서 카메라 포즈와 관련이 있다.  ${}^G\mathbf{p}_j$ 의 불확실성이 잔차에 영향을 미치지 않기 위해 수식 (3.26)의 잔차는  $\mathbf{H}_f^j$ 의 null 공간, 즉  $\mathbf{V}$ 에 투영된다. 투영된 잔차는 수식 (3.35)와 같다.

$$\mathbf{r}_o^j = \mathbf{V}^T \mathbf{r}^j = \mathbf{V}^T \mathbf{H}_{\mathbf{x}}^j \tilde{\mathbf{x}} + \mathbf{V}^T \mathbf{n}^j = \mathbf{H}_{\mathbf{x},o}^j \tilde{\mathbf{x}} + \mathbf{n}_o^j \quad (3.35)$$

EKF의 업데이트 단계는 수식 (3.35)에 기반하여 표준적인 방법으로 수행한다.

## 제4절 확장 칼만 필터 업데이트

앞의 절까지는 스테레오 카메라 포즈에서 정적 특징을 관찰하여 부과되는 기하학적 제약을 표현하는 측정 모델을 제시했다. 이번 절에서는 여러 특징을 관찰하는 데 따른 제약이 사용되는 확장 칼만 필터의 업데이트 단계를 자세히 설명한다.

지연된 measurement 업데이트는 두 가지 유형이 있다. measurement 업데이트 단계는 알고리즘이 특징을 잃거나 상태의 카메라 포즈 수가 한계에 도달하면 실행된다[11]. 버퍼가 가득 차면 카메라 상태의 3분의 1이 제한되며, 이는 실시간 구현에서 계산 로드의 갑작스러운 점프를 유발할 수 있다. 계산을 평균화하기 위해 각 타임 스텝에서 하나의 카메라 상태를 제외시키는 것이 바람직하다. 그러나 한 카메라 상태에서 특징의 measurement를 제거하는 것은 observation에 카메라 상태 간의 상대적 변환에 대한 정보가 포함되어있지 않기 때문에 다중 상태 제약 칼만 필터의 프레임워크에 실용적이지 않다. 수학적으로 이는 수식 (3.35)에서  $\mathbf{H}_{f_i}^j$ 의 null 공간의 부분 공간이라는 사실에 기인하고, 이는 수식 (3.35)를 기반으로 한 measurement 모델이 된다[34].

본 알고리즘에서는 업데이트 단계마다 두 카메라의 상태가 제거되며, 두 카메라 상태에서 얻은 모든 특징 observation은 measurement 업데이트에 사용된다. 앞

서 언급한 이유로 인해 두 카메라 상태의 스테레오 measurement는 동일한 특징일 경우에만 유용하다. 카메라 상태를 자주 제거하면 일부 유효한 observation이 무시될 수 있다.

주어진 타임 스텝에서 선택된  $L$ 개의 특징의 제약 조건이 처리되어야 한다. 앞 절에서 설명한 과정에 따라  $j = 1, 2, 3, \dots, L$ 일 때 잔차 벡터  $\mathbf{r}_o^j$ 와 이러한 각 특징에 대한 해당 measurement 자코비언  $\mathbf{H}_{\mathbf{x},o}^j$ 를 수식 (3.35)를 통해 계산한다. 모든 잔차를 단일 벡터에 쌓으면 다음 수식을 얻을 수 있다.

$$\mathbf{r}_o = \mathbf{H}_x \tilde{\mathbf{x}} + \mathbf{n}_o \quad (3.36)$$

여기서,  $\mathbf{r}_o$ 와  $\mathbf{n}_o$ 은 각각  $j = 1, 2, 3, \dots, L$ 일 때 블록 원소  $\mathbf{r}_o^j$ 와  $\mathbf{n}_o^j$ 를 갖는 벡터이고,  $\mathbf{H}_x$ 는  $j = 1, 2, 3, \dots, L$ 일 때 블록 행  $\mathbf{H}_x^j$ 를 갖는 행렬이다.

특징 측정치는 통계적으로 독립적이기 때문에, 노이즈 벡터  $\mathbf{n}_o^j$ 는 상관관계가 없다. 따라서 노이즈 벡터  $\mathbf{n}_o$ 의 공분산 행렬은  $\mathbf{R}_o = \sigma_{im}^2 \mathbf{I}_d$ 이고, 여기서  $d$ 는  $\mathbf{r}_o$ 의 차수이다. 이때 실제로 문제가 발생할 수 있는데, 이는  $d$ 가 상당히 큰 숫자일 수 있다는 것이다. 예를 들어, 10개의 카메라 포즈에서 각각 10개의 특징이 보이면 잔차의 차수는 170이다. 따라서 확장 칼만 필터의 업데이트에 대한 계산 복잡성을 줄이기 위해 행렬  $\mathbf{H}_x$ 의 QR 분해를 사용한다[36]. 따라서  $\mathbf{H}_x$ 를 다음과 같이 분해할 수 있다.

$$\mathbf{H}_x = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{T}_H \\ 0 \end{bmatrix} \quad (3.37)$$

여기서,  $\mathbf{Q}_1$ 과  $\mathbf{Q}_2$ 는 열이 각각  $\mathbf{H}_x$ 의 범위와 null 공간에 대한 기준을 형성하는 단위행렬이고,  $\mathbf{T}_H$ 는 상위 삼각 행렬이다. 이 정의로 수식 (3.37)은 다음과 같은 수식을 도출한다.

$$\mathbf{r}_o = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{T}_H \\ 0 \end{bmatrix} \tilde{\mathbf{x}} + \mathbf{n}_o \quad (3.38)$$

$$\begin{bmatrix} \mathbf{Q}_1^T \mathbf{r}_o \\ \mathbf{Q}_2^T \mathbf{r}_o \end{bmatrix} = \begin{bmatrix} \mathbf{T}_H \\ 0 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{Q}_1^T \mathbf{n}_o \\ \mathbf{Q}_2^T \mathbf{n}_o \end{bmatrix} \quad (3.39)$$

수식 (3.39)를 통해  $\mathbf{H}_x$ 의 범위의 기저 벡터에 잔차  $\mathbf{r}_o$ 를 투영함으로써 측정치에서의 모든 유용한 정보를 유지한다는 것이 분명해진다. 잔차  $\mathbf{Q}_2^T \mathbf{r}_o$ 는 노이즈일 뿐이고, 완전히 없앨 수 있다. 이러한 이유로 수식 (3.36)에 나타난 잔차 대신 확장 칼만 필터 업데이트에 다음과 같은 잔차를 사용한다.

$$\mathbf{r}_n = \mathbf{Q}_1^T \mathbf{r}_o = \mathbf{T}_H \tilde{\mathbf{x}} + \mathbf{n}_n \quad (3.40)$$

위의 수식에서  $\mathbf{n}_n = \mathbf{Q}_1^T \mathbf{n}_o$ 는 공분산 행렬이  $\mathbf{R}_n = \mathbf{Q}_1^T \mathbf{R}_o \mathbf{Q}_1 = \sigma_{im}^2 \mathbf{I}_r$ 이고,  $r$ 은  $\mathbf{Q}_1$ 의 열 개수이다. 확장 칼만 필터 업데이트는 Kalman gain을 계산하는 방식으로 진행한다.

$$\mathbf{K} = \mathbf{P} \mathbf{T}_H^T (\mathbf{T}_H \mathbf{P} \mathbf{T}_H^T + \mathbf{R}_n)^{-1} \quad (3.41)$$

마지막으로, 상태 공분산 행렬은 다음 수식에 따라 업데이트된다.

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I}_\xi - \mathbf{K} \mathbf{T}_H) \mathbf{P}_{k+1|k} (\mathbf{I}_\xi - \mathbf{K} (\mathbf{T}_H)^T) + \mathbf{K} \mathbf{R}_n \mathbf{K}^T \quad (3.42)$$

## 제5절 이미지 프로세싱

이미지에서 특징점을 감지하는 것은 광학 흐름 계산, 모션의 구조, 스테레오 이미지 쌍에서의 객체 추적 등을 포함한 머신 비전의 대부분 분야에 필수적이다. 이 접근 방식이 많이 사용되는 이유는 특징점이 이미지 변위를 안정적으로 계산할 수 있는 충분한 제약을 제공하고, 특징점을 처리함으로써 데이터가 원본 이미지 데이터에 비해 몇 배나 줄어들기 때문이다. 이는 실시간 실행에 특히 중요하다. 가장 직관적인 특징점 유형 중 하나는 점점이다. 모서리는 강력한 2차원 강도 변화를 보여주는 이미지 포인트이므로 인접 포인트와 잘 구분된다. 모서리 디텍터는

정보가 많은 이미지 위치에 대응하고 이미지 간에 안정적으로 일치할 수 있기 때문에 특징점 디텍터로 널리 사용되어왔다. 따라서, 포인트 특징을 추출하기 위해 모서리의 특징점을 추출하는 FAST(Features from Accelerated Segment) 알고리즘을 사용하였고, 스테레오 특징 매칭을 위해 KLT(Kanade-Lucas-Tomasi) 알고리즘을 사용하였다[35-36]. 마지막으로 특이치 제거를 위해 RANSAC 알고리즘을 참고하였다.

## 1. 모서리 검출을 위한 FAST 알고리즘

모서리 검출을 위한 FAST 알고리즘은 M. Trajković[35]의 논문을 참고하였다. 먼저, 수평 및 수직 강도 변화(intensity variation)인  $r_H$ 와  $r_V$ 를 계산한다.

$$r_H = (f_A - f_C)^2 + (f_{A'} - f_C)^2 \quad (3.43)$$

$$r_V = (f_B - f_C)^2 + (f_{B'} - f_C)^2 \quad (3.44)$$

여기서,  $C$ 는 그림 3.2에서와 같이 중심점을 나타내고,  $f_A, f_B, f_{A'}$ , 그리고  $f_{B'}$ 는 각 점에서의 이미지 강도를 나타낸다.

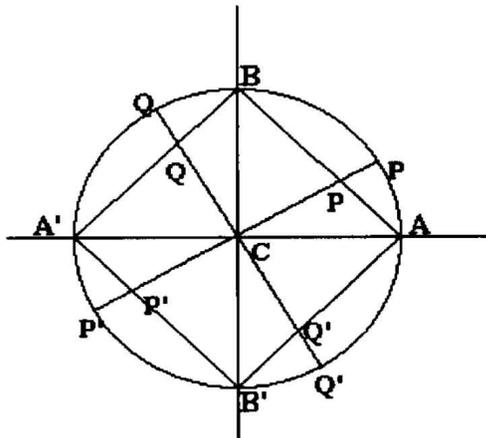


그림 3.2 선형 및 원형 픽셀 간 위치를 보여주는 핵 C의 1차 인접[35]

또한, CRF(Corner Response Function)는 다음과 같이 계산될 수 있다.

$$R = \min(r_H, r_V) \quad (3.45)$$

만약  $R$ 이 주어진 임계값보다 작아지면, 핵은 중심점이 아니므로 더 이상의 계산이 필요하지 않다. 하지만  $R$ 이 주어진 임계값보다 크면 대각선 모서리를 확인하기 위해 픽셀 간 근사치가 적용된다.

### 가. 선형 픽셀 간 근사치

CRF는 수식 (3.46)과 같이 사각형  $ABA'B'$ 를 따라 계산되고, 이때  $x$ 는 사각형 위 포인트의 위치를 결정하는 파라미터이다.  $P$ (혹은  $Q$ )와  $P'$ (혹은  $Q'$ )는 그림 3.2와 같이  $C$ 에 대해 반대에 위치한다. 응답함수는 수식 (3.47)과 (3.48)과 같다.

$$R = \min_{x \in (0, 1)} (r_1(x), r_2(x)) \quad (3.46)$$

$$r_1(x) = (f_P - f_C)^2 + (f_{P'} - f_C)^2 \quad (3.47)$$

$$r_2(x) = (f_Q - f_C)^2 + (f_{Q'} - f_C)^2 \quad (3.48)$$

픽셀 간의 위치 강도는 해당 끝점 강도의 선형 조합으로 계산되고, 다음과 같이 나타낼 수 있다.

$$f_P = (1 - x)f_A + xf_B$$

$$f_{P'} = (1 - x)f_{A'} + xf_{B'}$$

$$f_Q = (1 - x)f_{A'} + xf_B$$

$$f_{Q'} = (1 - x)f_A + xf_{B'}$$

(3.49)

$r_1(0) = r_2(0) = r_H$ 이고,  $r_1(1) = r_2(1) = r_V$ 이다. 수식 (3.49)를 수식 (3.47)과 (3.48)에 대입하면 다음과 같은 수식을 얻을 수 있다. 수식 (3.51)에서 (3.55)는 수식 (3.50)에 사용된 파라미터들을 나타낸다.

$$r_1(x) = A_1x^2 + 2B_1x + C, \quad r_2(x) = A_2x^2 + 2B_2x + C \quad (3.50)$$

where

$$C = r_H \quad (3.51)$$

$$B_1 = (f_B - f_A)(f_A - f_C) + (f_{B'} - f_{A'})(f_{A'} - f_C) \quad (3.52)$$

$$B_2 = (f_B - f_{A'})(f_{A'} - f_C) + (f_{B'} - f_A)(f_A - f_C) \quad (3.53)$$

$$A_1 = r_V - r_H - 2B_1 \quad (3.54)$$

$$A_2 = r_V - r_H - 2B_2 \quad (3.55)$$

$B = \min(B_1, B_2)$ ,  $A = r_V - r_H - 2B$ 라고 하면, CRF는 사각형  $ABA'B'$  위의 최솟값을 갖고,  $B < 0$ ,  $A + B > 0$ 이다. 또한, 최솟값은 수식 (3.56)과 같다.

$$R = C - \frac{B^2}{A} \quad (3.56)$$

만약  $B < 0$ ,  $A + B > 0$ 를 만족하지 않으면, 최솟값을 계산하기 위해 수식 (3.45)를 사용한다.

## 나. 원형 픽셀 간 근사치

원형의 경우, CRF는 그림 3.2에서 원  $ABA'B'$ 를 따라 계산되고, 다음과 같이

나타난다. 또한, 응답함수는 앞에서 설명한 바와 같다[35-36].

$$R = \min_{\alpha \in (0, \pi/2)} (r_1(\alpha), r_2(\alpha)) \quad (3.57)$$

$$r_1(\alpha) = (f_P - f_C)^2 + (f_{P'} - f_C)^2 \quad (3.58)$$

$$r_2(\alpha) = (f_Q - f_C)^2 + (f_{Q'} - f_C)^2 \quad (3.59)$$

픽셀 간의 위치 강도는 다음 수식을 따른다.

$$\begin{aligned}
 f_P - f_C &= (f_A - f_C)\cos(\alpha) + (f_B - f_C)\sin(\alpha) \\
 f_{P'} - f_C &= (f_{A'} - f_C)\cos(\alpha) + (f_{B'} - f_C)\sin(\alpha)
 \end{aligned} \quad (3.60)$$

$$f_Q - f_C = (f_{A'} - f_C)\cos(\alpha) + (f_B - f_C)\sin(\alpha)$$

$$f_{Q'} - f_C = (f_A - f_C)\cos(\alpha) + (f_{B'} - f_C)\sin(\alpha)$$

선형에서와 같이  $r_1(0) = r_2(0) = r_H$ 이고,  $r_1(\pi/2) = r_2(\pi/2) = r_V$ 이다. 수식 (3.60)을 수식 (3.58)과 (3.59)에 대입하면 다음과 같다.

$$r_1(\alpha) = A \cos(2\alpha) + B_1 \sin(2\alpha) + c \quad (3.61)$$

$$r_2(\alpha) = A \cos(2\alpha) + B_2 \sin(2\alpha) + c \quad (3.62)$$

where

$$A = \frac{r_H - r_V}{2}, \quad B = \frac{r_H + r_V}{2} \quad (3.63)$$

$$B_1 = (f_A - f_C)(f_B - f_C) + (f_{A'} - f_C)(f_{B'} - f_C) \quad (3.64)$$

$$B_2 = (f_{A'} - f_C)(f_B - f_C) + (f_A - f_C)(f_{B'} - f_C) \quad (3.65)$$

$B = \min(B_1, B_2)$ 를 정의하면 CRF가 최댓값을 갖는다는 것을 쉽게 보여줄 수 있고,  $B < 0$ 이고 CRF는 수식 (3.66)과 같아진다.

$$R = C - \sqrt{A^2 + B^2} \quad (3.66)$$

앞서 말한 것과 같이,  $B \geq 0$ 이면  $R$ 을 계산하기 위해 수식 (3.45)를 사용한다. 지금까지 설명한 FAST 알고리즘을 단계별로 정리하면 다음과 같다[35].

**[단계 1]** 저해상도 영상에서 모든 픽셀 위치에서 단순 CRF인 수식 (3.45)를 계산한다. 지정된 임계값( $T_1$ )보다 높은 응답을 가진 픽셀을 ‘잠재적 모서리’로 분류한다.

**[단계 2]** 각 잠재적 픽셀에 대해 전체 해상도 이미지를 사용한다. 1) 수식 (3.45)를 사용하여 CRF를 계산한다. 응답이 다른 임계값( $T_2$ )보다 낮으면 픽셀이 모서리가 아니므로 2)를 수행하지 않는다. 2) 앞서 설명한 대로 픽셀 간 근사치를 사용하여 새 응답을 계산한다. 응답이 임계값( $T_2$ )보다 낮으면 픽셀은 모서리가 아니다.

**[단계 3]** 최대 CRF를 갖는 픽셀을 찾아 모서리로 표시한다. 이 단계는 모서리 부근에서 둘 이상의 포인트에서 높은 CRF를 갖고, 가장 큰 포인트만 모서리 포인트로 선언되기 때문에 필요하다.

## 2. 스테레오 특징 매칭을 위한 KLT 알고리즘

스테레오 특징 매칭을 위한 KLT 알고리즘은 B. D. Lucas[36]의 논문을 참고하였다. 변환 이미지 정합(registration) 문제는 다음과 같이 특징지을 수 있다. 그

림 3.3과 같이 두 이미지의 각 위치  $x$ 에서 각각의 픽셀값을 제공하는 함수  $F(x)$ 와  $G(x)$ 가 주어진다. 일부 관심 영역  $R$ 에서 벡터  $x$ 에 대해  $F(x+h)$ 와  $G(x)$ 의 차이를 최소화하는 변이 벡터  $h$ 를 찾아야 한다.  $h$ 의 근사치는 충분히 작아야 한다 [36].

$$F(x) = \frac{F(x+h) - F(x)}{h} = \frac{G(x) - F(x)}{h} \quad (3.67)$$

$$h \approx \frac{G(x) - F(x)}{F(x)} \quad (3.68)$$

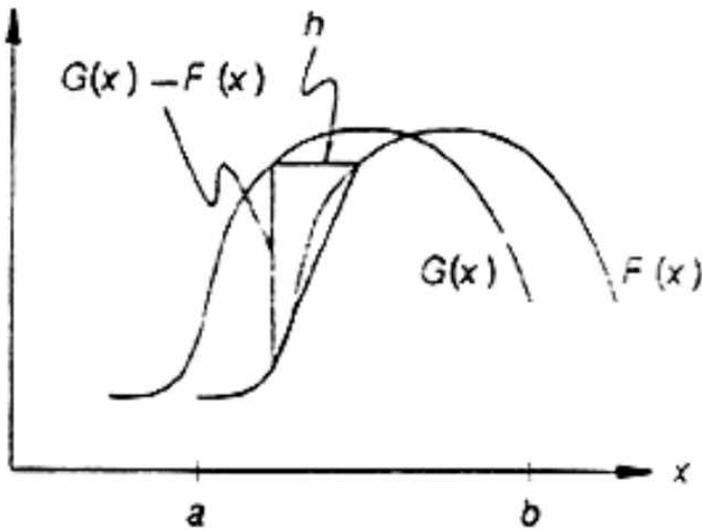


그림 3.3 일치시킬 두 곡선 [37]

수식 (3.68)에 나타난  $h$ 에 대한 근사치는  $x$ 에 따라 달라진다.  $x$ 의 다양한 값을 합치는 간단한 방법은 다음과 같이 평균을 내는 것이다.

$$h \approx \frac{\sum_x \frac{G(x) - F(x)}{F(x)}}{\sum_x 1} \quad (3.69)$$

$F(x)$ 가 선형인 경우 수식 (3.67)의 선형 근사치가 좋고,  $|F''(x)|$ 가 큰 경우에는 좋지 않다. 따라서, 각 항의 기여도를  $|F''(x)|$ 의 추정치에 반비례하여 수식 (3.69)의 평균에 가중치를 부여할 수 있다. 추정치 중 하나는 다음과 같다.

$$F'(x) \approx \frac{G'(x) - F'(x)}{h} \quad (3.70)$$

추정치는 평균에서 가중치로 사용되기 때문에,  $1/h$ 의 상수 부분을 없애고 수식 (3.71)과 같은 가중치 함수로 사용할 수 있다. 따라서, 가중치를 갖는  $h$ 의 추정치 평균은 수식 (3.72)와 같다.

$$w(x) = \frac{1}{|G(x) - F(x)|} \quad (3.71)$$

$$h \approx \frac{\sum_x \frac{w(x)[G(x) - F(x)]}{F(x)}}{\sum_x w(x)} \quad (3.72)$$

그런 다음,  $F(x)$ 를  $h$ 의 추정치로 이동시키고 이 과정을 반복하여 Newton-Raphson 반복(iteration) 유형을 얻을 수 있다. 이상적으로,  $h$  추정치의 시퀀스는 가장 좋은  $h$ 에 수렴될 것이다. 이 반복은 다음과 같이 표현된다.

$$h_0 = 0, \quad h_{k+1} = h_k + \frac{\sum_x \frac{w(x)[G(x) - F(x + h_k)]}{F(x + h_k)}}{\sum_x w(x)} \quad (3.73)$$

2차원 선형 근사치가 다른 형태로 발생하기 때문에, 앞서 설명한 도함수는 2차원으로 일반화가 잘되지 않는다. 또한, 수식 (3.68)은 곡선이 수평인 곳 즉  $F'(x) = 0$ 인 곳에서는 정의되지 않는다. 이 두 문제점은 수식 (3.74)의 선형 근사를 사용하여 해결할 수 있다. 이는 수식 (3.75)의 곡선 사이의 차이에 대한  $L_2$  표준 측정을 최소화하는  $h$ 를 찾는 것이다.

$$F(x + h) \approx F(x) + hF'(x) \quad (3.74)$$

$$N = \sum_x [F(x + h) - G(x)]^2 \quad (3.75)$$

$h$ 에 대한 오차를 최소화하기 위해서 다음과 같이 설정하고,  $h$ 의 추정치는 다음과 같다.

$$0 = \frac{\partial N}{\partial h} \approx \frac{\partial}{\partial h} \sum_x [F(x) + hF'(x) - G(x)]^2 \sum_x 2F'(x) [F(x) + hF'(x) - G(x)] \quad (3.76)$$

$$h = \frac{\sum_x F'(x) [G(x) - F(x)]}{\sum_x F'(x)^2} \quad (3.77)$$

따라서,  $h$ 의 추정치가 2차원 이상으로 일반화되었고,  $F'(x) = 0$ 에서 정의되지 않는 문제점이 해결되었다. 수식 (3.73)에 해당하는 가중치를 갖는 반복은 다음과 같이 나타낼 수 있다.

$$h_0 = 0, \quad h_{k+1} = h_k + \frac{\sum_x w(x)F'(x + h_k) [G(x) - F(x + h_k)]}{\sum_x w(x)F'(x + h_k)^2} \quad (3.78)$$

이제 이를 스테레오 카메라에 적용하는 방법에 대해 설명한다. 그림 3.4를 보면  $c$ 는 카메라 1의 좌표계에 대한 카메라 2의 방향과 위치를 설명하는 카메라 파라미터 벡터이다. 이러한 파라미터는 방위각, 고도, 팬(pan), 기울기, 그리고 roll이 있다[37].  $x$ 는 객체의 카메라 1 평면에 있는 이미지의 위치를 나타낸다. 객체가 카메라 1에서  $z$ 만큼 떨어져 있다고 가정한다. 카메라 1의 좌표계에 대해  $p$ 를 표현하여  $p$ 가 카메라 1의 방향에 의존하지 않도록 한다. 객체는 카메라 2의 평면에  $p$ 에서 객체의 위치와 카메라 파라미터에 따라 달라지는 위치  $q(p, c)$ 로 나타난다.  $G$ 와  $F$ 는 각각 픽셀 강도  $x$ 와  $q$ 를 갖는다[36].

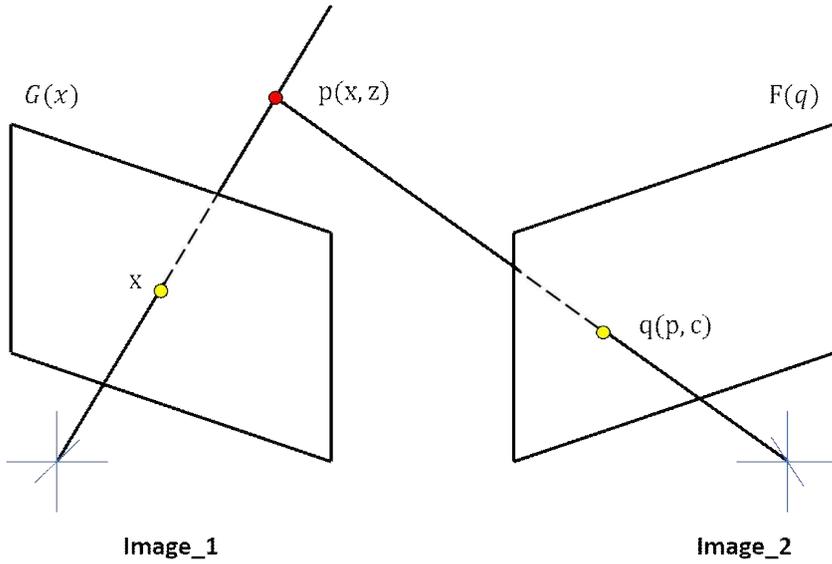


그림 3.4 스테레오 카메라 비전

먼저, 정확한 카메라 파라미터  $c$ 를 알고 있고, 객체까지의 거리를  $z$ 라고 가정한다. 거리  $z$ 에 대한 추정치를  $\Delta z$ 만큼 변화시키면 선형 근사치는 다음과 같다.

$$F(z + \Delta z) \approx F(z) + \Delta z \frac{\partial F}{\partial z} \quad (3.79)$$

$$\frac{\partial F}{\partial z} = \frac{\partial p}{\partial z} \frac{\partial q}{\partial p} \frac{\partial F}{\partial q} \quad (3.80)$$

여기서,  $\frac{\partial q}{\partial p}$ 는  $p$ 의 구성 요소에 대한  $q$ 의 부분 미분 행렬이고,  $\frac{\partial F}{\partial q}$ 는 이미지  $F(q)$ 의 공간 강도 기울기(spatial intensity gradient)이다. 따라서, 수식 (3.81)을 만족하는  $\Delta z$ 는 수식 (3.82)와 같다.

$$0 = \frac{\partial}{\partial \Delta z} N \approx \frac{\partial}{\partial \Delta z} \sum_x \left[ F + \Delta z \frac{\partial F}{\partial z} - G \right]^2 \quad (4.81)$$

$$\Delta z = \frac{\sum_x \frac{\partial F}{\partial z} [G - F]}{\sum_x \left( \frac{\partial F}{\partial z} \right)^2} \quad (3.82)$$

이번에는, 카메라 1에서 각  $n$ 개의 객체의 거리  $z_i (i = 1, 2, \dots, n)$ 은 알고 파라미터  $c$ 를 모른다고 가정한다. 이때 선형 근사 수식 (3.83)을 사용하여 수식 (3.84)와 같이 설정해주면  $\Delta c$ 에 대한 오류함수의 최소화를 할 수 있다.

$$F(c + \Delta c) \approx F(c) + \Delta c \frac{\partial q}{\partial c} \frac{\partial F}{\partial q} \quad (3.83)$$

$$0 = \frac{\partial}{\partial \Delta c} \sum_i \sum_x [F(c + \Delta c) - G]^2 \approx \frac{\partial}{\partial \Delta c} \sum_i \sum_x \left[ F + \Delta c \frac{\partial q}{\partial c} \frac{\partial F}{\partial q} - G \right]^2 \quad (3.84)$$

$$\Delta c \approx \left[ \sum_x \left( \frac{\partial q}{\partial c} \frac{\partial F}{\partial q} \right)^T [G - F] \right] \left[ \sum_x \left( \frac{\partial q}{\partial c} \frac{\partial F}{\partial q} \right)^T \left( \frac{\partial q}{\partial c} \frac{\partial F}{\partial q} \right) \right]^{-1} \quad (3.85)$$

### 3. 특이치 제거를 위한 RANSAC 알고리즘

특이치 제거를 위한 RANSAC 알고리즘은 B. Kitt[38]의 논문을 참고하였다. 독립적으로 이동하는 객체에 위치한 나머지 특징점은 RANSAC(Random Sample Consensus) 기반 특이치 제거를 사용한다. 랜덤으로 특징 대응의 부분 집합을 선택하고 이 부분 집합을 기반으로 자체 움직임(egomotion)을 추정한다. 사용된 부분 집합의 수는 다음과 같다[38].

$$n = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)} \quad (3.86)$$

여기서,  $p$ 는 하나 이상의 샘플에 특이치가 단독으로 포함될 확률,  $s$ 는 추정에 필요한 최소 데이터 점의 개수를 의미하고,  $\epsilon$ 은 데이터 집합에서 특이치의 가정된 백분율을 나타낸다[39].

## 제4장 지도 작성

### 제1절 지도의 종류

지도 작성은 로봇이 센서를 사용하여 주변 환경을 인식하고 감지하여 환경 지도를 만드는 것이다. 지도 작성에는 그리드 지도, 특징 기반 지도, 토폴로지컬 지도, 그리고 메트릭 지도 등 여러 가지 방법이 있다. 먼저, 그림 4.1과 같은 그리드 지도는 결과 지도를 그리드의 셀 집합으로 처리한다. 폐쇄된 위치에서 특징을 추출하는 방법이 불분명하기 때문에 주로 폐쇄된 환경에서 사용된다. 또한, 그리드 기반 지도는 일반적으로 2D에 적합하고, 스캔 매칭 방법을 사용하여 포즈를 바로잡는다. 반면 그림 5.2와 같은 특징 기반 지도는 지도를 특징 집합으로 처리한다. 특징은 장애물에 접해있는 것으로 간주되고, 이러한 종류의 지도는 일반적으로 폐쇄된 환경에서 특징을 추출하는 것이 까다롭기 때문에 넓은 실외 환경에 적합하다[40].

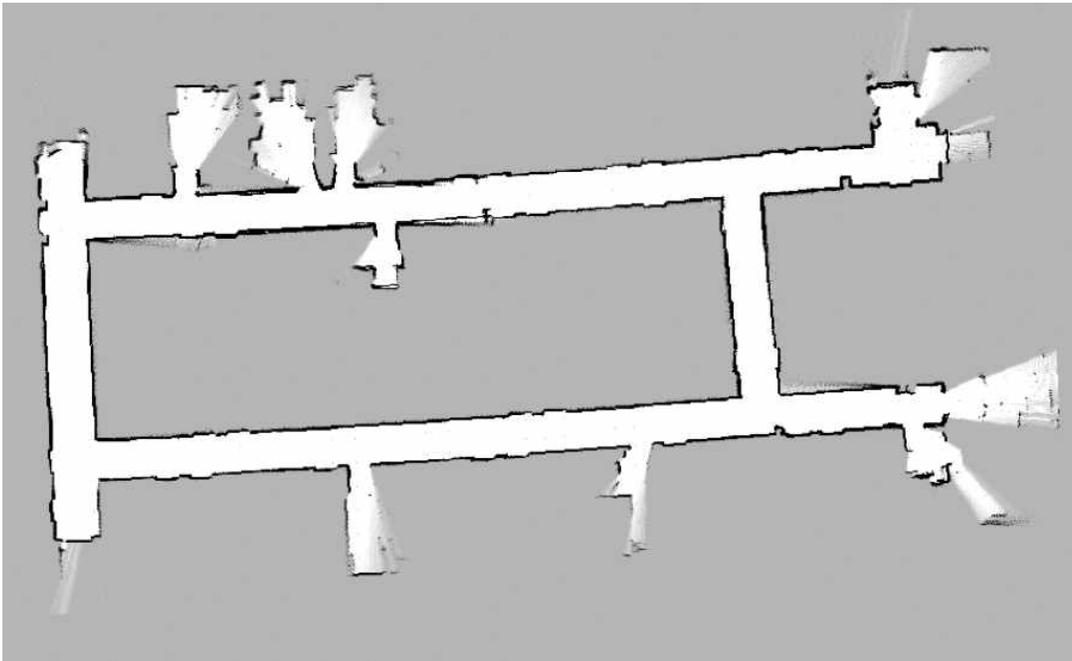


그림 4.1 그리드 지도 예시[40]

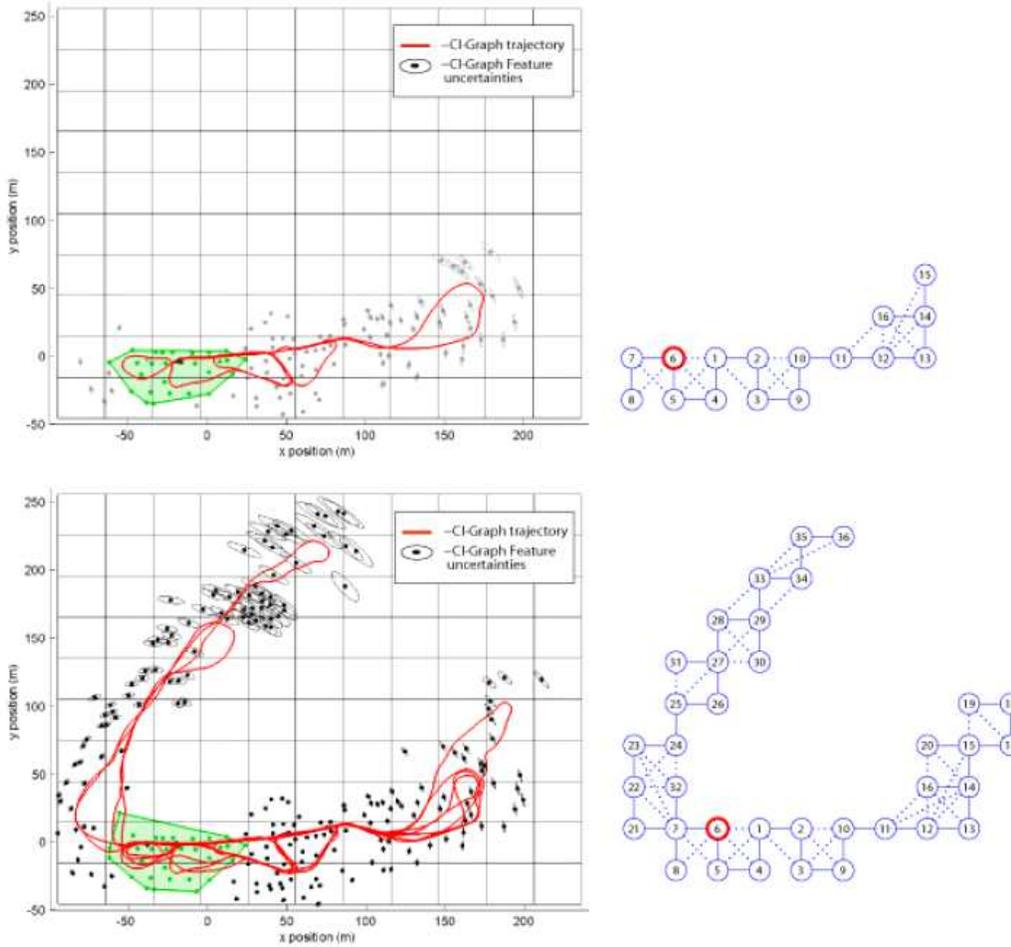


그림 4.2 특징 기반 지도 예시[40]

다음은, 토폴로지컬 지도에 대해 설명한다. 그림 4.3과 같은 토폴로지컬 지도는 그리드 지도에 비해 추성적인 방법으로, 어떠한 절대적 기준 없이 주변환경의 특징 사이의 관계만을 표현한다. 따라서, 일반적으로 토폴로지컬 지도는 노드와 원호(arc)로 구성된 그래프 통해 지도를 얻을 수 있다[41]. 여기서 노드는 환경의 고유한 위치를 나타내고, 원호는 이들 사이의 관계를 모델링한다. 이러한 지도는 단순하고 작으며, 저장 공간이 적게 필요하다. 메트릭(metric) 지도는 거리, 측정값 또는 크기와 같은 환경 세부 정보에 대한 많은 정보를 유지하며 가능한 정확한 환경을 나타내며, 일반적으로 글로벌 좌표계에 따라 만들어진다. 이 방법은 장애물 회피뿐만 아니라 이동체의 위치 파악 및 안내에 적합하다[42].

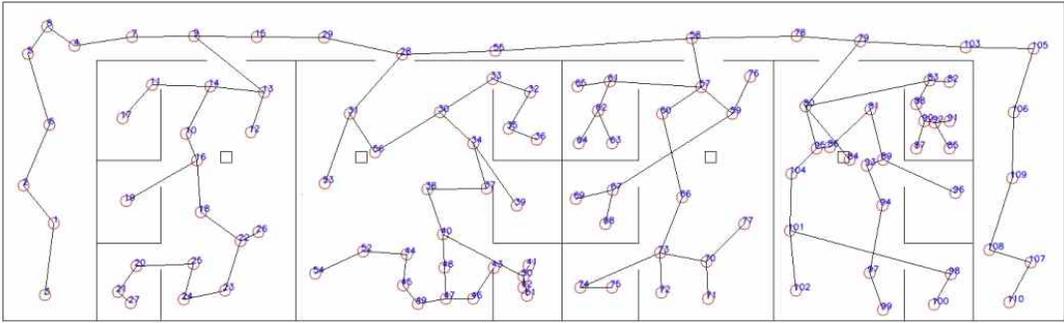


그림 4.3 토폴로지컬 지도 예시[43]

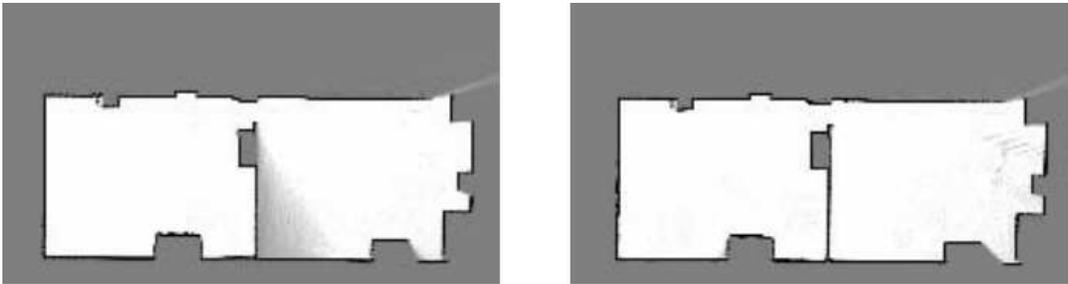


그림 4.4 메트릭 지도 예시[44]

## 제2절 RTAB-Map을 사용한 지도 작성

RTAB-Map(Real-Time Appearance-Based Mapping)은 2013년부터 오픈 소스 라이선스로 배포된 실시간 지도 작성을 할 수 있는 툴이다. 이는 대규모 및 장기적인 움직임을 처리하기 위해 그림 4.5와 같은 메모리 관리를 갖춘 외관 기반 루프 클로저(loop closure) 감지 방법으로 시작되었다[45]. 지각(perception) 모듈은 이미지를 얻어 SM(sensory memory)으로 전송하고, SM은 이미지 특징을 평가하여 데이터 크기를 줄이고 루프 클로저 탐지에 유용한 특징을 추출한다. 그런 다음, SM은 이미지 특징을 사용하여 새로운 location을 만들고 이를 STM(Short-Term Memory)에 전송한다. STM은 가중치를 업데이트하는 프로세스를 통해 최근에 생성된 location을 업데이트한다. 가중치 업데이트는 새로운 location이 STM의 마지막 위치와 유사하다고 판단되면 새로운 location에 해당 location을 병합하고 새로운 location의 가중치를 증가시킨다[46].

RTAB-Map은 시각적 및 LiDAR SLAM을 모두 지원하며, 사용자가 다양한 2D 및 3D 솔루션을 구현하고 로봇과 센서로 비교할 수 있는 틀을 하나의 패키지에 제공한다. RGB 이미지와 함께 깊이(depth) 이미지를 사용하여 지도를 구성하고, 그래프가 생성되며, 각 노드에는 해당 odometry 포즈가 있는 RGB 및 깊이 이미지가 포함된다. 또한, 각 노드 간의 변환을 나타내는 링크가 존재한다. 그래프가 업데이트 되면 RTAB-Map은 새로운 이미지를 그래프의 모든 이전 이미지와 비교하여 루프 클로저를 찾는다. 루프 클로저가 발견되면 그래프 최적화가 이루어지고 그래프의 포즈가 수정된다. 그래프가 각 노드에 대해 RGB 및 깊이 이미지에서 포인트 클라우드를 생성한다. 이 포인트 클라우드는 노드에서 포즈를 사용하여 변환한다. 그런 다음 3D 지도가 생성된다[47].

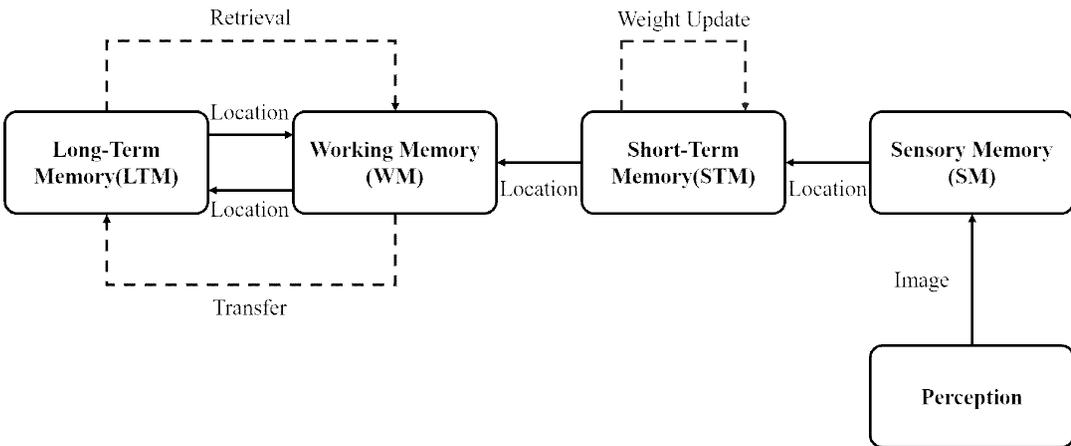


그림 4.5 RTAB-Map의 메모리 관리 모델[46]

## 제5장 실험 및 결과분석

### 제1절 센서 보정

본 연구에서 사용한 센서는 위드로봇의 oCamS-1CGN-U 모델 카메라이다. 이 모델은 스테레오 카메라이며 내장 IMU를 포함하고 있다. 카메라 사양은 표 5-1과 같고, 외관은 그림 6.1과 같다. 카메라의 좌표계는 그림 5.1에서와 같이 각 카메라의 좌표계는 일치하고, 오른쪽이 x축, 아래쪽이 y축, 그리고 앞쪽이 z축 방향이다. IMU의 좌표계는 앞쪽이 x축, 왼쪽이 y축, 그리고 위쪽이 z축 방향이다.

표 5-1. oCamS-1CGN-U 모델 카메라 사양

| 항 목                      | 내 용  |
|--------------------------|--|
| 영상 출력                    | - YUV422 (스테레오 영상)<br>- BayerRGB (분리된 영상)  |
| 지원 해상도(FPS)              | - 1280(*2) * 960 (45fps)<br>- 1280(*2) * 720 (60fps)<br>- 640(*2) * 480 (45fps)<br>- 640(*2) * 360 (60fps)<br>- 320(*2) *240 (60fps) |
| 셔터                       | - Global Shutter   |
| 컨트롤 가능한 파라미터             | - Gain<br>- Exposure (Absolute)<br>- White Balance RED<br>- White Balance BLUE   |
| Auto Exposure            | - 지원   |
| Auto White Balance       | - 지원하지 않음  |
| External Trigger         | - 지원하지 않음  |
| Lens Correction          | - 지원하지 않음  |
| White Balance Auto Setup | - 지원하지 않음  |

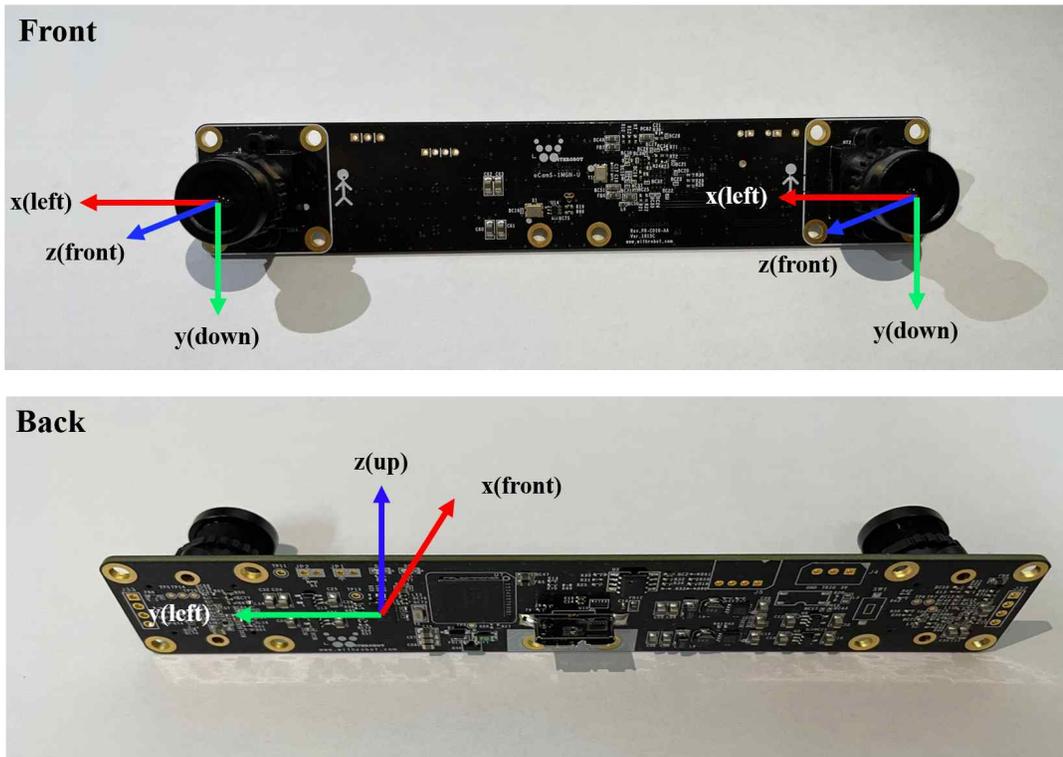


그림 5.1 oCamS-1CGN-U 외관

## 1. 스테레오 카메라의 보정 결과

보정을 하기 위해 Kalibr 툴을 사용하였고, 이는 멀티 카메라 보정, 카메라와 IMU 사이의 보정, 멀티 IMU 보정, 그리고 롤링 셔터 카메라 보정을 위해 사용하는 툴이다. 먼저, 스테레오 카메라와 IMU의 보정을 위해 두 개의 각 카메라 사이의 보정을 진행했다. 스테레오 카메라 보정을 위해 그림 5.2와 같은 April grid 패턴을 target으로 사용하였다. 사용된 April grid 패턴은 tag size 0.025m, space 0.006m로 tag spacing이 0.24인 6x6의 tag로 이루어져 있다.

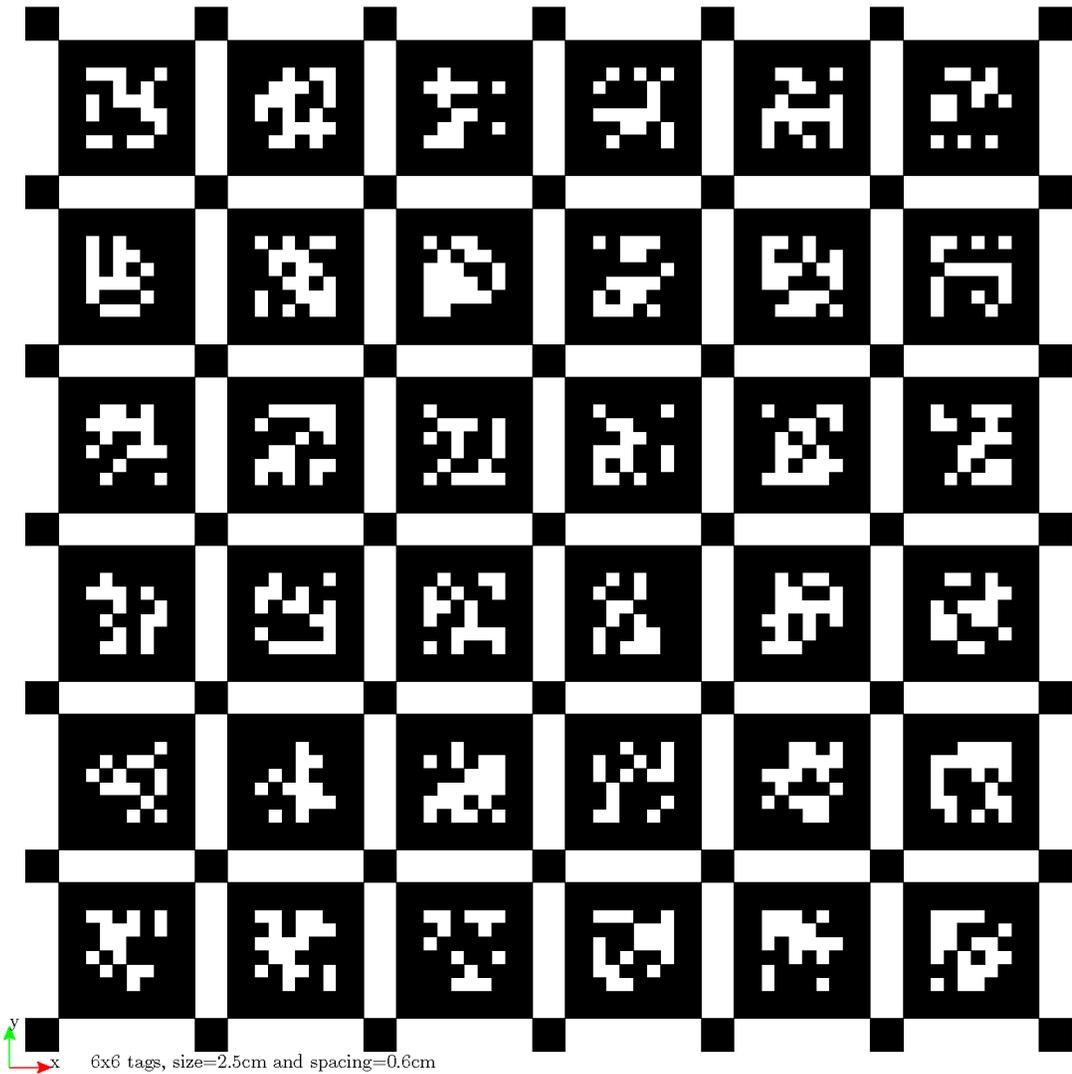
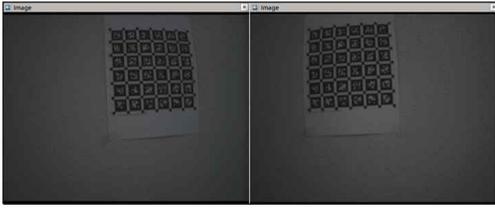
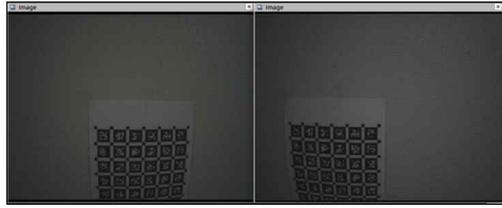


그림 5.2 스테레오 카메라 보정에 사용한 April grid 패턴

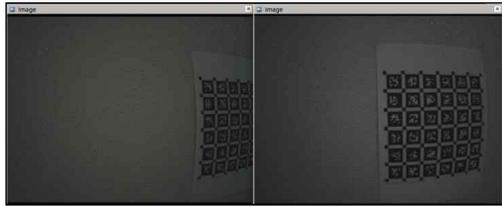
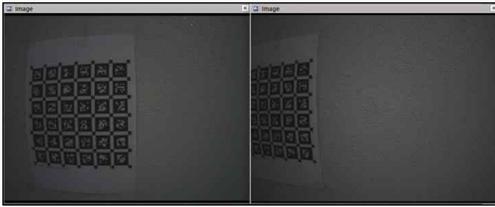
이 target을 스테레오 카메라의 두 카메라 모두 응시하도록 한 뒤, 상하좌우 및 앞뒤로 움직이면서 찍고, 왼쪽 오른쪽 각 카메라의 image raw 파일을 저장한다. 저장된 image raw 파일은 그림 5.3과 같다.



(a) 위아래로 움직인 카메라 image raw 데이터



(b) 좌우로 움직인 카메라 image raw 데이터

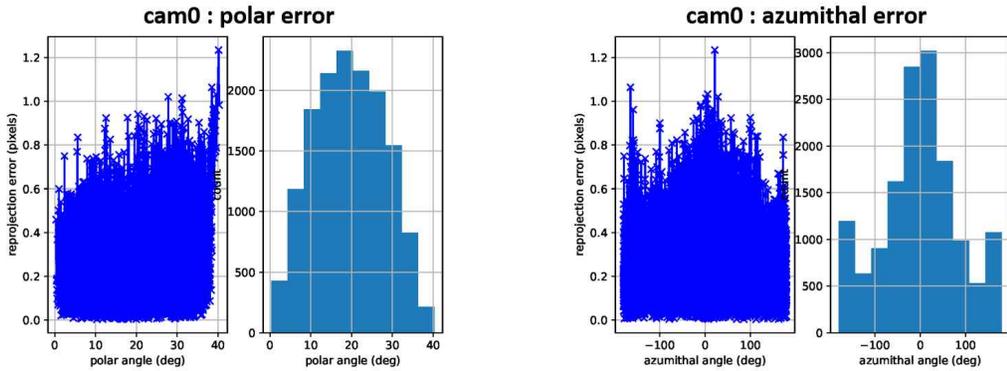


(c) 앞뒤로 움직인 카메라 image raw 데이터

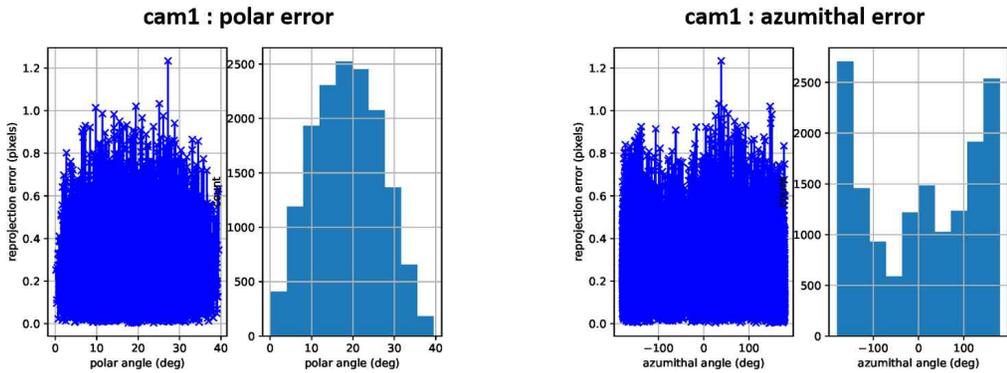
그림 5.3 bag 파일에 저장된 카메라 image raw 데이터

저장된 bag 파일과 bag 파일에 저장된 topic의 이름, 카메라 모델, 그리고 target.yaml 파일을 사용하여 스테레오 카메라의 보정을 진행한다. Target의 yaml 파일에는 target의 형식, 행 및 열의 개수, tag size 및 tagspacing에 대해 적혀 있다. 보정에 사용한 카메라의 투영 모델은 핀홀(pinhole) 카메라 모델이고, 왜곡 모델은 radial-tangential(radtan)이다.

스테레오 카메라 보정 후, 그림 5.4와 5.5와 같은 각 카메라의 polar error, azimuthal error, 그리고 재투영 오차(reprojection error)를 확인할 수 있다. polar error와 azimuthal error는 각각 polar angle과 azimuthal angle에서 발생하는 재투영 오차를 말한다. 재투영 오차란 투영된 영상 포인트와 관찰된 영상 포인트 사이의 거리이다.



(a) 왼쪽 카메라에 대한 polar error와 azimuthal error



(b) 오른쪽 카메라에 대한 polar error와 azimuthal error  
그림 5.4 스테레오 카메라의 polar error와 azimuthal error

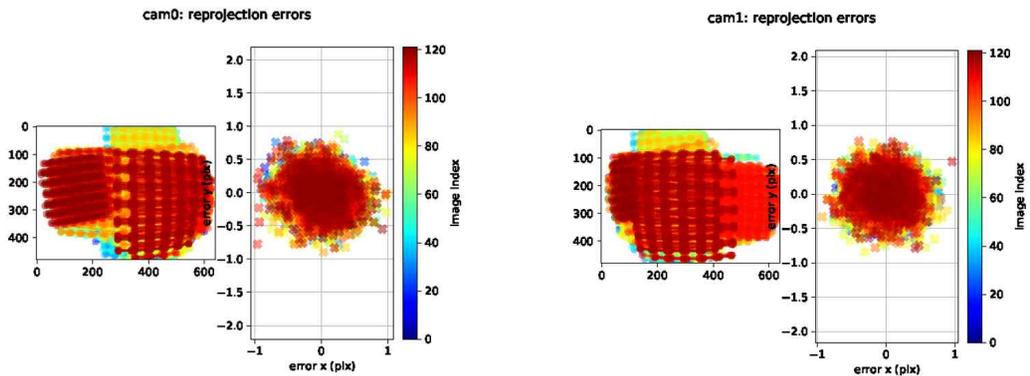


그림 5.5 스테레오 카메라의 재투영 오차

스테레오 카메라의 보정 결과 구해진 내부 파라미터는 다음 표와 같다. 이 파라미터는 ‘camchain.yaml’ 파일에 저장된다.

표 5-2. 보정 결과 구해진 카메라의 내부 및 외부 파라미터

| 카메라  | 내부 및 외부 파라미터            | 값  |
|------|-------------------------|--|
| cam0 | distortion coefficients | $[-0.4206, 0.1731, -1.5014 \times 10^{-5}, -0.0017]$   |
|      | intrinsics vector       | $[483.45, 484.82, 315.56, 247.66]$   |
| cam1 | transformation matrix   | $\begin{bmatrix} 0.9999 & -0.0004 & -0.0152 & -0.1202 \\ 0.0006 & 0.9999 & 0.0098 & -1.9093 \\ 0.0152 & -0.0098 & 0.9998 & -0.0014 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
|      | distortion coefficients | $[-0.4205, 0.1725, -0.0005, -0.0019]$  |
|      | intrinsics vector       | $[485.32, 486.22, 326.98, 235.87]$   |

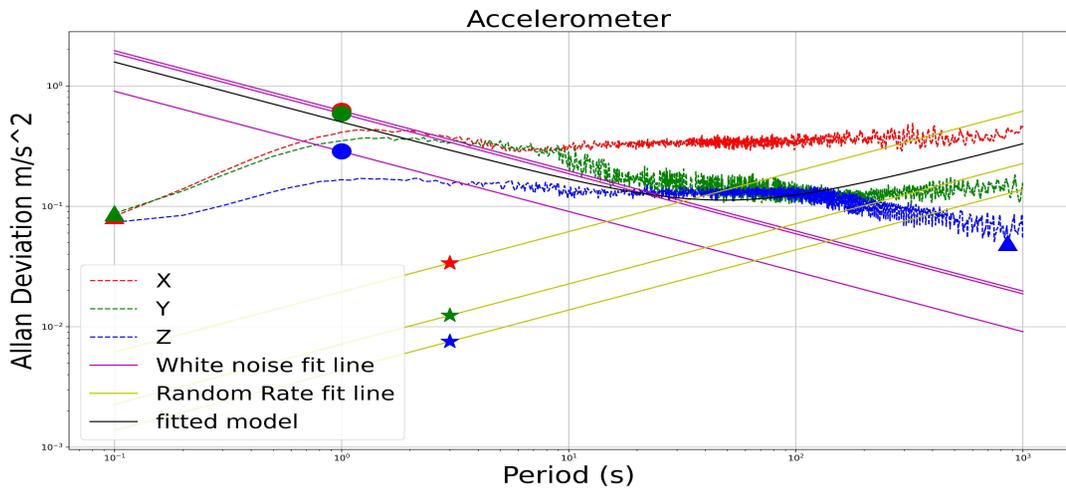
Transformation matrix에서 1행에서 3행, 1열에서 3열까지는 3x3의 회전 행렬을 나타내고, 4열의 1행에서 3행까지는 3x1의 두 카메라 사이의 위치 차이를 나타낸다. 실제 두 카메라 사이의 거리는 0.12[m]로 transformation matrix와 거의 일치하고, 두 카메라의 좌표계가 같기 때문에 회전 행렬의 대각 성분이 거의 1에 가까워야 한다. 따라서 두 카메라 사이의 보정이 잘 이루어진 것을 확인할 수 있다.

## 2. 카메라와 IMU 사이의 보정 결과

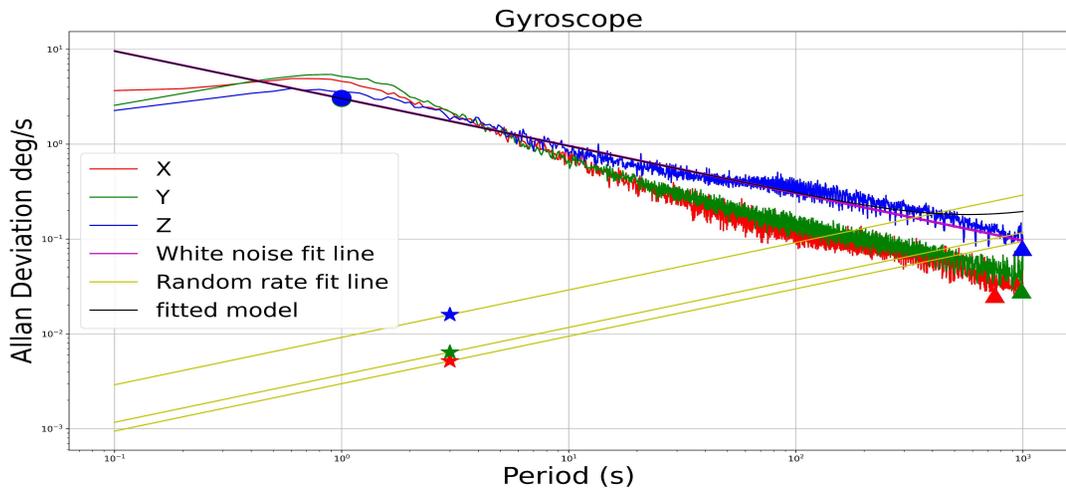
스테레오 카메라의 보정과 마찬가지로, 카메라와 IMU 사이의 보정을 위해 Kalibr 툴을 사용하였다. 보정에 앞서, 보정을 하기 위해 필요한 IMU noise 파라미터를 구해야 한다. IMU noise 파라미터는 Allan variance를 통해 얻었다. Allan variance를 진행하기 위해서 IMU의 데이터가 저장된 bag 파일을 생성해야 하고, IMU의 topic, rate, measure rate, bag 파일의 저장 시간에 대한 yaml 파일이 필요하다. 데이터는 IMU를 x축, y축, 그리고 z축 방향으로 움직이면서 적어도 3시간



그림 5.7은 각각 가속도계와 자이로스코프에 대한 Allan deviation을 시각화한 그래프이다. 이를 사용하여 Allan variance를 계산한다.



(a) 가속도계에 대한 Allan Deviation



(b) 자이로스코프에 대한 Allan Deviation

그림 5.7 가속도계와 자이로스코프에 대한 Allan Deviation 그래프

Allan deviation을 사용하여 Allan variance를 계산한 결과 가속도계에 대한 noise density와 random walk 및 자이로스코프에 대한 noise density와 random

walk가 구해진다. 값은 다음 표 5-4와 같다.

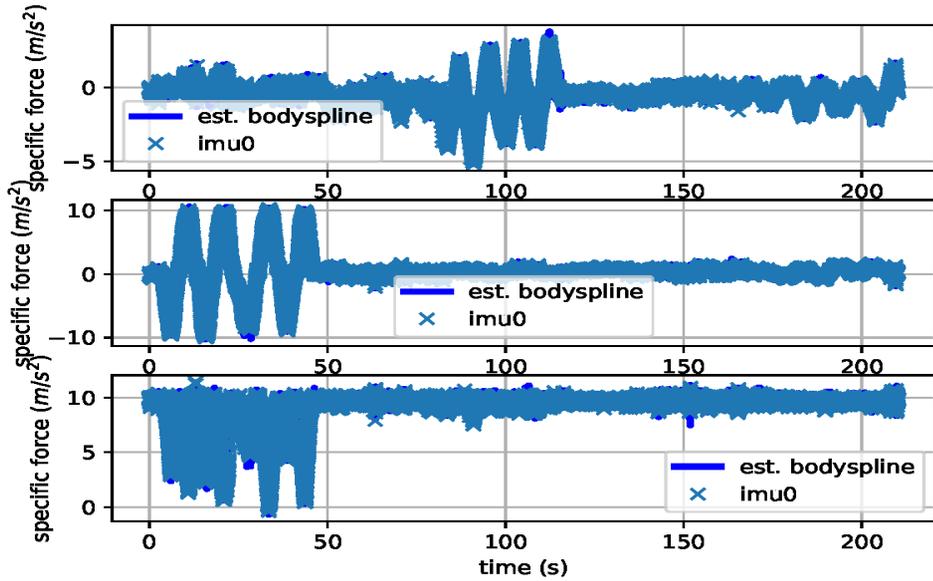
표 5-4. Allan variance 결과

| 파라미터   |               | 값  |
|--------|---------------|--|
| 가속도계   | noise density | $0.4983551923457137 \left[ \frac{m}{s^2} \frac{1}{\sqrt{Hz}} \right]$      |
|        | random walk   | $0.17887564416021323 \left[ \frac{m}{s^2} \frac{1}{\sqrt{Hz}} \right]$     |
| 자이로스코프 | noise density | $0.0527859239490238 \left[ \frac{rad}{s} \frac{1}{\sqrt{Hz}} \right]$      |
|        | random walk   | $0.2223121968942221488 \left[ \frac{rad}{s^2} \frac{1}{\sqrt{Hz}} \right]$ |

IMU의 noise 파라미터를 구하면, 카메라와 IMU 사이의 보정을 할 수 있게 된다. 카메라와 IMU 사이의 보정을 위해서는 카메라와 IMU의 데이터가 모두 담겨있는 bag 파일이 필요하다. 앞의 카메라 보정에서와 마찬가지로 target을 카메라가 바라보게 하고, 움직이면서 카메라와 IMU 데이터를 동시에 저장한다. 이때, 카메라의 topic은 20[Hz] 이하, IMU topic은 200[Hz] 이하의 속도로 저장해야 한다. Bag 파일, 앞서 얻은 스테레오 카메라 보정 결과, IMU noise 파라미터, 그리고 target의 yaml 파일을 사용하여 카메라와 IMU 사이의 보정을 진행한다.

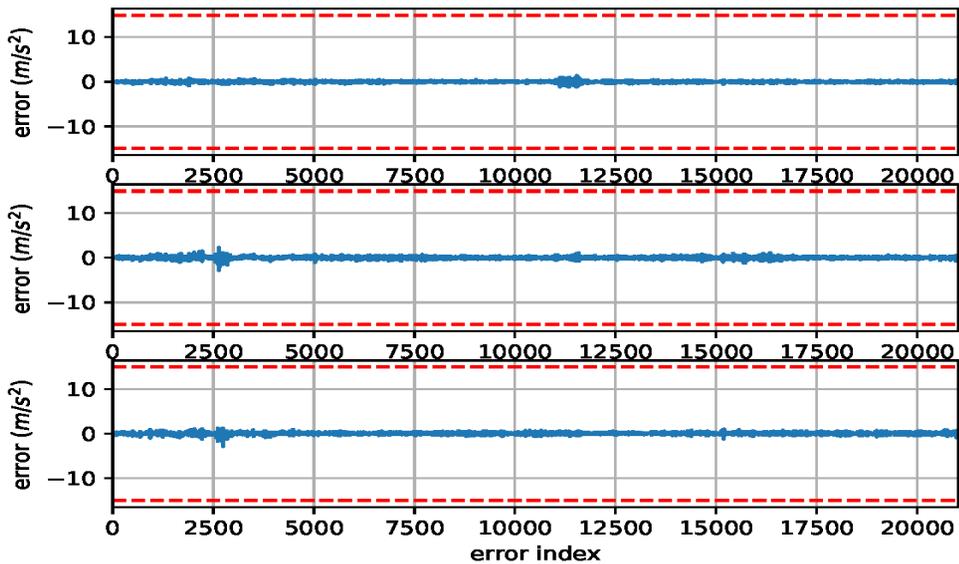
그림 5.8의 (a)는 IMU 프레임에서 예측된 가속도와 측정된 가속도를 비교한 그래프이다. 예측된 결과와 측정된 결과가 거의 일치하는 것을 볼 수 있다. (b)는 x, y, z축에 대한 IMU의 가속도계 오차를 나타낸다. 오차는 크게 나타나지 않는 것을 확인할 수 있다. 또한, 그림 5.9의 (a)는 body 프레임에서 예측된 각속도와 측정된 각속도를 비교한 그래프이다. y와 z축 방향으로의 각속도에서 예측된 각속도와 측정된 각속도의 차이가 보인다. (b)는 IMU의 각속도 오차를 나타낸다. x축에 비해 y와 z축 방향에서 상대적으로 큰 오차가 나타나는 것을 볼 수 있다. 그림 5.10은 카메라와 IMU 데이터가 저장된 bag 파일에서 두 카메라의 재투영 오차를 나타낸다. cam0이 cam1에 비해 상대적으로 오차가 큰 것을 볼 수 있고, y축 오차에 비해 x축 오차가 큰 것을 확인할 수 있다.

Comparison of predicted and measured specific force (imu0 frame)



(a) 예측된 가속도와 측정된 가속도 비교

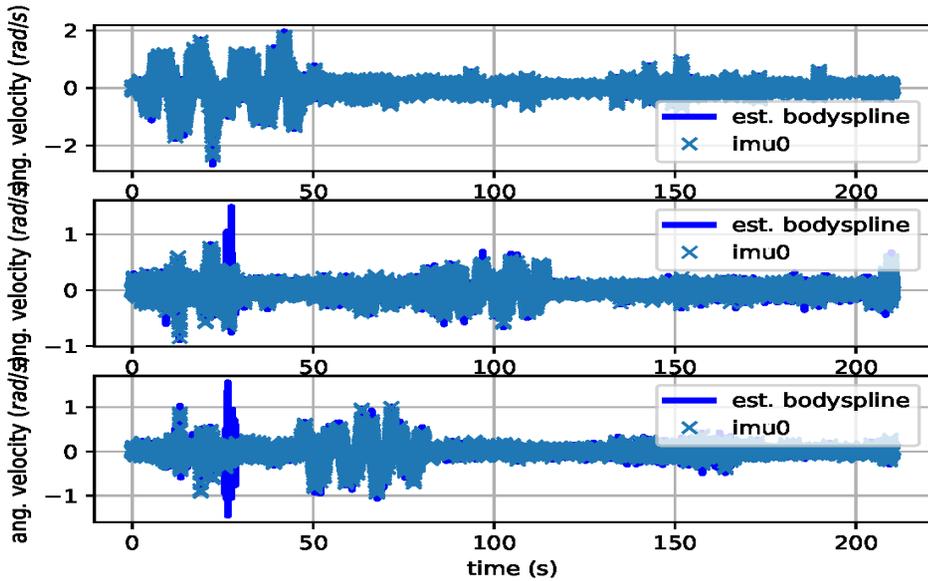
imu0: acceleration error



(b) 가속도계 오차

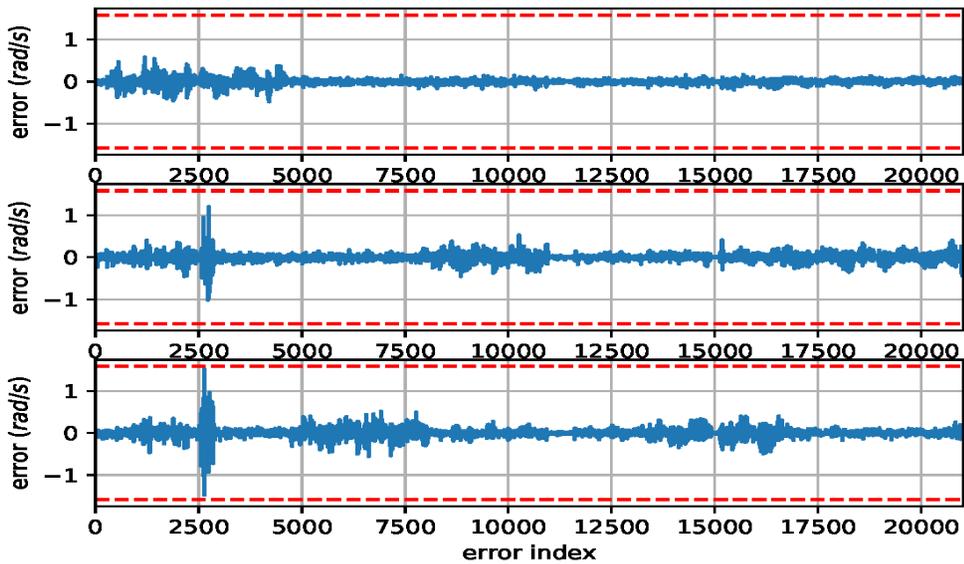
그림 5.8 가속도 비교 및 오차

Comparison of predicted and measured angular velocities (body frame)



(a) 예측된 각속도와 측정된 각속도 비교

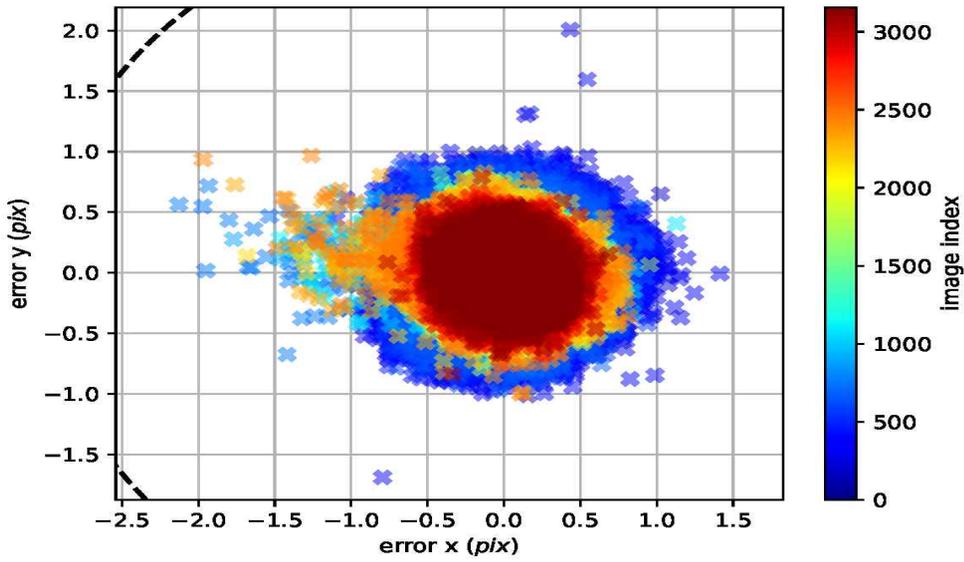
imu0: angular velocities error



(b) 각속도 오차

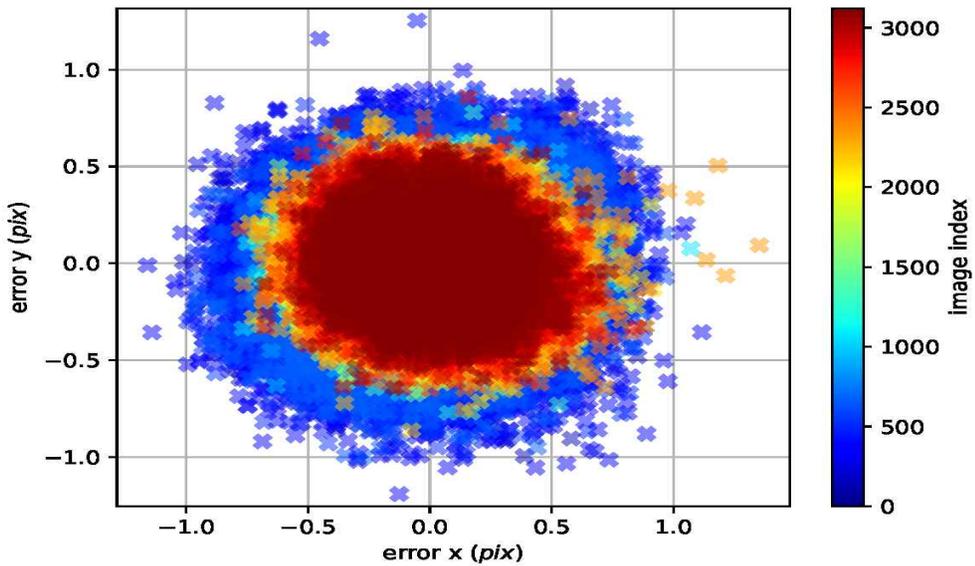
그림 5.9 각속도 비교 및 오차

cam0: reprojection errors



(a) cam0에 대한 재투영 오차

cam1: reprojection errors



(b) cam1에 대한 재투영 오차

그림 5.10 카메라에 대한 재투영 오차

최종 스테레오 카메라와 IMU 사이의 보정 결과는 표 5-5와 같다.

표 5-5. 카메라와 IMU 보정 결과

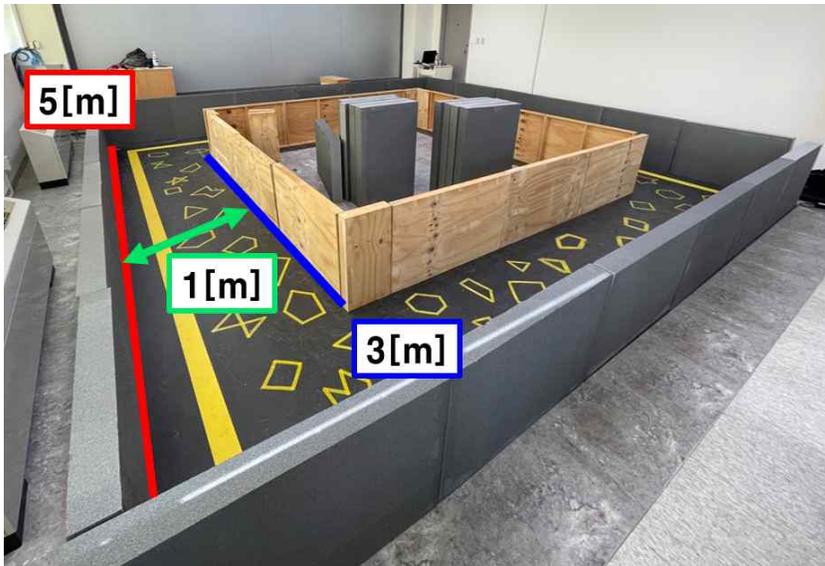
| 카메라  | Transformation matrix   |
|------|---|
| cam0 | $T_{cam0, imu} = \begin{bmatrix} 0.00576405 & -0.99995996 & 0.00684482 & 0.00366595 \\ 0.00365529 & -0.00682382 & -0.99997004 & 0.00014416 \\ 0.99997671 & 0.0057889 & 0.00361581 & -0.00040003 \\ 0 & 0 & 0 & 1 \end{bmatrix}$     |
| cam1 | $T_{cam1, imu} = \begin{bmatrix} -0.00947521 & -0.99992924 & 0.00719284 & -0.11653622 \\ 0.01349822 & -0.00732061 & -0.99988210 & 0.00012315 \\ 0.99986400 & -0.00937701 & 0.13566631 & -0.00181500 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
|      | $T_{cam1, cam0} = \begin{bmatrix} 0.99988382 & -0.00040392 & -0.01523747 & -0.12020778 \\ 0.005539 & 0.99995143 & 0.00984015 & -0.00001909 \\ 0.01523276 & -0.00984745 & 0.99983548 & -0.00146944 \\ 0 & 0 & 0 & 1 \end{bmatrix}$   |

여기서,  $T_{cam0, imu}$  는 IMU 프레임을 cam0 프레임으로 변환하는 변환 행렬,  $T_{cam1, imu}$  는 IMU 프레임을 cam1 프레임으로 변환하는 변환 행렬, 그리고  $T_{cam1, cam0}$  는 cam0 프레임을 cam1 프레임으로 변환하는 변환 행렬을 나타낸다.

## 제2절 시각 관성 주행 거리 측정 및 RTAB-Map 결과

시각 관성 주행 거리 측정과 3D 지도 작성을 위해 조선대학교 IT 융합대학 6103호 공유교육실습실에서 실험을 진행하였다. 실험실의 트랙은 그림 5.11의 (a)와 같이 안쪽은 3m×3m이고, 바깥쪽은 5m×5m, 그리고 폭 1m로 와 같고, 실험실에 배치되어있는 트랙에 내부 및 외부 벽을 설치하였다. 또한, (a), (b)와 같이 바닥과 천장에는 시각 관성 주행 거리 측정에 필요한 특징점을 위해 랜드마크를 부착하였다. 랜드마크의 형태는 꼭짓점을 갖는 도형을 무작위로 설치하였다. 실험실 트랙 바닥이 검은색에 빛 반사가 많이 되는 재질이기 때문에, 랜드마크는 상대적으로 빛 반사가 적고, 너무 밝지 않은 것으로 선택하였다. 카메라의 성능의 한계와 알고리즘을 실행시킨 노트북의 성능의 한계로 인해 최대한 실험 환경을 맞추는

쪽으로 실험을 진행하였다. 실험은 카메라가 바닥, 정면, 그리고 천장을 향한 경우의 세 가지 상황으로 나누었다.



(a) 실험 트랙 및 바닥에 부착된 랜드마크

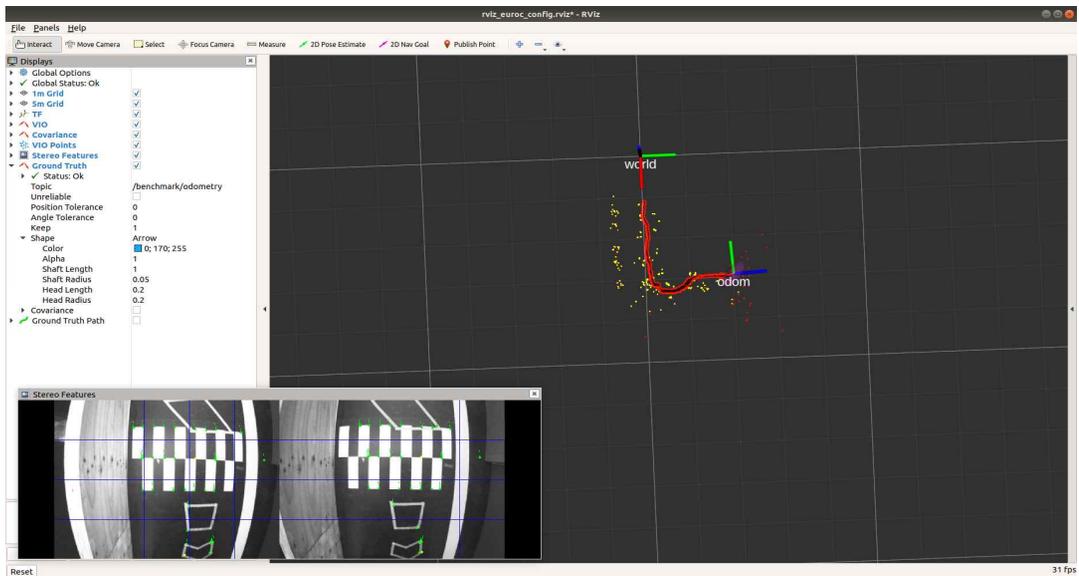


(b) 천장에 부착된 랜드마크

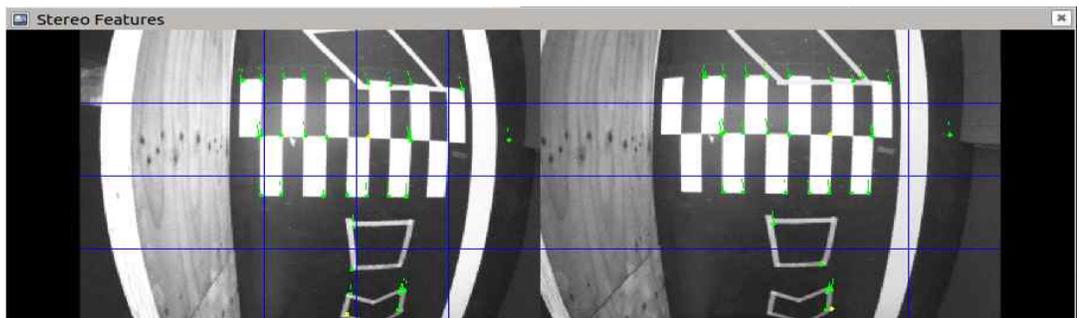
그림 5.11 실험 환경

## 1. 카메라가 바닥을 향한 경우

카메라를 바닥을 향하도록 하는 경우 시각 관성 주행 거리 측정 알고리즘을 실행시킨 후, ROS 환경에서의 시각화 툴인 rviz를 통해 추출된 특징점과 카메라의 추정된 위치를 시각화하였다. 그림 5.12는 카메라가 바닥을 향하도록 하고 실험실의 트랙을 2바퀴 도는 실험이다. 실험은 블라인드를 내리고 불을 켜 상태로 진행하였다.



(a) 시각 관성 주행 거리 측정 시각화

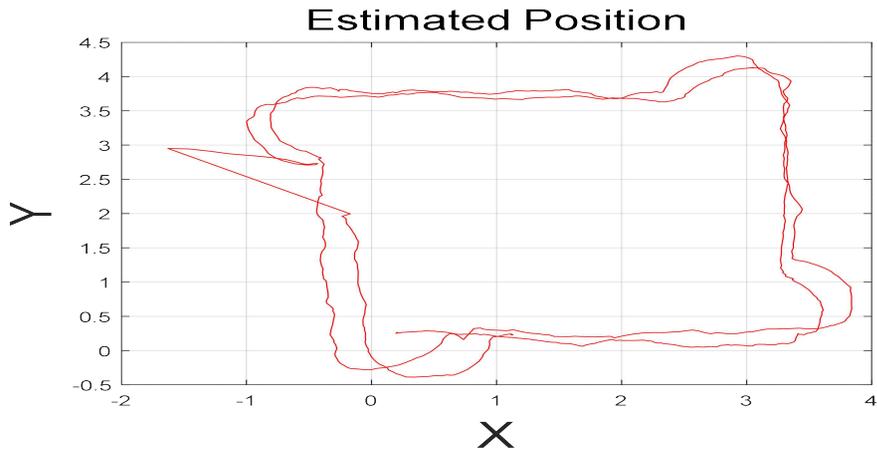


(b) 스테레오 카메라의 특징점 추출

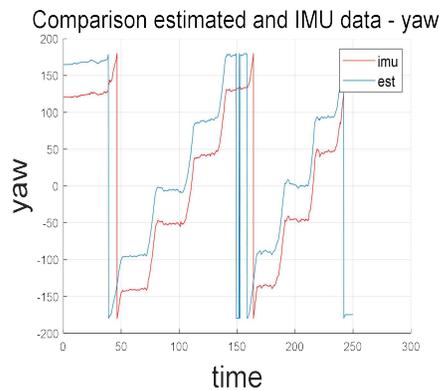
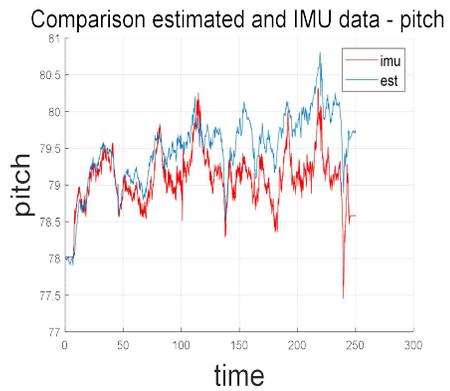
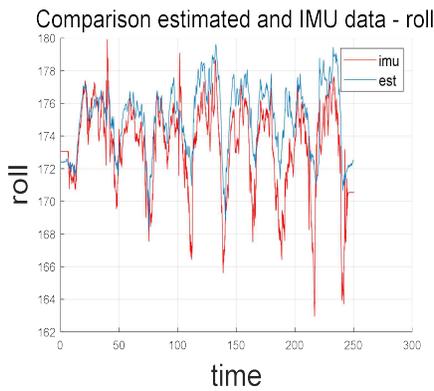
그림 5.12 바닥을 향한 경우 시각 관성 주행 거리 측정 시각화

여기서, (a)의 rviz 상의 점들은 (b)의 스테레오 카메라에서 추출된 특징점 중 두 카메라에서 동시에 인식된 점들을 나타낸다. ‘world’ 좌표계는 실험을 처음 시작한 위치와 자세를 나타내는데, 이때 카메라를 바닥을 향하게 하였기 때문에 자세는 pitch 방향으로 +90도 회전해야 초기 자세가 된다. ‘odom’ 좌표계는 카메라에 내장된 IMU의 위치 및 자세를 나타낸다. 시각 관성 주행 거리 측정을 통해 추정된 위치와 자세를 bag 파일로 저장하여 MATLAB을 통해 그래프를 그려 확인하였다.

트랙의 크기가 약 4mx4m에서 5mx5m이고, 추정된 위치로 확인한 경로는 약 4mx4m로 출력되었다. 이는 참값이 없고, 사람이 직접 카트를 밀어 실험을 진행하였기 때문에, 참값과 정확하게 비교할 수는 없지만, 범위 안에 들어오는 것을 알 수 있다. 중간에 튀는 현상은 갑작스럽게 특징점이 추출되지 않아 생기는 드리프트로 인해 발생한다. 또한, 자세의 경우 pitch가 90도 이기 때문에 추정값과 실제 IMU 센서값 사이의 오차가 약 50도 정도로 크게 발생하였다. 오차를 50도로 유지하지만, 자세의 패턴은 거의 추정값과 IMU 센서값이 거의 일치하는 것을 볼 수 있다. 오차의 원인에 대한 설명은 다음 장에서 자세하게 다룰 것이다.



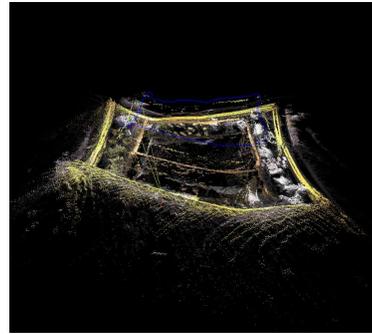
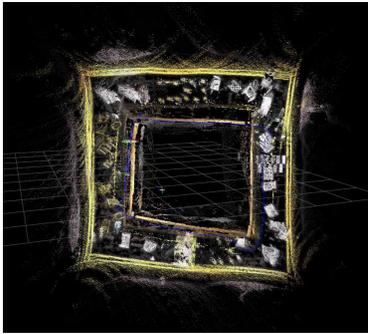
(a) 시각 관성 주행 거리 측정을 통해 추정된 위치 그래프



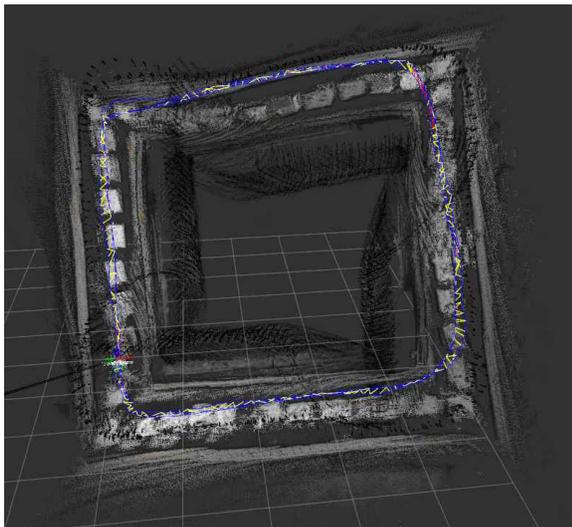
(b) 시각 관성 주행 거리 측정을 통해 추정된 자세 그래프

그림 5.13 바닥을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프

카메라를 바닥을 향하게 하고 RTAB-Map을 통해 3D 지도 작성을 진행한 결과는 그림 5.14와 같다. 카메라를 바닥을 향하게 한 경우에는 위의 그림과 같이 지도 작성 결과가 90도로 나오게 된다. 이는 그림 5.15와 같이 카메라의 입장에서는 자신이 바닥을 향한 것이 아닌, 바닥이 90도로 들려있다고 느끼기 때문이다. 그림 5.14의 (a)에서는 카메라를 장애물이 잘 보이도록 위쪽에 설치하여 흔들림이 더 심해져 지도가 휘어 나오는 것을 볼 수 있다. 흔들리지 않도록 아래쪽에 설치하여 실험을 진행한 경우에는 (b)와 같이 평평하게 나오게 된다.



(a) 흔들렸을 경우



(b) 흔들리지 않았을 경우

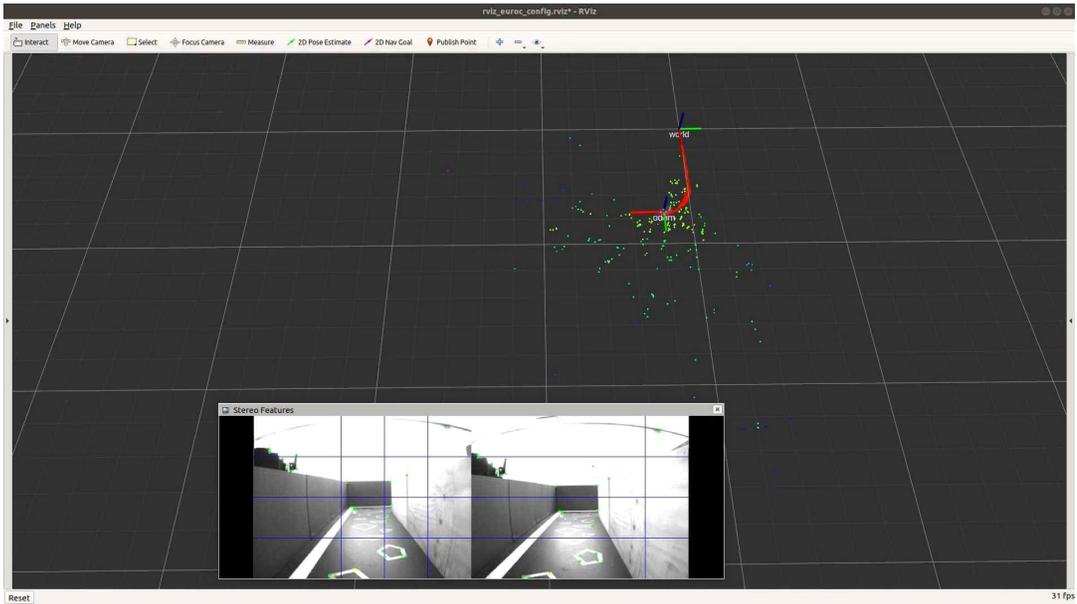
그림 5.14 바닥을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과



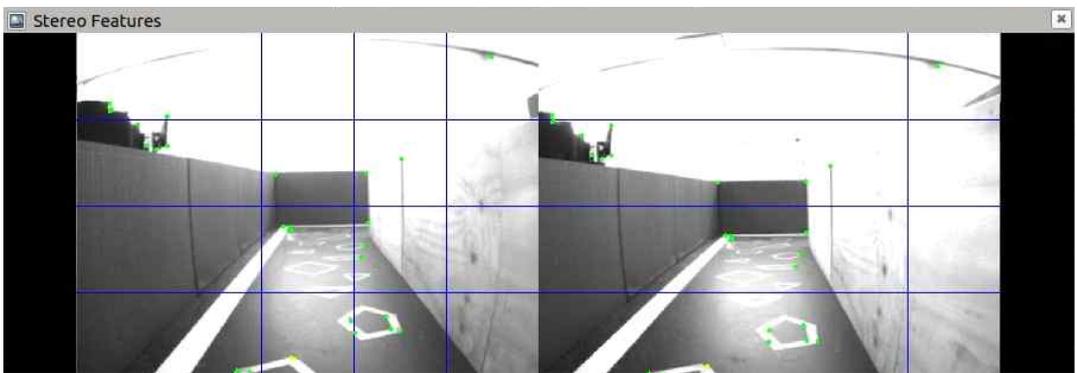
그림 5.15 바닥을 향했을 경우 카메라 화면

## 2. 카메라가 정면을 향한 경우

다음은, 카메라를 정면을 향하도록 고정하고 진행한 실험이다. 실험은 바닥을 향한 경우와 마찬가지로 블라인드를 내리고 조명을 켜 상태로 진행하였다. 그림 5.16은 카메라를 고정시키고 시각 관성 주행 거리 측정을 실행하여 rviz 상에서 시각화하여 추출된 특징점과 카메라의 추정된 위치를 보여준다. 마찬가지로, (a)의 점들은 (b)의 스테레오 카메라에서 추출된 특징점 중 두 카메라에서 모두 추출된 특징점이다. ‘world’ 좌표계는 실험을 처음 시작한 위치와 자세를 나타낸다. ‘odom’ 좌표계는 카메라에 내장된 IMU의 위치 및 자세를 나타낸다. 또한, 시각 관성 주행 거리 측정을 통해 추정된 위치와 자세를 bag 파일로 저장하여 MATLAB을 통해 그래프를 그려 확인하였다. 이는 그림 5.17에 나타나 있다.

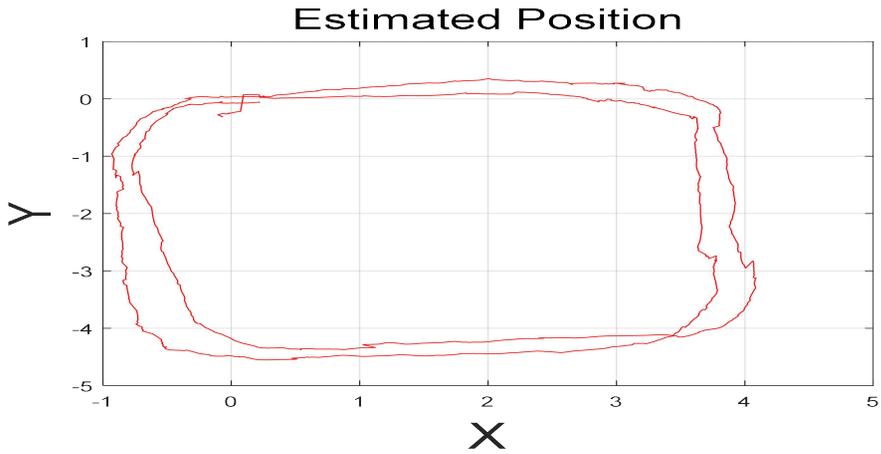


(a) 시각 관성 주행 거리 측정을 시각화

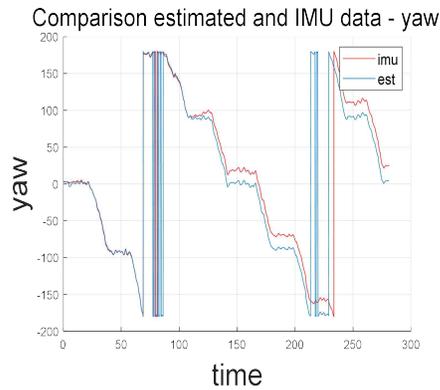
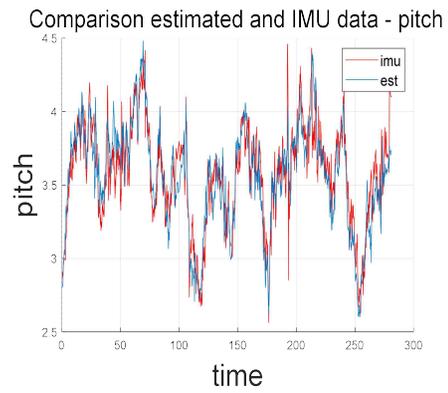
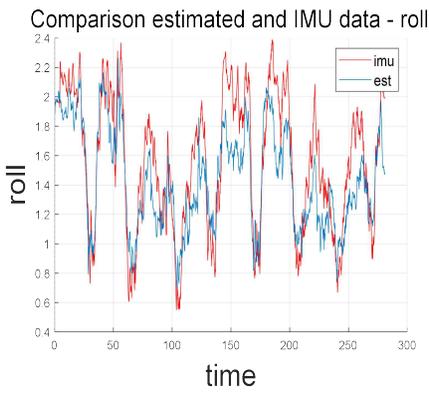


(b) 스테레오 카메라의 특징점 추출

그림 5.16 정면을 향한 경우 시각 관성 주행 거리 측정 시각화



(a) 시각 관성 주행 거리 측정을 통해 추정된 위치 그래프



(b) 시각 관성 주행 거리 측정을 통해 추정된 자세 그래프

그림 5.17 정면을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프

추정된 위치는 트랙의 크기 4m×4m에서 5m×5m와 비슷한 약4.3m×4.5로 나타나고, 자세는 한바퀴를 돌았을 때는 추정값과 IMU 센서값이 거의 일치했지만, 이후에는 약 20도의 오차를 유지한다. 이는 카메라의 성능 및 컴퓨터 성능과도 관련이 있다. 데이터를 저장하기 위해 bag 파일을 만들 때, 컴퓨터의 성능으로 인해 버벅임이 생기게 된다. 이때, 저장된 bag 파일로 실험을 진행하면 파일 자체가 조금씩 끊겨 저장되기 때문에 중간에 끊긴 값들로 인해 정확도가 떨어지게 된다. 또한, 외벽에 막혀 특징점이 추출되지 않는 경우 등에도 드리프트가 발생하여 정확도가 떨어질 수 있다. 그렇지만 자세의 패턴은 거의 추정값과 IMU 센서값이 거의 일치하는 것을 볼 수 있다.

그림 5.13과 5.17을 통해 카메라가 바닥을 향한 경우보다 정면을 향한 경우가 시각 관성 주행 거리 측정의 성능이 더 좋은 것을 볼 수 있다. 카메라가 앞을 향했을 경우, RTAB-Map을 실행한 결과는 그림 5.18과 같다. 3D 지도 작성 결과 루프 클로저가 잘 이루어진 것을 볼 수 있고, 카메라가 바닥을 향했을 경우와는 다르게 지도가 바닥에 평평하게 만들어진 것을 볼 수 있다.

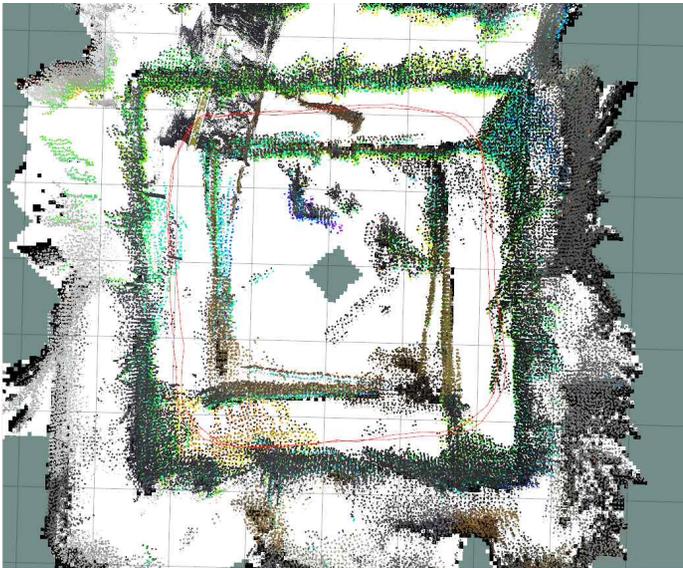
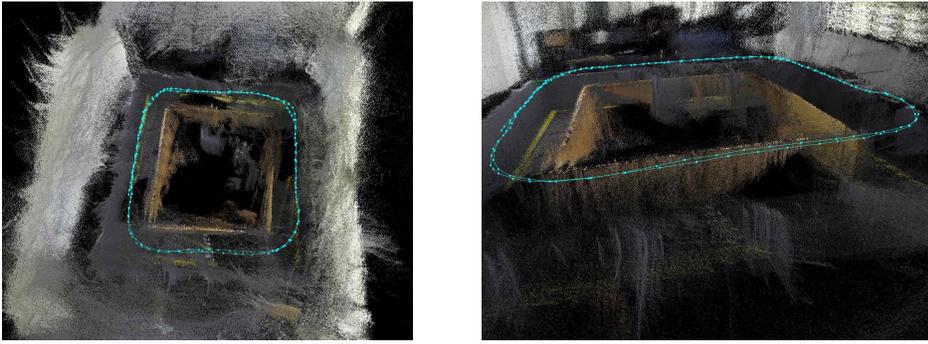
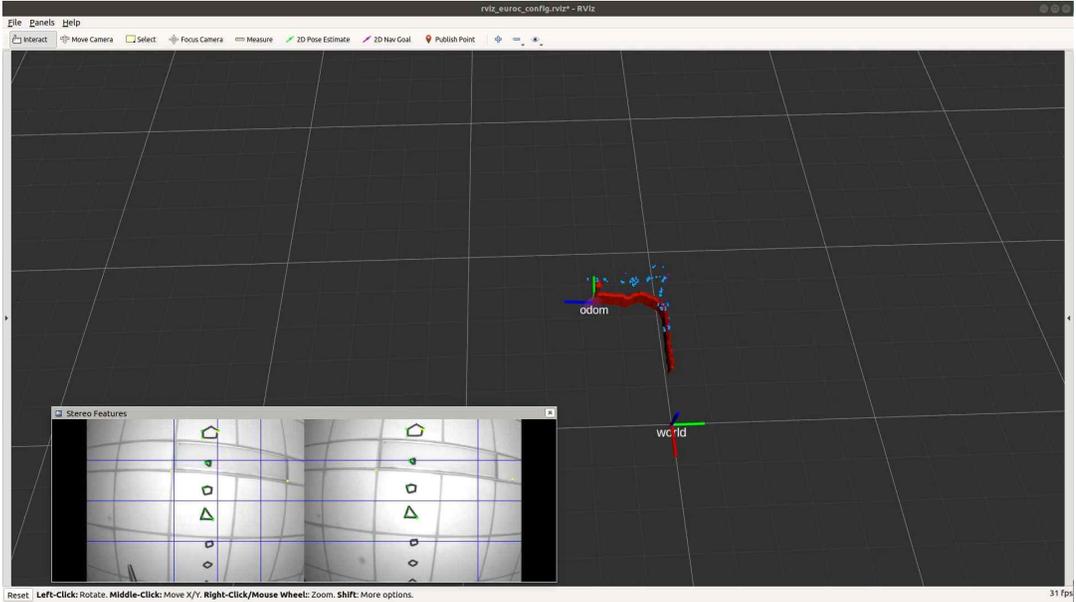


그림 5.18 정면을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과

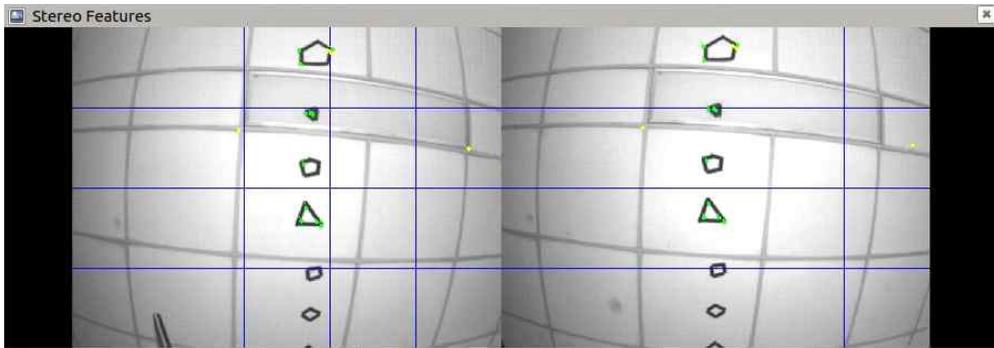
### 3. 카메라가 천장을 향한 경우

마지막으로, 카메라가 천장을 향한 경우이다. 실험은 블라인드를 내리고 조명을 끈 상태로 진행하였다. 이 경우도 바닥을 향한 경우와 비슷하게 pitch가  $-90^\circ$ 이다. 그림 5.19는 카메라를 천장을 향하게 하고 시각 관성 주행 거리 측정 알고리즘을 시각화한 것이다. 카메라를 천장을 향하게 하는 경우는 다른 경우에 비해 빛이 카메라 렌즈를 통해 많이 들어가기 때문에 환경 조건이 까다롭다. 너무 밝으

면 렌즈를 통해 아무것도 보이지 않아, 특징점을 찾지 못하게 된다. 따라서, 빛 조절이 천장 실험의 가장 중요한 환경 조건이다. 그림 5.20은 카메라가 천장을 향했을 경우 추정된 위치와 자세를 MATLAB을 통해 그래프로 나타낸 것이다.

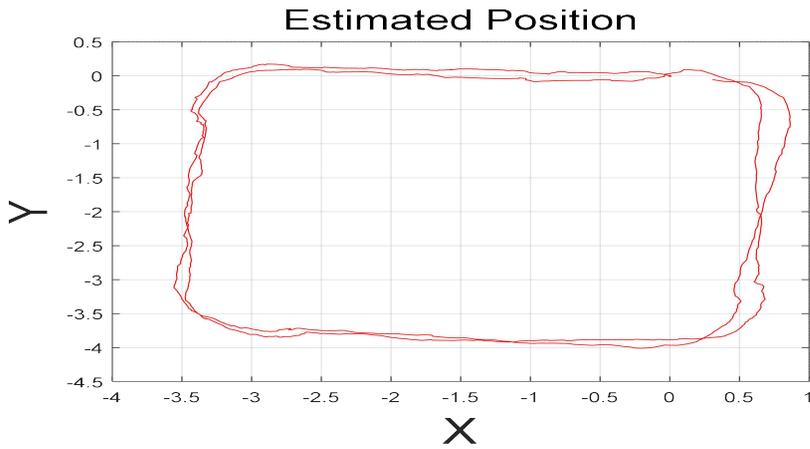


(a) 시각 관성 주행 거리 측정 시각화

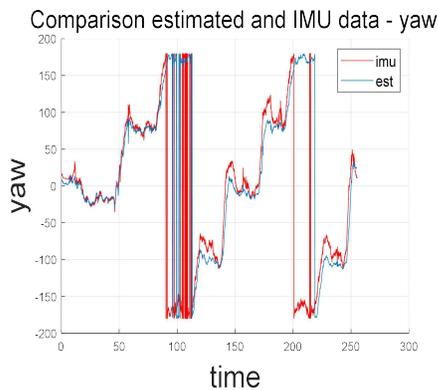
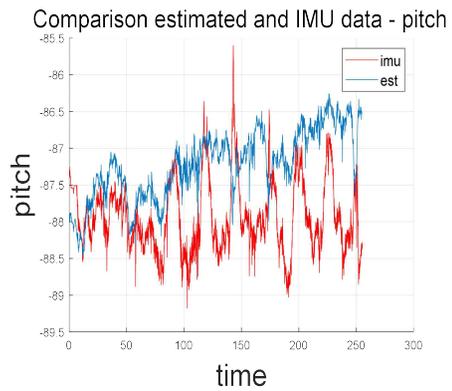
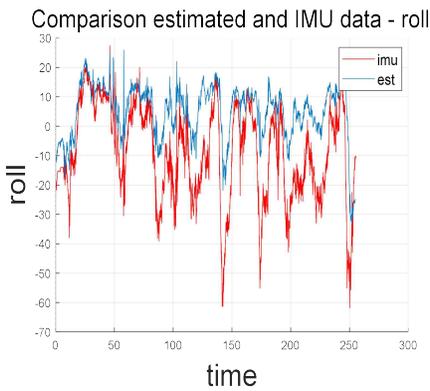


(b) 스테레오 카메라의 특징점 추출

그림 5.19 천장을 향한 경우 시각 관성 주행 거리 측정 시각화



(a) 시각 관성 거리 측정을 통해 추정된 위치 그래프



(b) 시각 관성 주행 거리 측정을 통해 추정된 자세 그래프

그림 5.20 천장을 향한 시각 관성 주행 거리 측정의 결과 추정된 위치 및 자세 그래프

시각 관성 주행 거리 측정의 결과 경로는 약 4m×4m로 트랙 크기인 4m×4m에서 5m×5m와 거의 일치하고, 자세의 경우 추정된 결과와 IMU 센서값의 오차가 바닥을 향한 경우보다 크지 않지만, 약 15도 정도의 오차를 보인다. 이는 마찬가지로 bag 파일을 저장하는 과정에서 노트북의 성능으로 인해 데이터가 끊겨 들어와 정확도에 영향을 미치고 추정된 결과의 드리프트 때문에 발생한다. 카메라를 천장을 향하게 하고 RTAB-Map을 실행한 결과는 그림 5.21과 같다.

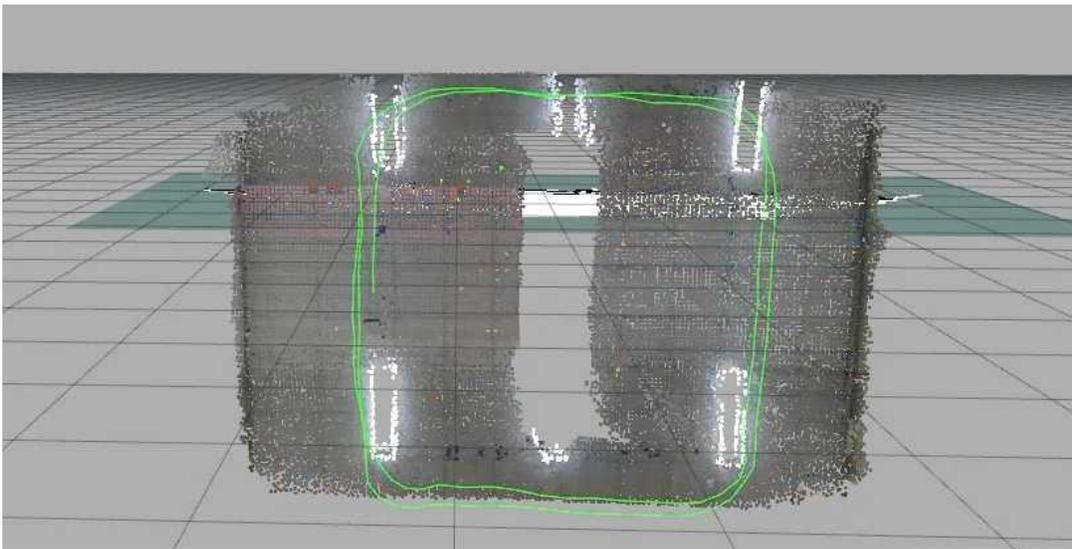
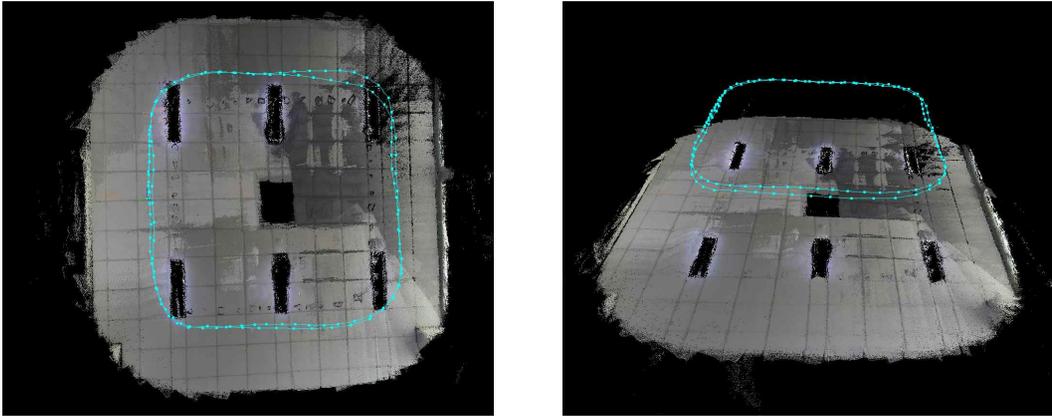


그림 5.21 천장을 향했을 때 RTAB-Map을 사용한 3D 지도 작성 결과

카메라가 천장을 향했을 때 RTAB-Map 결과 바닥을 향한 경우와 마찬가지로 천장이 90도로 서 있다고 생각하기 때문에 지도가 90도로 기울여 생성되었다. 또한, 루프 클로저가 작동하지 않은 것을 확인할 수 있다. 보통 초기 위치로 돌아오면 루프 클로저가 작동해야 하지만, 천장의 경우 모든 면이 비슷하게 생겼기 때문에 새로 들어온 이미지가 기존의 이미지인지 새로운 이미지인지 판단할 수 없어 작동하지 않았다. 그림 5.21처럼 평면에서 보았을 때는 지도가 한 층으로 보였지만, 그림 5.22와 같이 측면에서 보면 한 면으로 생성되어야 할 지도가 루프 클로저가 실행되지 않아 여러 층으로 생성된 것을 볼 수 있다. RTAB-Map의 경우 3가지 경우를 모두 비교해 보았을 때, 카메라가 천장을 향한 경우 성능이 가장 좋지 않게 나타난다.

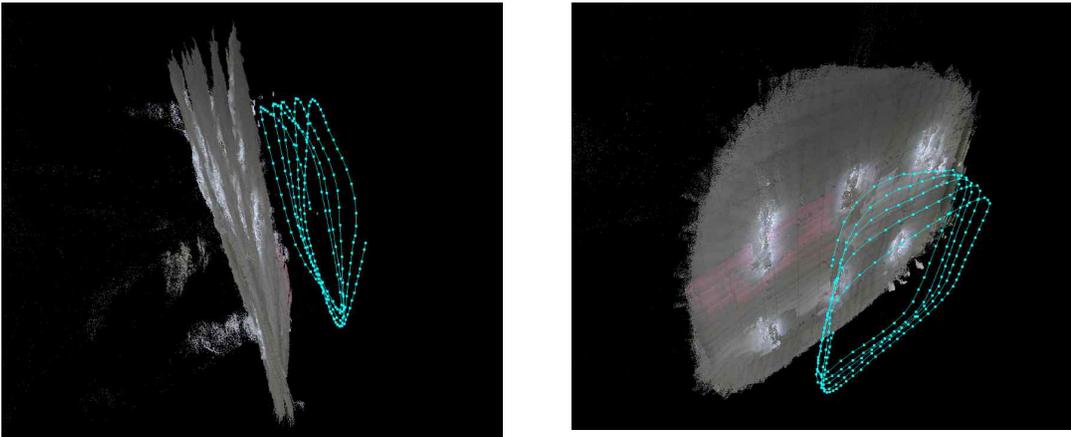


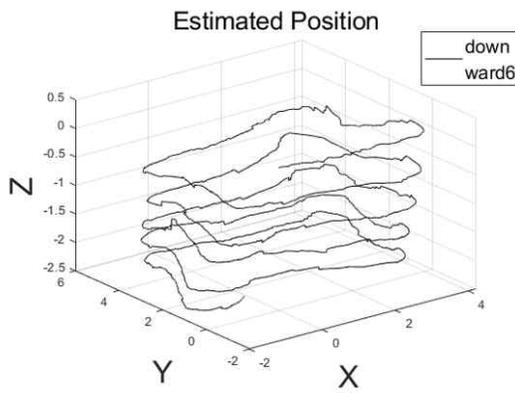
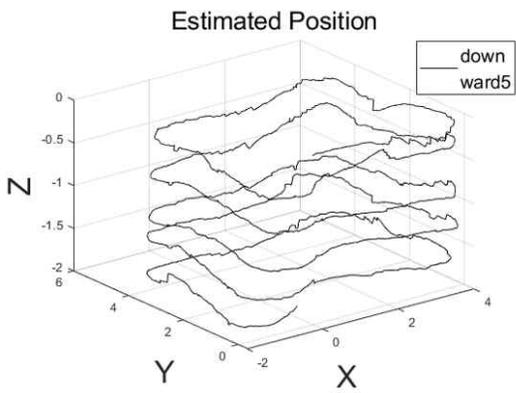
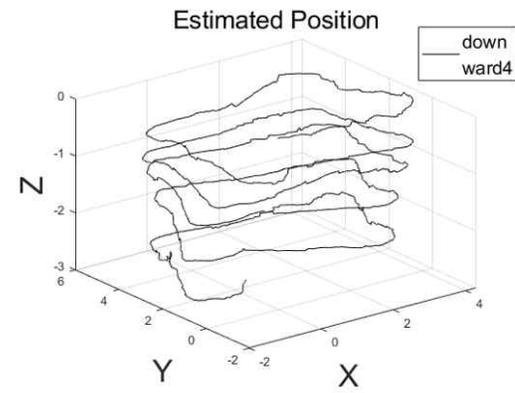
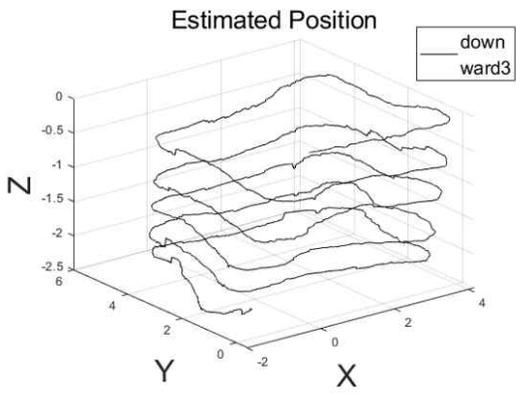
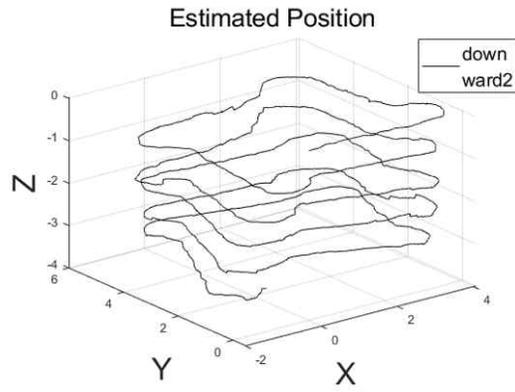
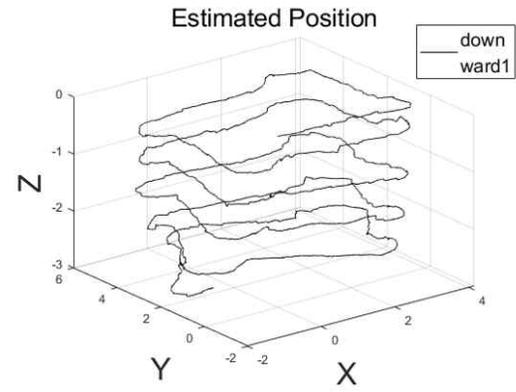
그림 5.22 루프 클로저가 작동하지 않은 천장 지도 작성 결과

## 제3절 시각 관성 주행 거리 측정 성능 비교

앞서 진행한 실험과 다르게, 더 많은 실험을 통해 각 방향에 대한 시각 관성 주행 거리 측정에 대한 성능을 비교하였다. 이를 위해 앞의 실험과 동일한 조건으로 진행하였다. 또한, 카메라를 카트에 설치해 사람이 직접 끌면서 실험하기 때문에 경로가 같을 순 없지만, 카메라가 움직인 경로를 최대한 같게 하여 실제 경로를 최대한 동일하게 하였다. 실험은 바닥, 정면, 그리고 천장에 대해 각각 트랙 5 바퀴를 도는 실험을 7번씩 진행하였다.

### 1. 카메라가 바닥을 향한 경우

그림 5.23은 카메라가 바닥을 향한 경우 추정된 위치에 대한 X-Y-Z 3차원 그래프이다. 카메라가 바닥을 향한 경우 창문의 블라인드는 내리고, 조명은 켜 상태로 실험을 진행하였다. 카메라의 높이는 고정되어있으므로 z축에 대한 값은 0으로 유지되어야 한다. 하지만, 실험 결과 카메라가 바닥을 향한 7번의 실험에 대한 z축의 평균은  $-1.2181[m]$ 로 오차가 상당히 큰 것을 볼 수 있다. 또한, 그림 5.24는 7번의 실험에 대한 z축의 통계를 나타내고, 평균 최댓값은  $0[m]$ , 평균 최솟값은  $-2.55[m]$ , 그리고 평균 표준편차는  $0.7552[m]$ 이다.



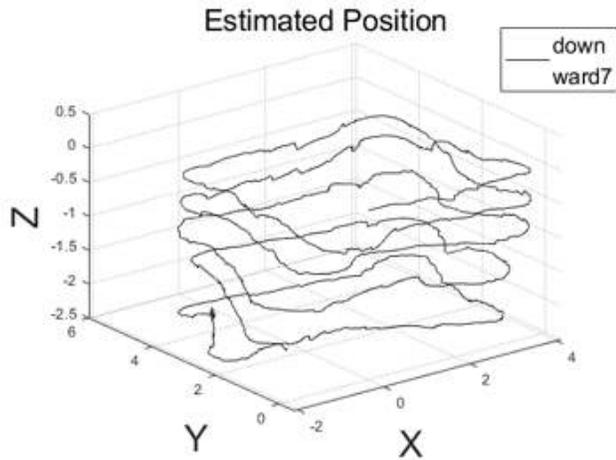


그림 5.23 바닥을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프

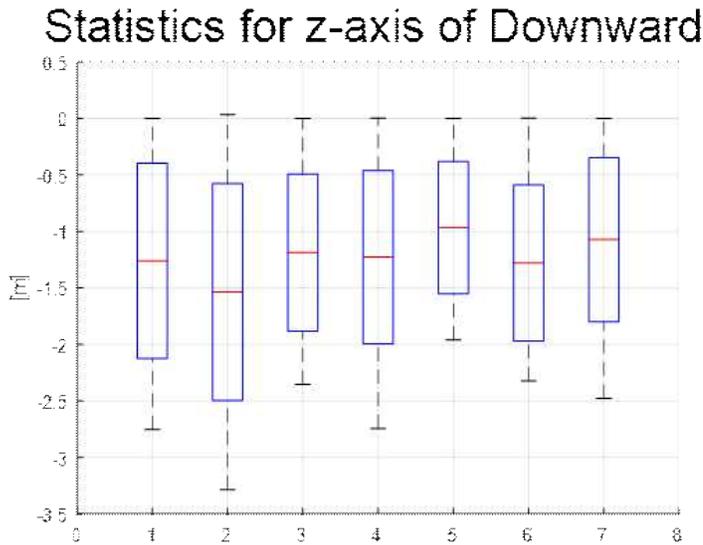
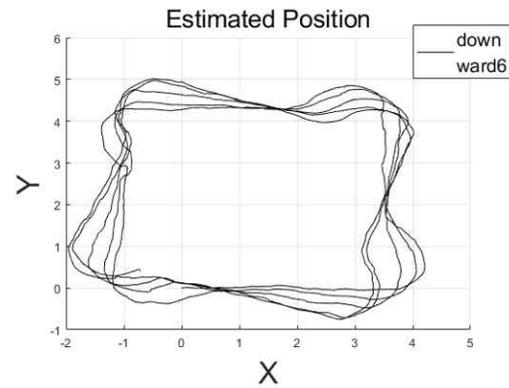
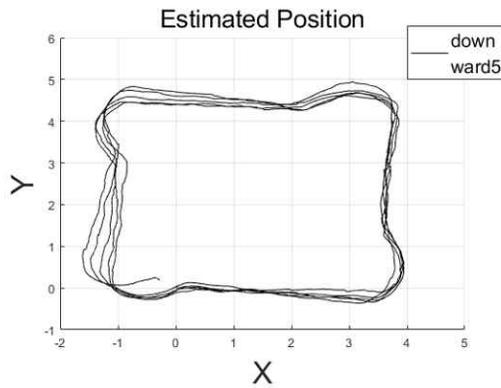
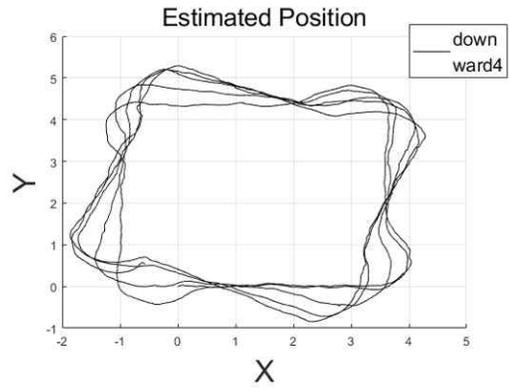
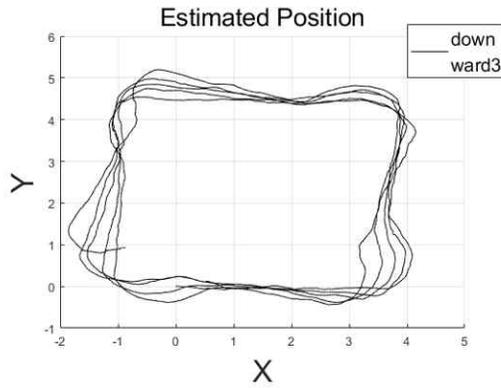
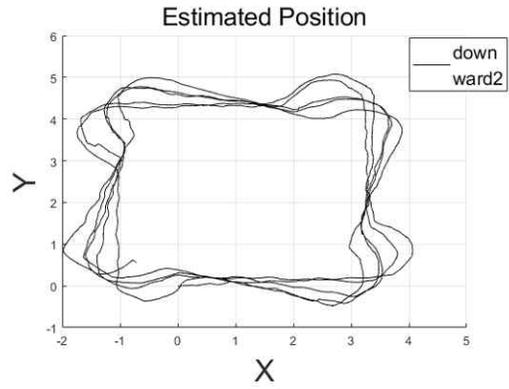
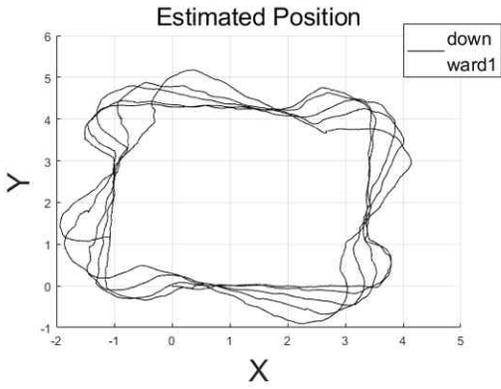


그림 5.24 바닥을 향한 경우 7번의 실험에 대한 z축 통계

여기서, 빨간 실선은 평균값, 파란색 직사각형의 윗변은 평균값에 표준편차를 더한 값이고, 아랫변은 평균값에서 표준편차를 뺀 값이다. 또한, 위쪽의 검은 가로 실선은 최댓값, 그리고 아래쪽 검은 가로 실선은 최솟값을 의미한다. 그림 5.24에서 나타났듯이 오차가 매우 큰 것을 볼 수 있다. 이는 카메라가 바닥을 향했을 때 생기는 바닥에서의 빛 반사로 특징점 추출에 실패하여 측정값이 다중 상

태 제약 칼만 필터에 입력되지 않아 time update만 계속해서 진행되고 measurement update는 진행되지 않기 때문이다. 따라서 공분산 값이 점점 커져 오차가 증가하게 되어 드리프트가 발생하게 된다[48]. 이러한 이유로 드리프트가 발생하여 높이가 0[m]로 고정되지 않고, 오차가 누적되어 점점 커지게 된다.

그림 5.25의 X-Y 2차원 그래프를 보면 아래쪽으로 뿐만 아니라 같은 경로로 이동하였는데도 불구하고 경로가 일정하지 않은 것을 볼 수 있다. 경로의 모서리 부분의 굴곡은 실험을 진행할 때 카메라가 바닥을 향하도록 설치하면 카트 일부가 영상에 들어가므로 더 앞에 설치함으로써 생기게 된다. 앞의 3차원 그래프와 마찬가지로 x, y축에 대해서도 드리프트가 발생하여 하나의 경로로 수렴하지 않는 것을 볼 수 있다. 이는 앞서 설명했듯이 빛 반사로 인한 특징점 추출에 실패함으로써 생긴 공분산 증가로 인한 드리프트 때문이다.



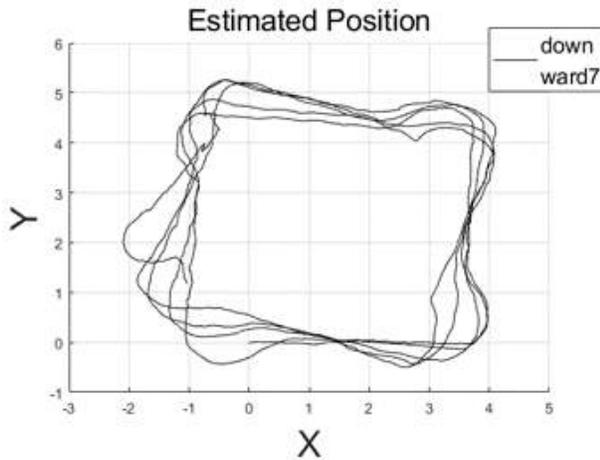
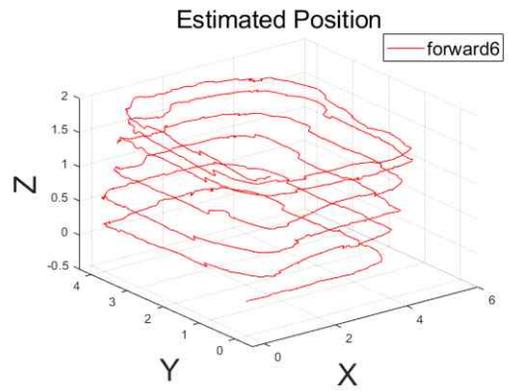
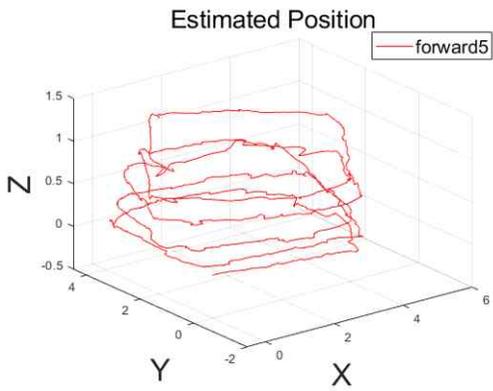
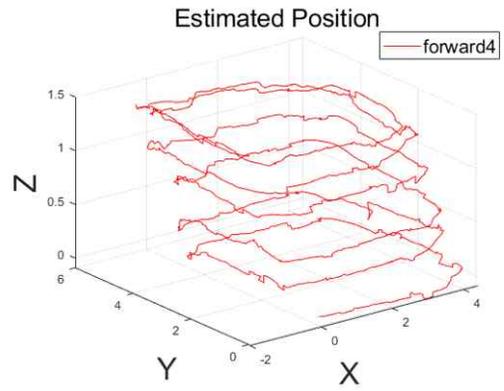
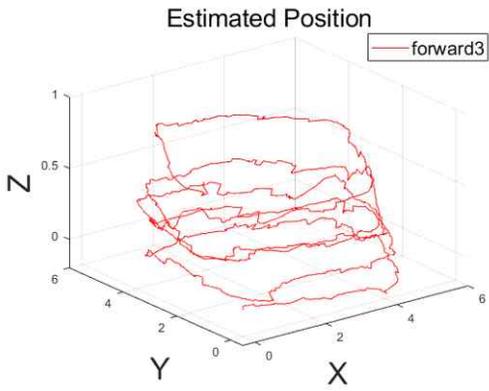
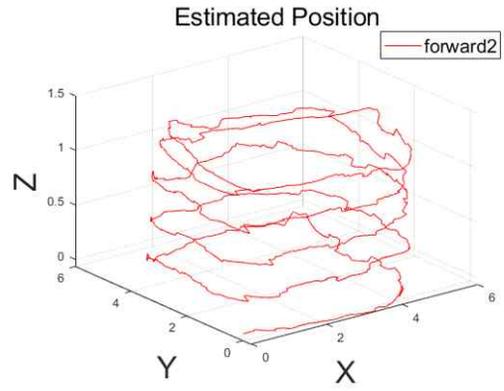
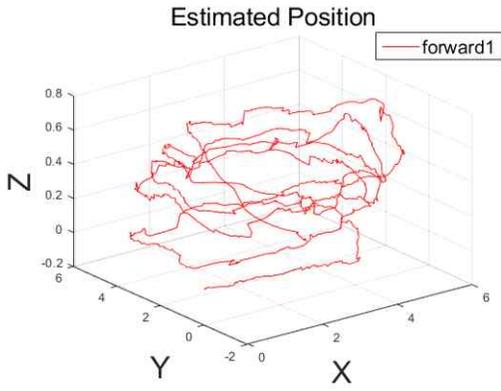


그림 5.25 바닥을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프

## 2. 카메라가 정면을 향한 경우

그림 5.26은 카메라가 정면을 향한 경우 추정된 위치에 대한 X-Y-Z 3차원 그래프이다. 카메라가 정면을 향한 경우도 마찬가지로 창문의 블라인드는 내리고, 조명은 켜 상태로 실험을 진행하였다. 정면 실험도 카메라의 높이는 고정되어있으므로 z축에 대한 값은 0으로 유지되어야 한다. 하지만, 실험 결과 카메라가 정면을 향한 7번의 실험에 대한 z축의 평균은 0.5853[m]로 오차가 상당히 큰 것을 볼 수 있다. 또한, 그림 5.27은 7번의 실험에 대한 z축의 통계를 나타내고, 평균 최댓값은 1.26[m], 평균 최솟값은 -0.10[m], 그리고 평균 표준편차는 0.3906[m]이다.



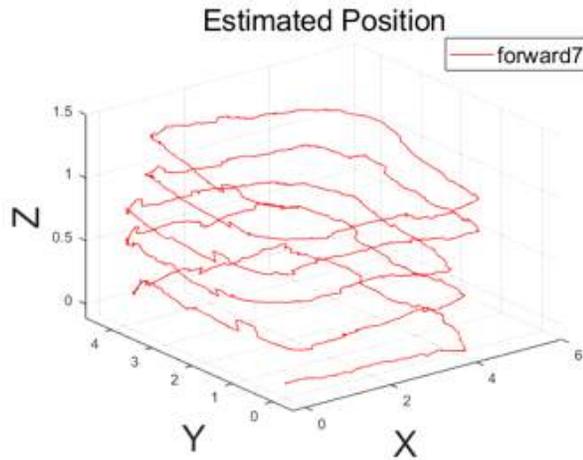


그림 5.26 정면을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프

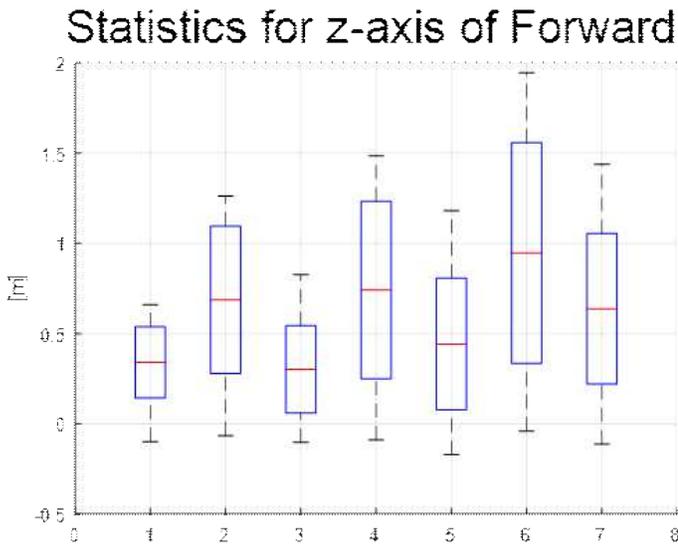
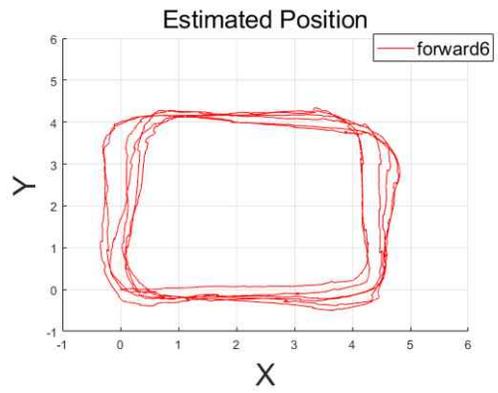
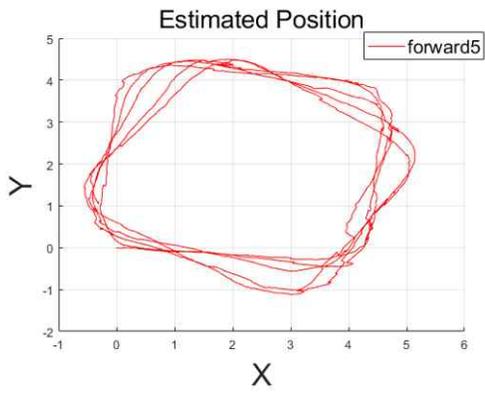
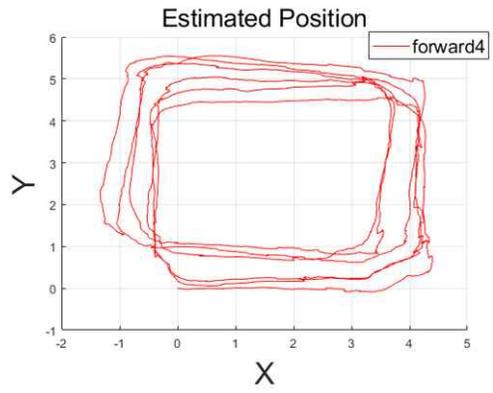
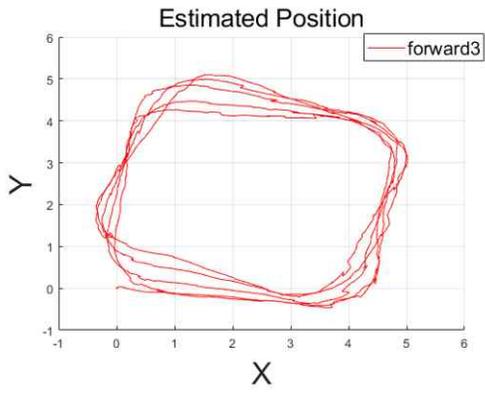
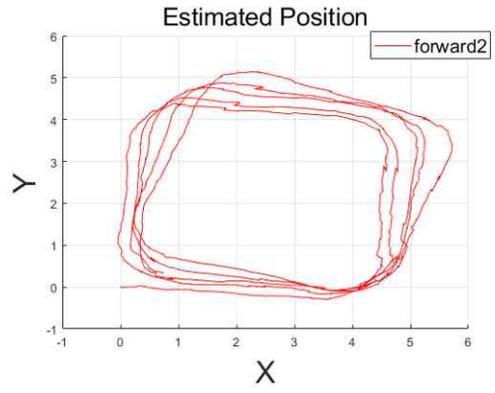
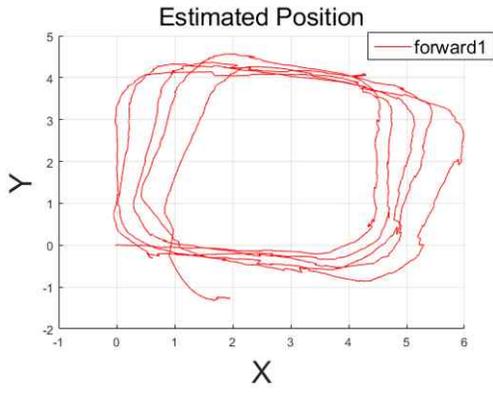


그림 5.27 정면을 향한 경우 7번의 실험에 대한 z축 통계

바닥의 경우에는 빛 반사로 인한 특징점 추출에 실패함으로써 공분산이 증가하여 생기는 드리프트로 오차가 크게 나타났지만, 정면의 경우에는 실험 환경의 벽으로 인해 모서리를 돌 때 카메라가 외벽에 막혀 특징점을 추출할 수 없기 때문에 생기게 된다. 따라서 벽에 막혀 특징점 추출 실패로 드리프트가 발생하면서 높이가 0[m]로 고정되지 않고, 위아래로 흔들리는 결과를 얻게 되었다.

그림 5.28의 X-Y 2차원 그래프를 보면 경로가 일정하지 않은 것을 볼 수 있다. 정면의 경우 카메라를 카트에 직접 설치하였기 때문에 바닥을 향한 경우와는 다르게 모서리 부분의 굴곡이 생기지 않는다. 위의 그림을 통해 앞의 3차원 그래프와 마찬가지로  $x$ ,  $y$ 축에 대해서도 드리프트가 발생하여 하나의 경로로 수렴하지 않는 것을 볼 수 있다.



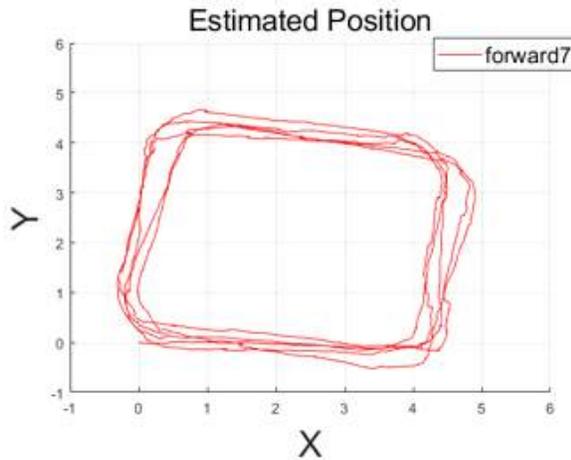
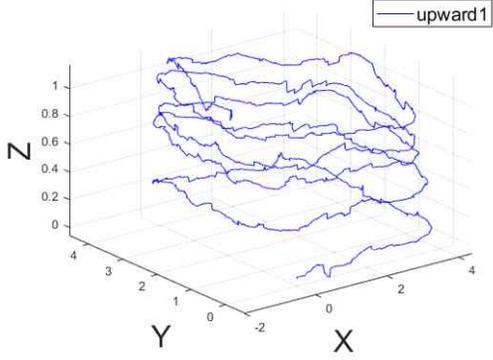


그림 5.28 정면을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프

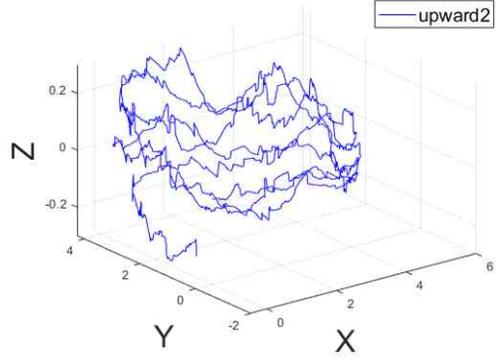
### 3. 카메라가 천장을 향한 경우

그림 5.29는 카메라가 천장을 향한 경우 추정된 위치에 대한 X-Y-Z 3차원 그래프이다. 카메라가 천장을 향한 경우는 다른 경우와는 다르게 카메라에 빛이 많이 들어가기 때문에 창문의 블라인드는 내리고, 조명은 끈 상태로 실험을 진행하였다. 천장 실험도 카메라의 높이는 고정되어있으므로 z축에 대한 값은 0으로 유지되어야 한다. 하지만, 실험 결과 카메라가 정면을 향한 7번의 실험에 대한 z축의 평균은 0.1076[m]로 앞의 다른 두 경우보다 오차가 작은 것을 볼 수 있다. 또한, 그림 5.30은 7번의 실험에 대한 z축의 통계를 나타내고, 평균 최댓값은 0.42[m], 평균 최솟값은 -0.19[m], 그리고 평균 표준편차는 0.1267[m]이다.

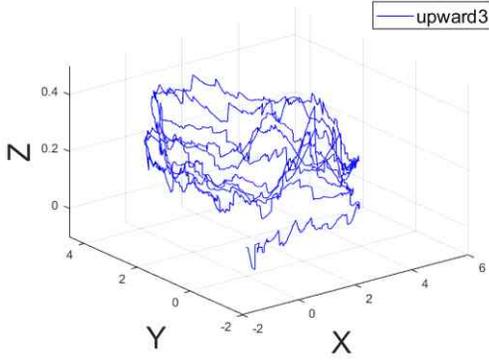
Estimated Position



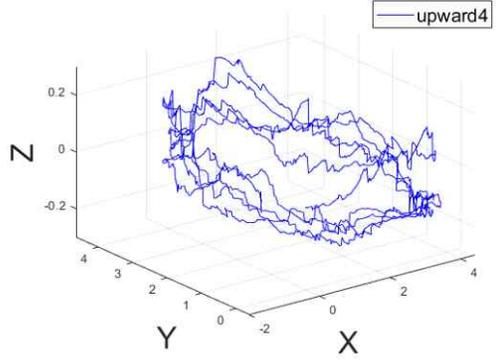
Estimated Position



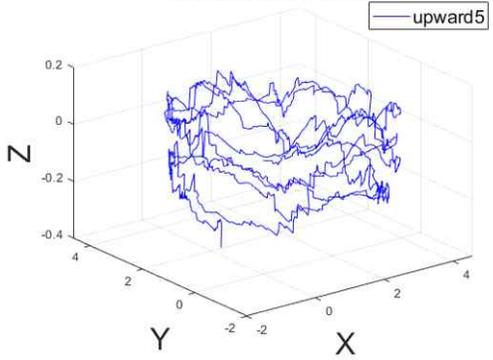
Estimated Position



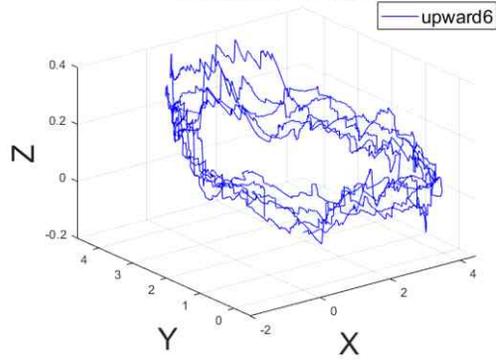
Estimated Position



Estimated Position



Estimated Position



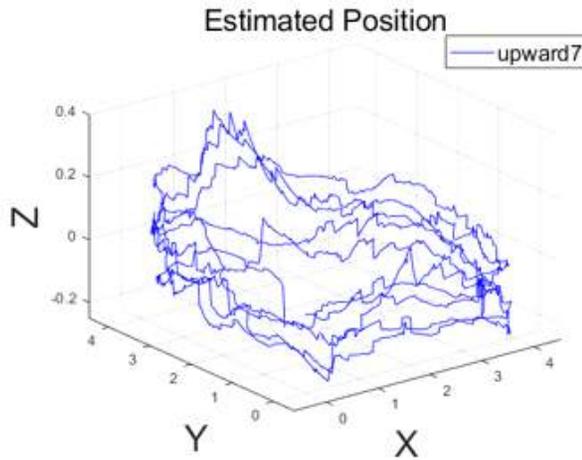


그림 5.29 천장을 향한 경우 5바퀴에 대한 추정된 위치의 3차원 그래프

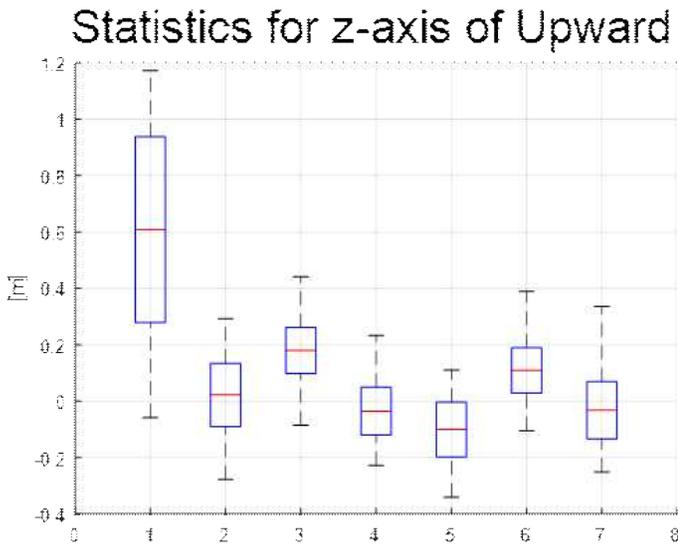
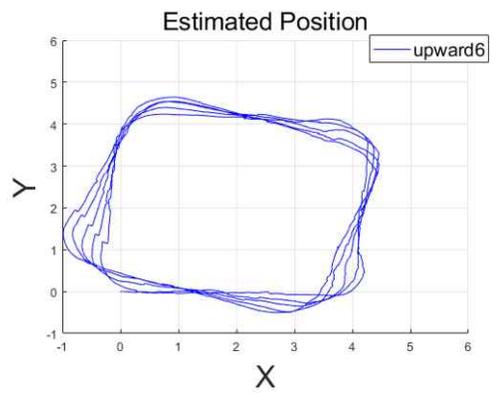
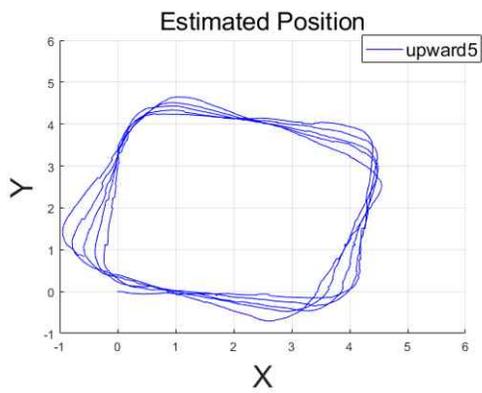
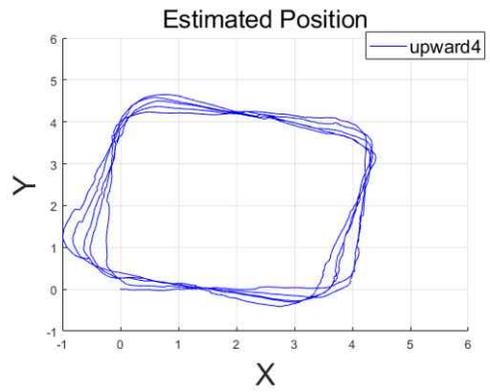
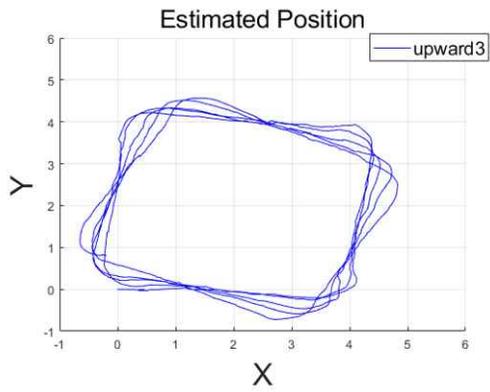
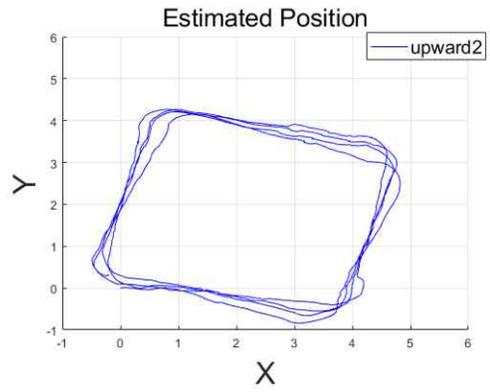
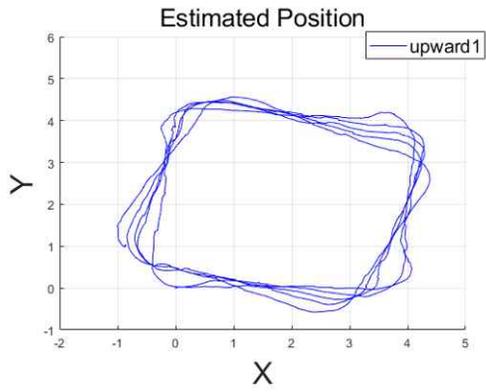


그림 5.30 천장을 향한 경우 7번의 실험에 대한 z축 통계

다른 경우들에 비해 카메라가 천장을 향했을 때 성능이 가장 좋게 나타난다. 이는 천장의 경우 바닥의 경우처럼 빛 반사가 일어나지 않고, 정면의 경우처럼 벽이 앞을 가로막지 않는다. 따라서 다른 경우들에 비해 특징점 추출에 실패하는 경우가 적다. 그럼에도 드리프트가 발생하는 이유는 카메라가 천장을 향하면 빛이 카메라 렌즈로 직접적으로 들어와 빛의 영향으로 인해 영상이 밝아져 아무것도 잡

히지 않아 특징점 추출에 실패하기 때문이다. 그림 5.31의 X-Y 2차원 그래프를 보면 경로가 일정하지 않은 것을 볼 수 있다. 천장의 경우도 마찬가지로 카메라를 카트에 직접 설치하였기 때문에 바닥을 향한 경우와는 다르게 모서리 부분의 굴곡이 생기지 않는다. 3차원 그래프와 같이  $x$ ,  $y$ 축에 대해서도 드리프트가 생긴 것을 볼 수 있다. 하지만, 바닥, 정면의 경우보다는 경로 수렴도가 좋은 것을 알 수 있다.



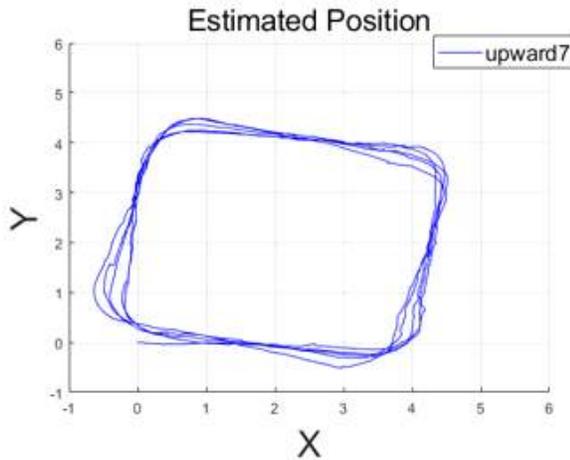


그림 5.31 천장을 향한 경우 5바퀴에 대한 추정된 위치의 X-Y 그래프

#### 4. 각 방향에 대한 시각 관성 주행 거리 측정 성능 비교

앞서 진행한 세 가지 방향에 대한 시각 관성 주행 거리 측정의 성능을 비교한다. 성능은 z축 기준으로 비교하였고, z축에 대한 평균, 최댓값, 최솟값, 그리고 표준편차를 구해 나타냈다. 표 5-6은 바닥, 정면, 그리고 천장 방향의 각 실험에 대한 z축 기준 평균을 나타내고, 표 5-7은 세 가지 방향의 각 실험에 대한 최댓값, 최솟값 및 최댓값과 최솟값의 차의 크기를 보여준다. 마지막으로 표 5-8은 세 가지 방향의 각 실험에 대한 표준편차를 나타낸다.

표 5-6을 보면, 바닥, 정면, 그리고 천장을 향한 경우 평균은 각각  $-1.2181[m]$ ,  $0.5853[m]$ ,  $0.1076[m]$ 로 0에 제일 가까운 천장을 향한 경우 오차가 제일 적은 것을 알 수 있다. 반면 카메라가 바닥을 향한 경우 오차가 제일 큰 것으로 나타난다. 표 5-7에서는 바닥을 향한 경우 최댓값은  $0[m]$ , 최솟값은  $-2.55[m]$ 로 최댓값과 최솟값 차의 크기는  $2.56[m]$ 이고, 정면을 향한 경우 최댓값은  $1.26[m]$ , 최솟값은  $-0.10[m]$ 로 최댓값과 최솟값 차의 크기는  $1.35[m]$ 로 나타난다. 마지막으로 천장을 향한 경우 최댓값은  $0.42[m]$ , 최솟값은  $-0.19[m]$ 로 최댓값과 최솟값 차의 크기는  $0.62[m]$ 이다. 따라서, 천장을 향한 경우 최댓값과 최솟값 차의 크기가  $0.62$ 로 제일 작게 나타났다. 또한, 표 5-8을 보면, 바닥의 경우 표준편차의 평균이  $0.7552[m]$ , 정면의 경우는  $0.3906[m]$ , 그리고 천장의 경우

0.1267[m]로 나타난다. 바닥의 경우가 표준편차의 평균이 가장 크고, 천장의 경우가 가장 작다. 표 6-6, 표 6-7, 그리고 표 6-8을 보면 바닥을 향한 경우의 성능이 가장 떨어지고 천장을 향한 경우의 성능이 가장 좋은 것을 확인할 수 있다.

표 5-6. 각 방향에 대한 z축 기준 평균

| 방향 |   | 방향             |             |               |
|----|---|----------------|-------------|---------------|
|    |   | Downward (m)   | Forward (m) | Upward (m)    |
| 횟수 | 1 | -1.2598        | 0.3410      | 0.6087        |
|    | 2 | -1.5351        | 0.6869      | 0.0225        |
|    | 3 | -1.1871        | 0.3013      | 0.1805        |
|    | 4 | -1.2273        | 0.7419      | -0.0352       |
|    | 5 | -0.9664        | 0.4422      | -0.1002       |
|    | 6 | -1.2786        | 0.9466      | 0.1092        |
|    | 7 | -1.0727        | 0.6372      | -0.0320       |
| 평균 |   | <b>-1.2181</b> | 0.5853      | <b>0.1076</b> |

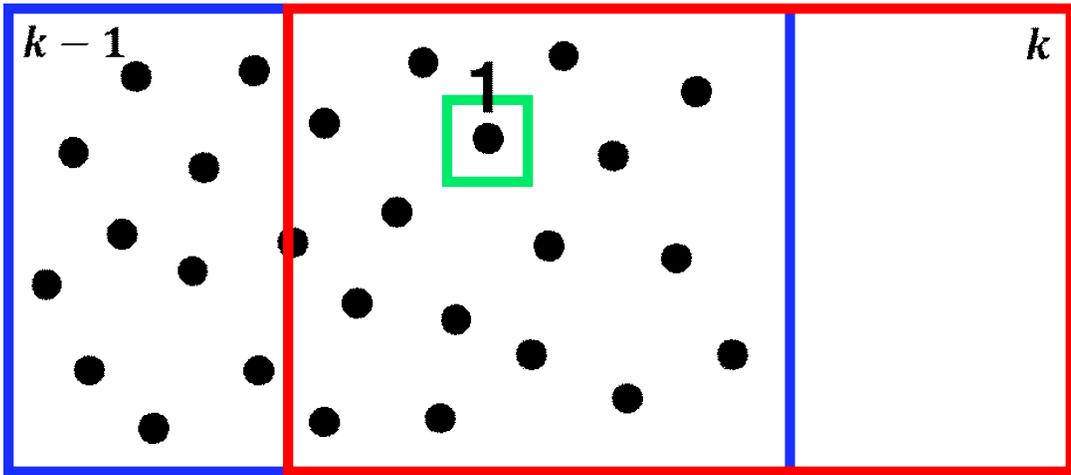
표 5-7. 각 방향에 대한 z축 기준 최댓값 및 최솟값

| 방향 |   | Downward(m) |       |             | Forward(m) |       |         | Upward(m) |       |             |
|----|---|-------------|-------|-------------|------------|-------|---------|-----------|-------|-------------|
|    |   | max         | min   | max-min     | max        | min   | max-min | max       | min   | max-min     |
| 횟수 | 1 | 0           | -2.75 | 2.75        | 0.66       | -0.10 | 0.76    | 1.17      | -0.06 | 1.23        |
|    | 2 | 0.04        | -3.28 | 3.32        | 1.26       | -0.07 | 1.33    | 0.29      | -0.28 | 0.57        |
|    | 3 | 0           | -2.35 | 2.35        | 0.83       | -0.10 | 0.93    | 0.44      | -0.08 | 0.53        |
|    | 4 | 0           | -2.74 | 2.74        | 1.48       | -0.09 | 1.57    | 0.23      | -0.23 | 0.46        |
|    | 5 | 0           | -1.96 | 1.96        | 1.18       | -0.17 | 1.35    | 0.11      | -0.34 | 0.45        |
|    | 6 | 0           | -2.32 | 2.33        | 1.94       | -0.04 | 1.98    | 0.39      | -0.10 | 0.49        |
|    | 7 | 0           | -2.48 | 2.48        | 1.44       | -0.11 | 1.55    | 0.34      | -0.25 | 0.59        |
| 평균 |   | 0           | -2.55 | <b>2.56</b> | 1.26       | -0.10 | 1.35    | 0.42      | -0.19 | <b>0.62</b> |

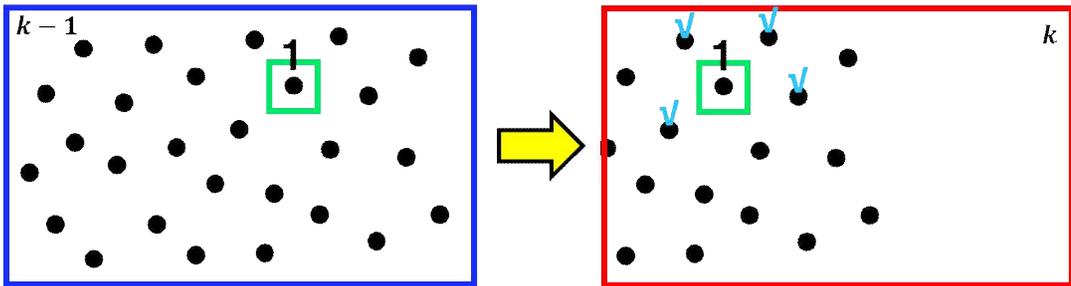
표 5-8. 각 방향에 대한 z축 기준 표준편차

| 방향       |   | Downward(m)   | Forward(m) | Upward(m)     |
|----------|---|---------------|------------|---------------|
|          |   | 횟수            |            |               |
| 표준<br>편차 | 1 | 0.8628        | 0.1973     | 0.3293        |
|          | 2 | 0.9590        | 0.4090     | 0.1119        |
|          | 3 | 0.6963        | 0.2417     | 0.0819        |
|          | 4 | 0.7668        | 0.4918     | 0.0849        |
|          | 5 | 0.5857        | 0.3655     | 0.0968        |
|          | 6 | 0.6909        | 0.6120     | 0.0803        |
|          | 7 | 0.7249        | 0.4167     | 0.1016        |
| 평균       |   | <b>0.7552</b> | 0.3906     | <b>0.1267</b> |

또한, 바닥과 정면의 경우에는 천장에 비해 랜드마크가 더 많음에도 불구하고 시각 관성 주행 거리 측정에 대한 성능은 천장의 경우가 가장 좋게 나타났다. 이는 data association과 관련이 있다. Data association은 이전 시간의 이미지에서 추출된 특징점과 현재 추출된 특징점을 서로 연관시키는 것으로, 이전 시간의 이미지에서 추출된 특징점과 현재 추출된 특징점이 일치하는지를 찾는다. 이 과정에서 특징점이 과도하게 많아지면 성능이 더 낮아질 수 있다. 예를 들어, 그림 5.32와 같이 추출된 특징점의 개수가 과도하게 많으면 알고리즘에서 이전 시간 k-1에서 추출된 1번 점이 현재 시간 k에서 추출된 1번 점과 같은 것인지 확인하는 과정에서 정확하지 않은 판단을 할 가능성이 커지게 된다. 또는, 추출된 특징점들이 가까이 모여있으면 마찬가지로 정확하지 않은 판단을 할 가능성이 커진다.



(a) 이전 시간과 현재 시간에서의 같은 특징점



(b) 주변의 많은 특징점으로 인해 Data Association에 실패할 수 있는 상황  
 그림 5.32 특징점 추출 과정에서의 Data Association

## 제6장 결론

본 논문은 자율주행을 위한 스테레오 카메라와 IMU를 사용한 시각 관성 주행 거리 측정으로 위치 및 자세 추정과 RTAB-Map을 사용한 3D 지도 작성을 진행하였다. 기존의 시각 관성 주행 거리 측정과 3D 지도 작성은 일반적으로 카메라를 정면을 향하게 한 실험 한 가지 경우에 대해 진행하였고, 어느 방향을 보고 진행하는지는 중요한 것이 아니었다. 하지만, 본 논문에서는 카메라가 바닥, 정면, 그리고 천장을 향하도록 하는 실험을 모두 진행하여 3가지 경우의 특성을 파악하였고, 그 성능에 대해 분석하였다. 또한, 카메라가 바닥을 향한 실험을 함으로써 지상에서 움직이는 이동체와 드론뿐만 아니라 수중에서 작업하는 지형 탐지 로봇 등에도 적용할 수 있다.

실험 결과, 시각 관성 주행 거리 측정의 경우에는 z축에 대한 통계를 살펴보았을 때, 카메라가 천장을 향하는 경우 오차가 3가지 경우 중 가장 적어 성능이 가장 좋게 나타났고, 3D 지도 작성은 주변이 비슷하게 생겨 새로 들어온 이미지가 새로운 이미지인지 기존의 이미지인지 판단이 잘되지 않는 곳은 루프 클로저가 이루어지지 않아 생성된 지도가 겹겹이 쌓이는 것을 확인하였다. 이는 시각 관성 주행 거리 측정 알고리즘에는 루프 클로저의 개념이 없어 시간이 흐르면 오차가 쌓이게 되면서 점점 오차가 커지게 될 뿐만 아니라, 하드웨어의 성능과 주변 환경으로 인해 특징점이 추출되지 않으면 드리프트가 발생하고 공분산이 커지는 특성이 있기 때문이다. 또한, RTAB-Map은 루프 클로저의 기능은 있지만 카메라가 천장을 향한 실험에서와 같이 주변의 환경이 비슷한 경우 등 상황에 따라 실행되지 않을 수 있기 때문이다.

향후에는, 본 논문에서는 시각 관성 주행 거리 측정과 3D 지도 작성을 따로 진행했지만, 이를 동시에 사용하여 상태 추정 및 지도 작성의 정확성과 안정성을 높이는 연구를 수행할 계획이다. 그리고 3D 지도 작성에 대해 카메라뿐 아니라 LiDAR를 동시에 사용하여 정확도를 더 높이는 연구를 진행하여 시각 관성 주행 거리 측정과 함께 진행하면 더 좋은 결과를 도출해낼 수 있을 것이다. 또한, 하드웨어의 성능을 높여 제시한 시각 관성 주행 거리 측정과 RTAB-Map을 사용한 3D 지도 작성을 사용한 자율주행에 대해 연구할 계획이다.

## 참고문헌

- [1] 정광욱, and 유지환, “자율주행자동차의 실시간 고정밀 위치추정을 위한 파티클필터와 확장칼만필터 융합 위치추정 알고리즘 개발,” 제어로봇시스템학회 논문지, vol. 25, no. 6, pp. 526–533, 2019.
- [2] 이재환, and 최진규, “자율 주행 로봇의 실시간 위치 추정에 관한 연구,” 한국정보기술학회논문지, vol. 11, no. 12, pp.29–34, 2013.
- [3] Kiwon Yeom, “Kinematic and Dynamic Controller Design for Autonomous Driving of Car-like Mobile Robot,” International Journal of Mechanical Engineering and Robotics Research, vol. 9, no. 7, pp. 1085–1064, 2020.
- [4] 김시습, 허준호, 양성모, and 기창두, “광류 및 초음파 센서를 이용한 모바일 로봇의 위치 추정 및 자율주행,” 한국정밀공학회 학술발표대회 논문집, pp. 511–512, 2008.
- [5] M. M. Mostafa, A. M. Moussa, N. E. Sheimy, and A. B. Sesay, “A smart hybrid vision aided inertial navigation system approach for UAVs in a GNSS denied environment,” NAVIGATION Journal of the Institute of Navigation, vol. 65, no. 4, pp. 533–547, 2018.
- [6] K. Wong, Y. Gu, and S. Kamijo, “Mapping for Autonomous Driving: Opportunities and Challenges,” IEEE Intelligent Transportation Systems Magazine, vol. 13, no. 1, pp. 91–106, 2020.
- [7] 노성우, 김태균, and 고낙용, “로봇 위치 예측에 기반을 둔 ICP 알고리즘을 이용한 지도 작성,” 한국전자통신학회논문지, vol. 8, no. 4, pp. 575–582, 2013.
- [8] 도정찬, 성병문, and 이왕현, “3차원 라이더를 이용한 자율주행차의 지도 작성과 위치 추정에 관한 연구,” 제어로봇시스템학회 국내학술대회 논문집, pp. 137–139, 2019.
- [9] Y. Li, and J. I. Guzman, “Lidar for Autonomous Driving: The principles, challenges, and trends for automotive lidar and perception systems,” IEEE Signal Processing Magazine, vol. 37, no. 4, pp. 50–61, 2020.
- [10] 박준수, and 박순용, “영상 특징점의 거리값을 이용한 Stereo Visual Inertial,” 전자공학회논문지, vol. 58, no. 7, pp. 77–87, 2021.
- [11] A. I. Mourikis, and S. I. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation,” IEEE International Conference on Robotics and Automation, vol. 2, pp. 3565–3572, 2007.

- [12] D. G. Kottas, and S. I. Roumeliotis, “Efficient and Consistent Vision-aided Inertial Navigation using Line Observations,” IEEE International Conference on Robotics and Automation, pp. 1540–1547, 2013.
- [13] Y. Yang, P. Ceneva, K. Ekenhoff, and G. Huang, “Visual-Inertial Odometry with Point and Line Features,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5447–2454, 2019.
- [14] K. Lenac, A. Kitanov, R. Cupee, and I. Petrović, “Fast Planar Surface 3D SLAM Using LIDAR,” Robotics and Autonomous Systems, vol. 92, pp. 197–220, 2017.
- [15] J. M. Sáez, and F. Escolano, “A Global 3D Map-Building Approach Using Stereo,” IEEE International Conference on Robotics and Automation, vol. 2, pp. 1197–1202, 2004.
- [16] J. Li, X. Zhang, J. Li, Y. Liu, and J. Wang, “Building and optimization of 3D semantic map based on Lidar and camera fusion,” Neurocomputing, vol. 409, pp. 394–407, 2020.
- [17] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, “A Collaborative Approach to In-Place Sensor Calibration,” Information processing in sensor networks, pp. 301–316, 2003.
- [18] B. Fu, F. Han, Y. Wnag, Y. Jiao, X. Ding, Q. Tan, L. Chen, M. Wang, and R. Xiong, “High-Precision Multicamera-Assisted Camera-IMU Calibration: Theory and Method,” IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–17, 2021.
- [19] J. Weng, P. Cohen, and M. Herniou, “Calibration of Stereo Cameras using a Non-linear Distortion Model,” Proceedings. 10th International Conference on Pattern Recognition, vol. 1, pp. 246–253, 1990.
- [20] J. Weng, P. Cohen, and M. Herniou, “Camera Calibration with Distortion Models and Accuracy Evaluation,” IEEE Transactions on pattern analysis and machine intelligence, vol. 14, no. 10, pp. 965–980, 1992.
- [21] Q. Memon, and S. Khan, “Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks,” International Journal of Systems Science, vol. 32, no. 9, pp. 1155–1159, 2001.
- [22] J. Maye, P. Furgale, and R. Siegwart, “Self-supervised Calibration for

- Robotic Systems,” IEEE Intelligent Vehicles Symposium (IV), pp. 473-480, 2013.
- [23] S. Jennings, and S. Eshghi, “Stereo Camera-IMU Calibration,” <https://www.researchgate.net/publication/352712224>, June, 2019.
- [24] P. Furgale, J. Rehder, and R. Siegwart, “Unified Temporal and Spatial Calibration for Multi-Sensor Systems,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1280-1286, 2013.
- [25] L. B. Pupu, “Characterization of Errors and Noises in MEMS Inertial Sensors Using Allan Variance Method,” MS thesis. Universitat Politècnica de Catalunya, 2016.
- [26] D. W. Allan, “Statistics of Atomic Frequency Standards,” , Proceedings of the IEEE, vol. 54, no. 2, pp. 221-230, 1966.
- [27] M. Marinov, and Z. Petrov, “Allan Variance Analysis on Error Characters of Low-cost MEMS Accelerometer MMA8451Q,” , International conference of scientific paper AFASES, 2014.
- [28] N. E. Sheimy, H. Hou, and X. Niu, “Analysis and Modeling of Inertial Sensors using Allan Variance,” IEEE Transactions on instrumentation and measurement, vol. 57, no. 1, pp. 140-149, 2007.
- [29] W. Stockwell, “Angle Random Walk,” Application Note. Crossbow Technologies Inc, pp. 1-4, 2003.
- [30] H. V. Do, Y. H. Kim, Y. S. Kwon, S. H. Kang, H. J. Kim, and J. W. Song, “ An Adaptive Approach based on Multi-State Constraint Kalman Filter for UAVs,” 2021 21st International Conference on Control, Automation and Systems (ICCAS), pp. 481-485, 2021.
- [31] 차재혁, and 박찬국, “특징점 깊이를 고려한 시간 관성 주행거리계 기법 간 성능 비교,” 한국항공우주학회 학술발표회 초록집, pp. 257-258, 2019.
- [32] L. E. Clement, V. Peretroukhin, J. Lambert, and J. Kelly, “The Battle for Filter Supremacy: A Comparative Study of the Multi-State Constraint Kalman Filter and the Sliding Window Filter,” 2015 12th Conference on Computer and Robot Vision. IEEE, pp. 23-30, 2015.
- [33] M. Ramezani, and K. Khoshelham, “Vehicle Positioning in GNSS-Deprived Urban Areas by Stereo Visual-Inertial Odometry,” IEEE Transactions on Intelligent Vehicles, vol. 3, no. 2, pp. 208-217, 2018.
- [34] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C.

- J. Taylor, and V. Kumar, “Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.
- [35] M. Trajković, and M. Hedley, “Fast corner detection,” *Image and vision computing*, vol. 16, no. 2, pp. 75–87, 1998.
- [36] B. D. Lucas, and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” *Proceedings of Imaging Understanding Workshop*, pp. 121–130, 1981.
- [37] D. B. Gennery, “Stereo-Camera Calibration,” *Proceeding of Imaging Understanding Workshop*, pp. 101–107, 1979.
- [38] B. Kitt, A. Geiger, and H. Lategahn, “Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme,” *2010 IEEE Intelligent Vehicles Symposium*. IEEE, pp. 486–492, 2010.
- [39] J. Civera, O. G. Grasa, J. Davision, and J. M. M. Montiel, “1-Point RANSAC for EKF-Based Structure from Motion,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3498–3504, 2009.
- [40] S. Zandara, and A. E. Nicholson, “Square Root Unscented Particle Filtering for Grid Mapping,” *Australasian Joint Conference on Artificial Intelligence*, pp. 121–130, 2009.
- [41] 최장혁, 이진선, 송재복, 정우진, 김문상, 박성기, and 최종석, “이동로봇의 주행을 위한 토폴로지컬 지도의 작성,” *제어자동화시스템공학 논문지*, vol. 8, no. 6, pp. 492–497, 2002.
- [42] E. G. Fidalgo, and A. Ortiz, “Vision-based topological mapping and localization methods: A survey,” *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, 2015.
- [43] B. Kaleci, C. M. Senler, O. Parlaktuna, and U. Gürel, “Constructing Topological Map from Metric Map Using Spectral Clustering,” *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 139–145, 2015.
- [44] H. J. Chang, C. S. G. Lee, Y. C. Hu, and Y. H. Lu, “Multi-Robot SLAM with Topological/Metric Maps,” *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1467–1472, 2007.
- [45] N. Ragot, R. Khemmar, A. Pokala, R. Rossi, and J. Y. Ertaud,

- “Benchmark of Visual SLAM Algorithms: ORB-SLAM2 vs RTAB-Map,” 2019 Eighth International Conference on Emerging Security Technologies (EST). IEEE, pp. 1-6, 2019.
- [46] M. Labbé, and F. Michaud, “Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation,” IEEE Transactions on Robotics, vol. 29, no. 3, pp. 734-745, 2013.
- [47] D. Scaramuzza, and F. Fraundorfer, “Visual Odometry [tutorial],” IEEE robotics & automation magazine, vol. 18, no. 4, pp. 80-92, 2011.
- [48] D. G. Kottas, K. J. Wu, and S. I. Roumeliotis, “Detecting and Dealing with Hovering Maneuvers in Vision-aided Inertial Navigation Systems,” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 3172-3179, 2013.