

# 2023 전기 졸업과제

## 중간 보고서

[C 네트워크/시스템 분과]

### LiDAR 센서 데이터 기반 3D 공간 구축

지도교수: 김원석

팀명: 카트라이다

201824616 홍주혁

202055640 남예진

202055647 우현우

## <목차>

### 1 요구조건 및 제약사항 분석

#### 1.1 요구조건

#### 1.2 제약사항 분석 및 수정 사항

##### 1.2.1 기존 제약사항

##### 1.2.2 추가 제약사항

### 2 설계 상세화 및 변경 내역

#### 2.1 데이터 수집 장치

##### 2.1.1 구현 기능

##### 2.1.2 사용 장비

##### 2.1.3 사용 개발 도구

#### 2.2 Rectangle 추출 프로그램

##### 2.2.1 구현 기능

#### 2.3 바닥, 벽, 장애물 구분 및 3차원 공간 구현

##### 2.3.1 구현 기능

#### 2.4 플로우 차트

##### 2.4.1 전체 플로우 차트

##### 2.4.2 단계별 플로우 차트

### 3 갱신된 과제 추진 계획

### 4 과제 진행 내용

### 5 보고 시점까지의 과제 수행 내용 및 중간 결과

#### 5.1 데이터 수집 장치

#### 5.2 Rectangle 추출 프로그램

##### 5.2.1 코드로 생성된 정육면체 포인트 클라우드 변환 결과

##### 5.2.2 프라이브루크 대학의 OctoMap scan 포인트 클라우드 변환 결과

#### 5.3 바닥, 벽, 장애물 구분 및 3차원 공간 구현

### 6 참고 자료

## 1 요구조건 및 제약 사항 분석

### 1.1 요구조건

LiDAR 센서를 이용하여 실내를 측정한 데이터셋을 기반으로 벽, 바닥 그리고 공간 상의 장애물을 3 차원 가상 환경에서 시각화

- LiDAR 센서를 활용하여 3D 포인트 클라우드 데이터 수집
- Raspberry Pi 를 활용하여 AWS RoboMaker 에 LiDAR 센서 측정값 전달 및 LiDAR 센서가 장착된 카트 바퀴 조작
- Wi-Fi 를 활용하여 Raspberry Pi 와 AWS RoboMaker 간 통신을 수행
- AWS RoboMaker 를 활용하여 PC 와 Raspberry Pi 를 연결하는 클라우드 ROS 환경 구축, ROS 를 통해 카트 조작 신호 전달 및 SLAM 적용
- Python 을 활용하여 원본 공간의 벽, 바닥, 장애물 데이터 추출
- Unity 를 활용하여 공간 시각화 및 화면 조작과 장애물 렌더링 여부 선택 기능을 제공하는 사용자 인터페이스 구축

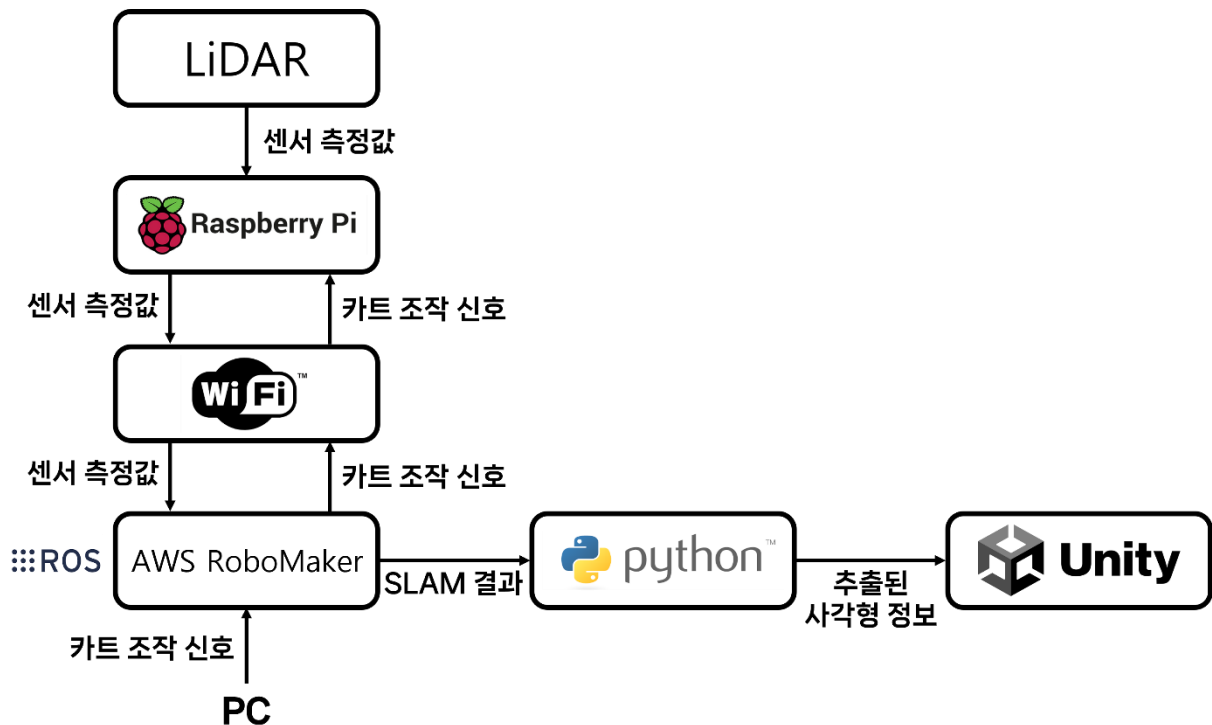


그림 1 전체 구상도

## 1.2 제약사항 분석 및 수정 사항

### 1.2.1 기존 제약사항

- 공간 구축에 필요한 충분한 정보 수집 과정이 필요
  - LiDAR 센서가 장착된 카트를 직접 조작하여 누락되는 공간이 없도록 충분한 정보 수집 진행
- 필요한 연산 과정을 모두 수행하기에 부족한 Raspberry Pi의 성능
  - Raspberry Pi는 LiDAR 센서 데이터 전송 및 카트 조작 신호를 받아 카트 바퀴를 움직이는 작업만 수행
- 비교적 저가의 LiDAR 센서 사용에 따른 오차
  - 테스트 공간 제작 후 수집한 데이터로 오차 제거 알고리즘 구현 예정
- 움직이는 물체가 LiDAR 센서에 스캔되는 경우 발생하는 오차
  - 움직이는 물체가 존재하지 않는 공간을 제작 후, 데이터 수집은 해당 공간에서만 진행
  - 데이터 수집 과정에서 움직이는 물체가 LiDAR 센서 앞을 지나가지 않도록 주의

### 1.2.2 추가 제약사항

- 직육면체 외의 물체가 있는 공간의 경우 벽, 장애물, 바닥의 구분이 어려움
  - 공간의 벽, 바닥, 장애물을 모두 3차원 직육면체로 제작
  - 공간의 벽, 바닥, 장애물의 위치는 차량의 이동만으로 충분히 스캔 가능하도록 구성
- 2개 이상의 근접한 장애물이 나란히 놓여있는 경우 장애물 분리가 어려움
  - 장애물의 간격을 LiDAR 센서에서 다른 장애물로 인식할 정도로 이격하여 배치
  - 장애물 데이터를 분류할 때 면의 법선 벡터를 이용하여 하나의 장애물로 합쳐지지 않도록 구현

## 2 설계 상세화 및 변경 내역

### 2.1 데이터 수집 장치

#### 2.1.1 구현 기능

- SLAM 적용 완료
- ROS msg 를 통해 LiDAR 센서 위치 정보를 이동시키는 알고리즘 구현 완료
- Raspberry Pi 를 통해 조작이 가능한 카트 제작 예정

#### 2.1.2 사용 장비

- LiDAR
  - Cygbot 사의 CygLiDAR 사용
  - 2D/3D 거리 데이터를 동시에 측정 가능
  - SLAM 에 필요한 3D 포인트 클라우드 데이터를 ToF 방식으로 수집
- Raspberry Pi
  - Raspberry Pi 3 Model A+ 사용
  - 2.5A micro USB 를 전원으로 사용하므로 보조배터리를 카트에 함께 장착하여 사용
  - 무선 LAN 을 지원하므로 Wi-Fi 통신을 통해 AWS RoboMaker 와 연결
  - LiDAR 센서에서 수집된 데이터를 AWS RoboMaker 에 전송
  - AWS RoboMaker 의 ROS 에서 카트 조작 신호를 받아 카트 바퀴 조작을 수행
- 카트
  - 디바이스마트의 아두이노 블루투스 스마트카 키트 [DV1801-KIT] 사용
  - 전진, 후진, 제자리 회전 가능
  - 카트 조립 후 Raspberry Pi, 보조배터리, LiDAR 센서를 장착 및 연결
  - Raspberry Pi 에서 카트의 바퀴 조작을 담당
  - 키트에서 주어진 캐스터로 인해 덜컹거리는 현상이 발생하면 볼 캐스터로 교체

### 2.1.3 사용 개발 도구

- ROS
  - 로봇 애플리케이션 구축에 사용하는 라이브러리 및 도구 모음
  - 장비나 프로그램에 부여된 노드 간의 데이터 통신 인터페이스를 정의하여 지원
  - CygLiDAR 나 OctoMap 모두 ROS 내 라이브러리 존재
  - 필요한 Python 코드를 작성하여 실행하고 새로운 노드를 만들어 데이터 통신 가능
- OctoMap
  - 로봇이 주변 환경을 인식하고 위치를 파악함과 동시에 지도를 작성하는 기술인 SLAM 의 한 종류
  - Octree 를 기반으로 작동하며, 센서 측정값을 추가하는데 제한이 없고 map 의 범위를 정할 필요 없이 동적으로 확대되는 장점
  - Cygbot 공식 유튜브에서도 OctoMap 을 사용한 SLAM 을 시연
  - ROS 의 PointCloud2.msg 를 받아 3D map 을 생성하고 이를 binary tree 나 octree 파일로 저장하는 기능 지원
  - binary tree 나 octree 파일을 포인트 클라우드 파일로 변환하는 기능 지원
- AWS RoboMaker
  - 인프라를 관리하지 않고도 로봇 개발 시뮬레이션을 실행, 크기 조정 및 자동화할 수 있는 클라우드 기반 시뮬레이션 서비스
  - Raspberry Pi 와 PC 를 직접 연결하지 않고도 클라우드 환경을 통해 데이터 전송 가능
  - ROS 를 지원하므로 마치 한 PC 에서 구현하듯이 설계가 가능
  - AWS free tier 를 통해 25 SU 시간동안 무료로 사용 가능

## 2.2 Rectangle 추출 프로그램

### 2.2.1 구현 기능

- RANSAC과 위치 기반 BFS 클러스터링을 통한 포인트 그룹화 기능 구현 완료
- 한 rectangle을 이루는 포인트 그룹을 rectangle로 변환 기능 구현 완료
- 한 변이 맞닿고 평면의 각도가 비슷한 rectangle 합병 기능 구현 완료

## 2.3 바닥, 벽, 장애물 구분 및 3차원 공간 구현

### 2.3.1 구현 기능

- Rectangle 점을 바탕으로 Convex Hull 알고리즘을 활용하여 벽과 장애물의 면을 구성하는 점을 구분하는 기능 구현 완료
- 벽을 이루는 점을 참고하여 바닥과 벽면을 3 차원 공간에 생성 완료
- 장애물 분류 및 3 차원 공간 생성 예정
- 원본 공간과 공간 상의 장애물을 따로 나타내는 기능이 있는 UI 구현 예정

## 2.4 플로우 차트

### 2.4.1 전체 플로우 차트

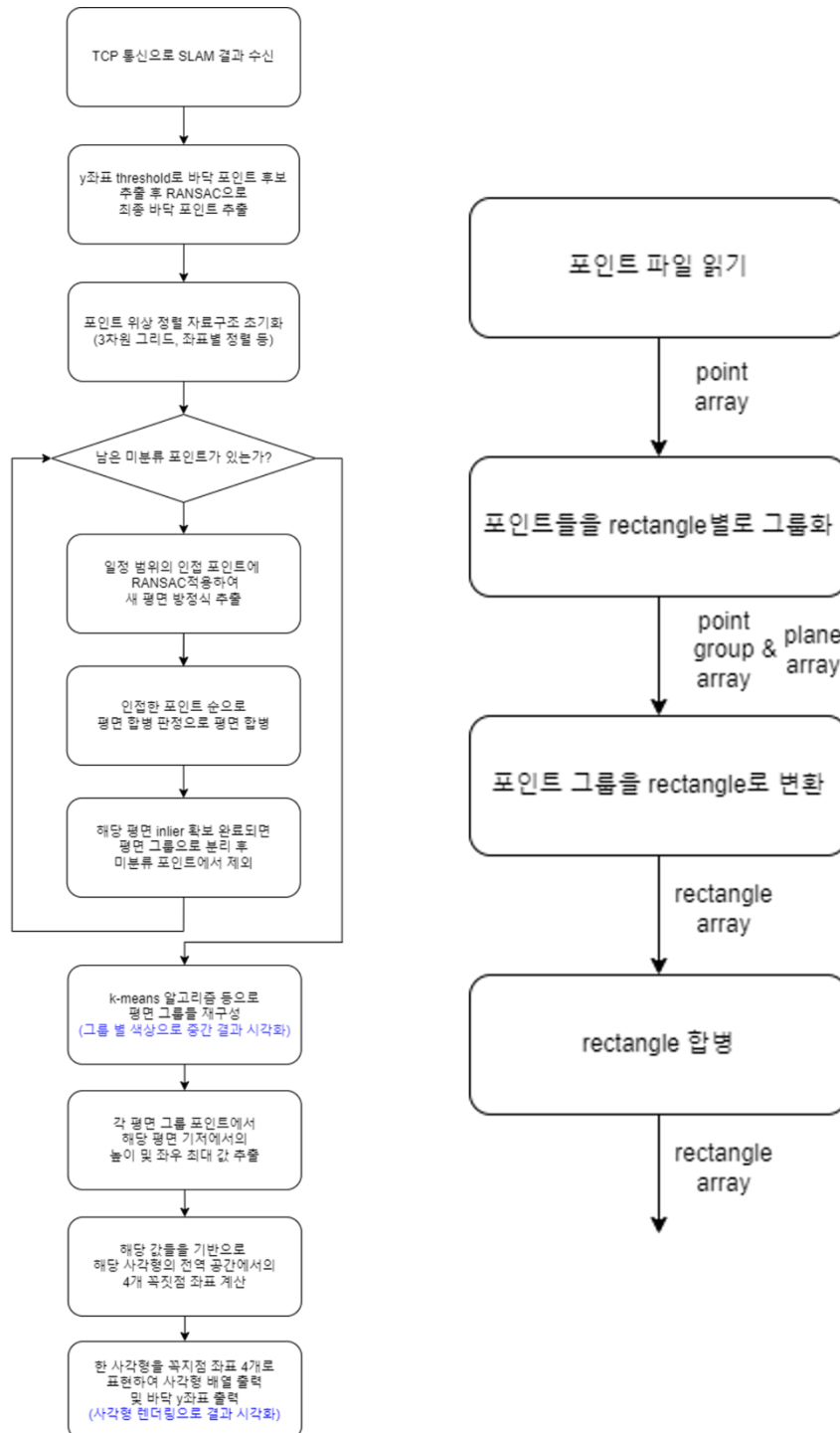


그림 2 (좌) 착수 보고서의 사각형 추출 알고리즘 플로우 차트

(우) 구현된 Rectangle 추출 프로그램 전체 플로우 차트



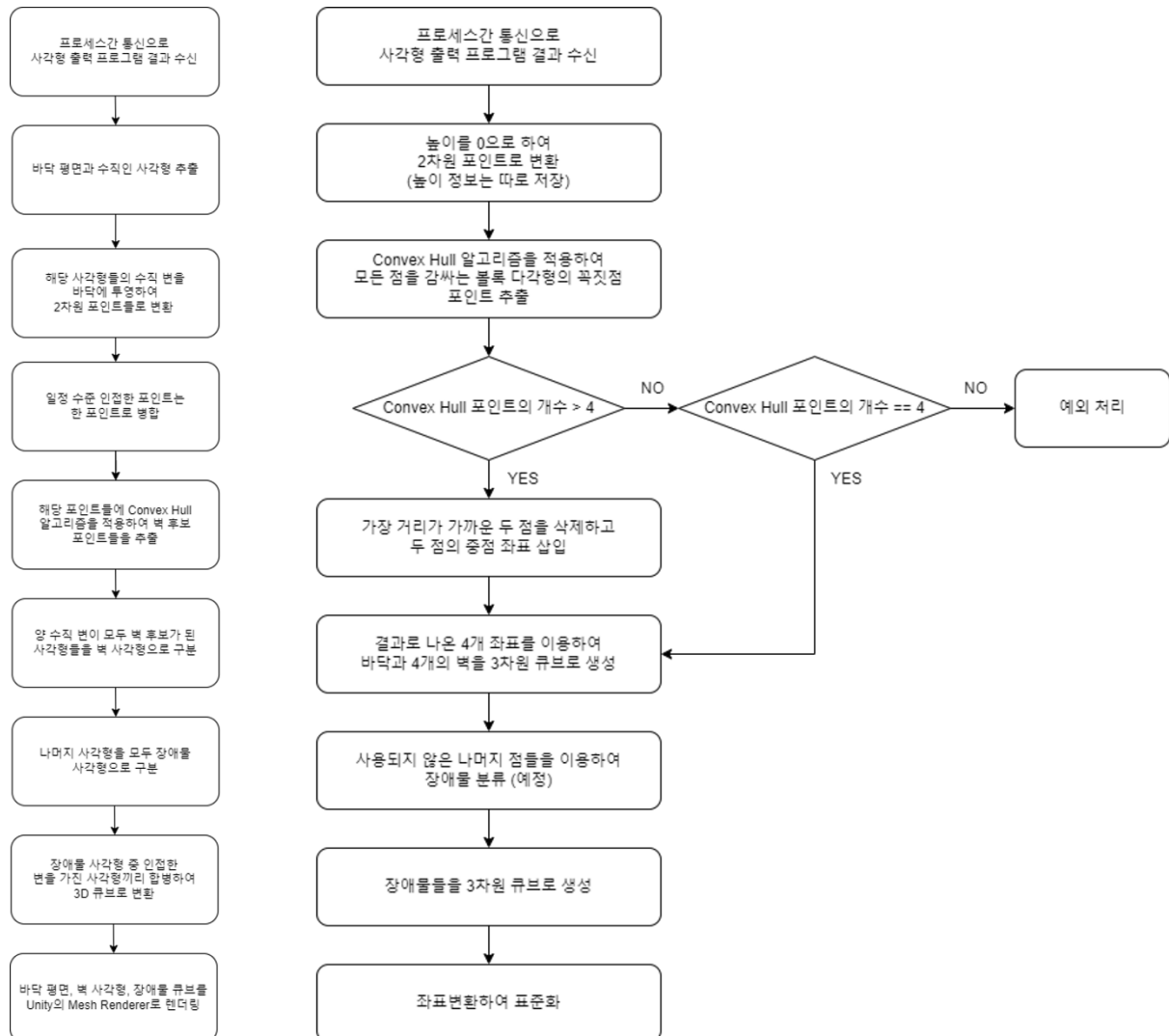


그림 3 (좌) 착수 보고서의 바닥, 벽, 장애물 구분 알고리즘 플로우 차트  
(우) 진행중인 벽, 장애물 구분 및 3차원 공간 구현에 대한 전체 플로우 차트

## 2.4.2 단계별 플로우 차트

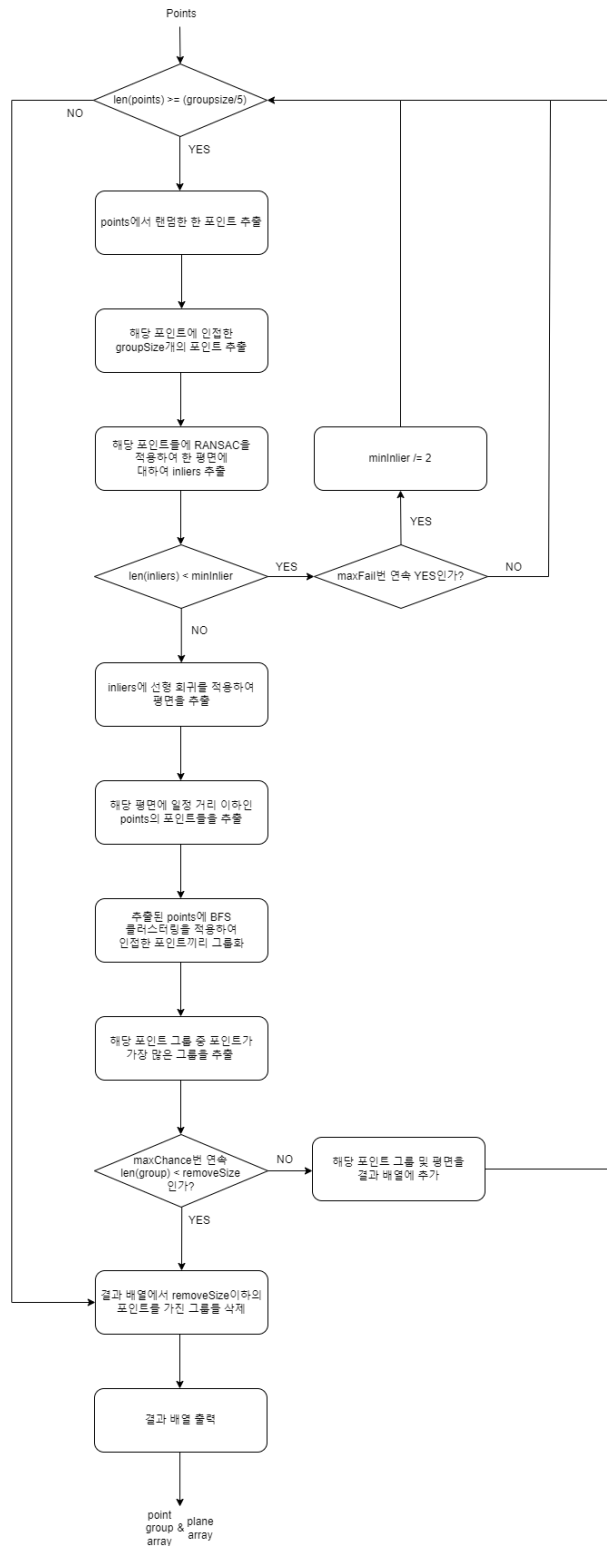


그림 4 포인트 그룹화 단계 플로우 차트

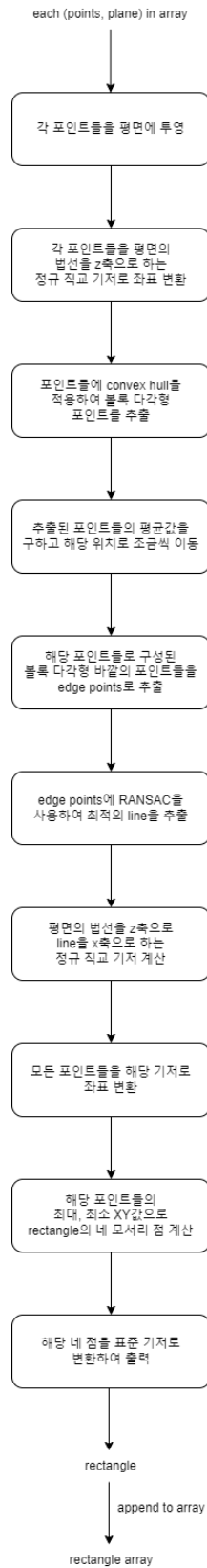


그림 5 Rectangle 변환 단계 플로우 차트

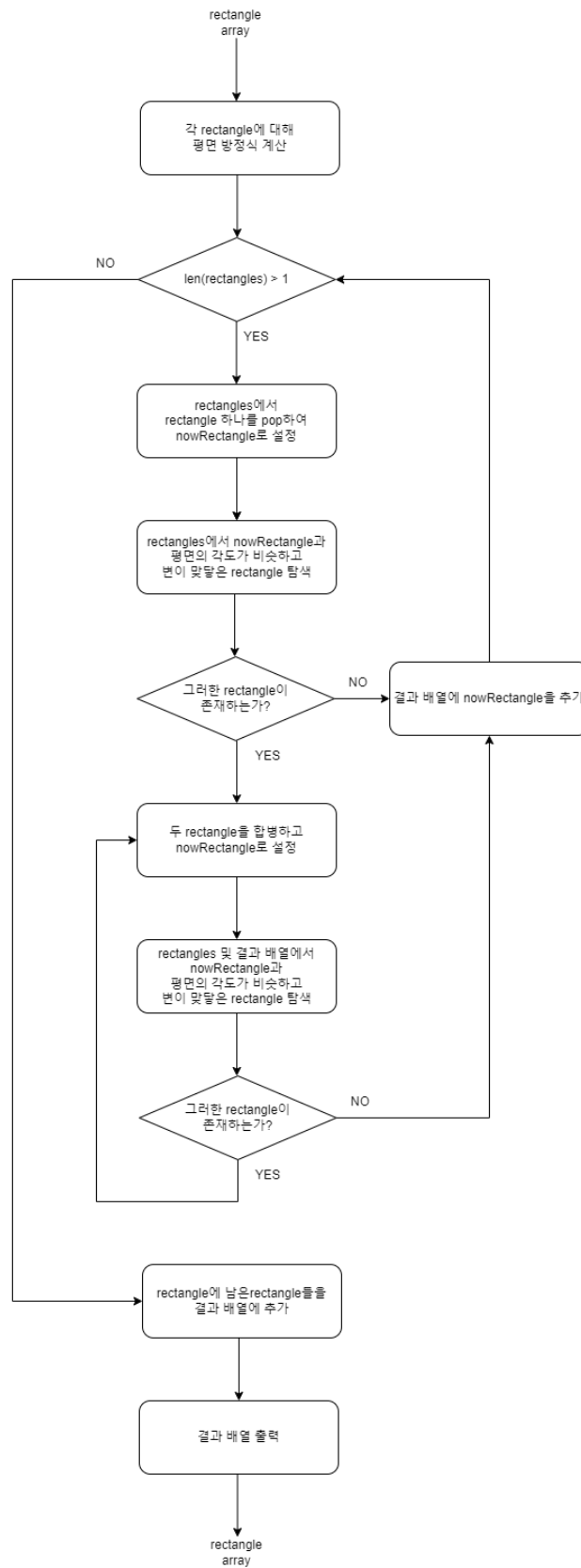


그림 6 Rectangle 합병 단계 플로우 차트

## 3 갱신된 과제 추진 계획

홍주혁

6월				7월				8월				9월			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ROS, SLAM 관련 기술 스터디															
			SLAM 구현												
				카트 조작 프로그램 개발											
						중간보고서 작성									
							카트 및 테스트 공간 제작								
								알고리즘 구현 지원							
										테스트 및 성능 평가					
												최종 보고서 작성 및 발표 준비			

표 1 과제 추진 계획 (홍주혁)

우현우

6월				7월				8월				9월			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
RANSAC, 계산 기하학 관련 기술 스터디															
				Rectangle 추출 프로그램 개발											
						중간보고서 작성									
									통신 시스템 구현						
										테스트 및 성능 평가					
												최종 보고서 작성 및 발표 준비			

표 2 과제 추진 계획 (우현우)

남예진

6월				7월				8월				9월			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
계산기하학, Unity 관련 기술 스터디															
				벽, 장애물 구분 알 고리즘 구현											
							Unity 프로그램 개발								
						중간보고서 작성									
										테스트 및 성능 평가					
												최종 보고서 작성 및 발표 준비			

표 3 과제 추진 계획 (남예진)

#### 4 과제 진행 내용

- 홍주혁
  - SLAM 적용 완료
  - 카트 조작 프로그램 개발 진행 중
- 우현우
  - Rectangle 추출 프로그램 1 차 구현 완료
  - Rectangle 추출 프로그램 최적화 및 정확도 향상 진행 중
- 남예진
  - 바닥, 벽 3 차원 공간 구현 완료
  - 장애물 3 차원 공간 구현 진행 중

## 5 보고 시점까지의 과제 수행 내용 및 중간 결과

### 5.1 데이터 수집 장치

- SLAM (OctoMap)

※ 카트 배송이 완료되지 않아 LiDAR 센서를 한 위치에 고정시킨 채로 공간 데이터를 수집



그림 2 LiDAR 센서 데이터 수집 공간 (문틀과 벽)

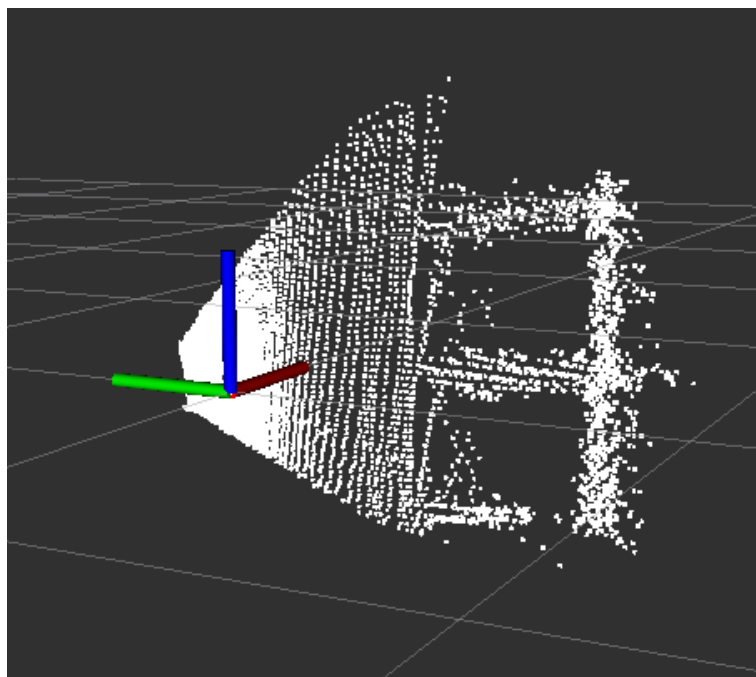


그림 3 LiDAR 센서 데이터 수집 결과

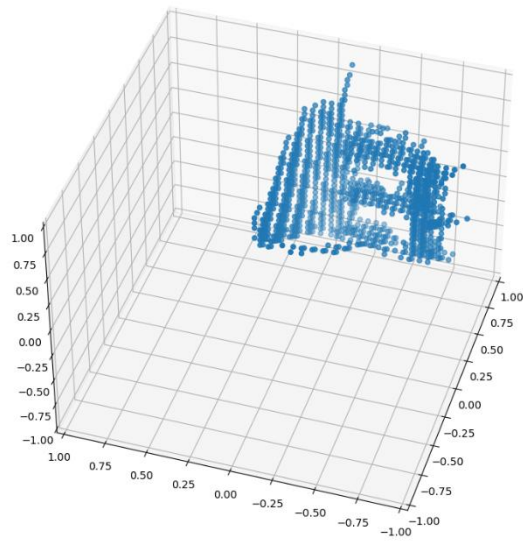


그림 4 SLAM 후 포인트 클라우드 변환 결과를 Python으로 시각화



## 5.2 Rectangle 추출 프로그램

### 5.2.1 코드로 생성된 정육면체 포인트 클라우드 변환 결과

- 파일 읽기 단계 결과

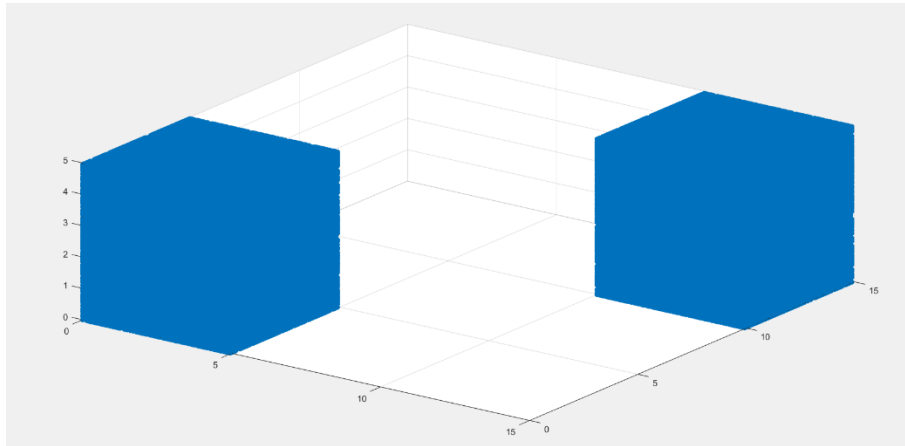


그림 5 파일 읽기 단계 결과

- 포인트 그룹화 단계 결과

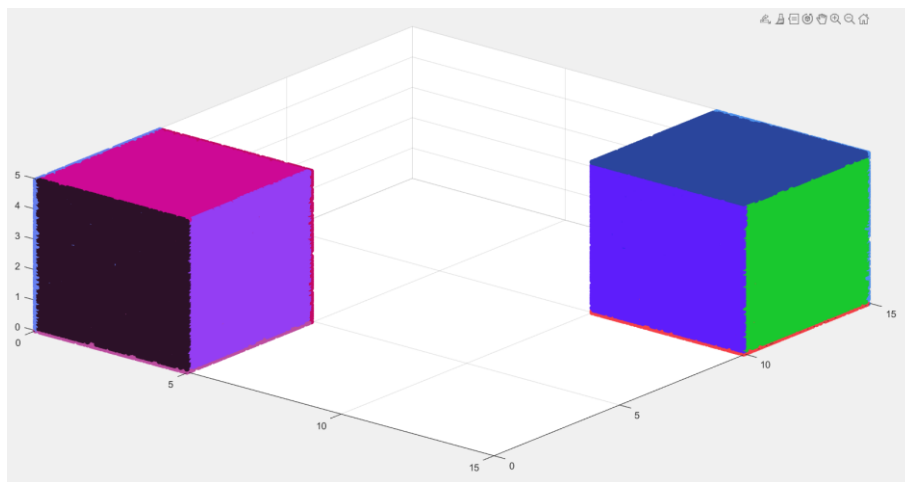


그림 6 포인트 그룹화 단계 결과

- Rectangle 변환 단계 결과

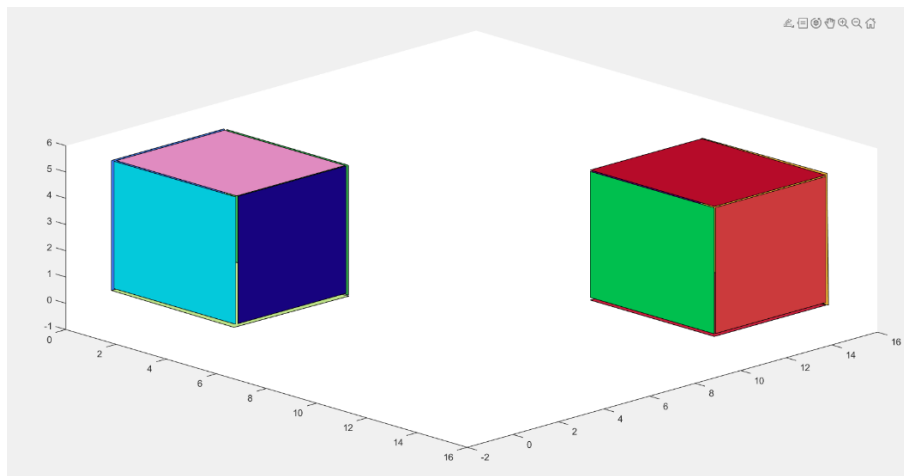


그림 7 Rectangle 변환 단계 결과

- Rectangle 합병 단계 결과

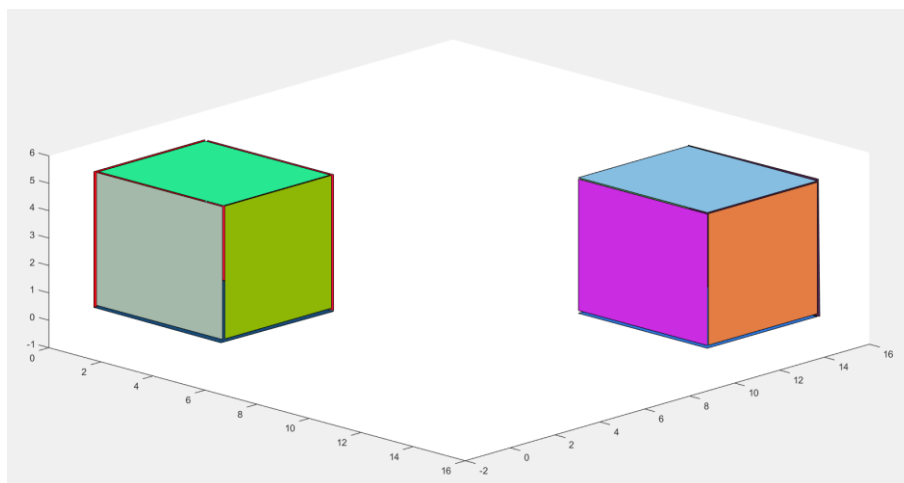


그림 8 Rectangle 합병 단계 결과

### 5.2.2 프라이브루크 대학의 OctoMap scan 포인트 클라우드 변환 결과

- 파일 읽기 단계 결과

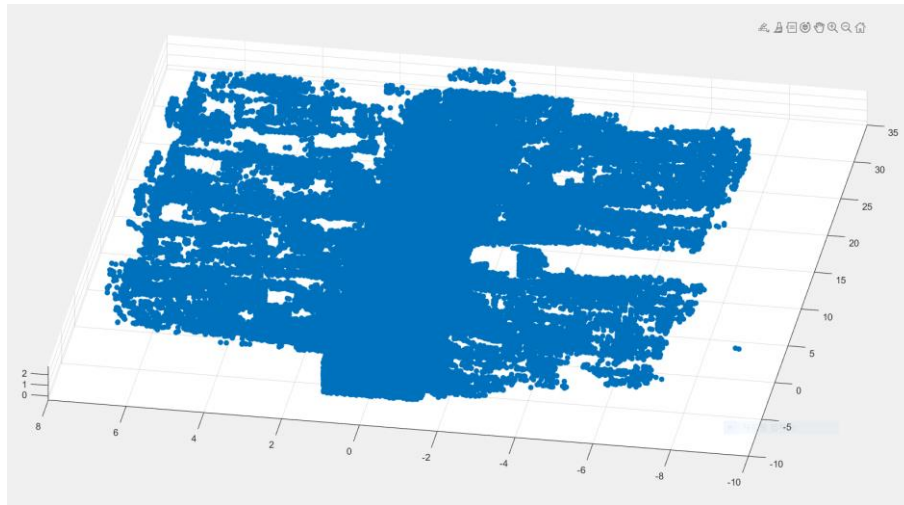


그림 9 파일 읽기 단계 결과

- 포인트 그룹화 단계 결과

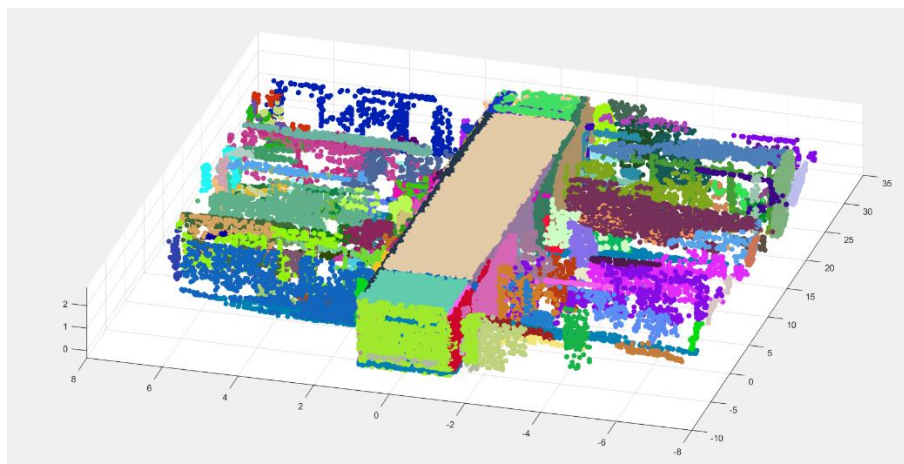


그림 10 포인트 그룹화 단계 결과

- Rectangle 변환 단계 결과

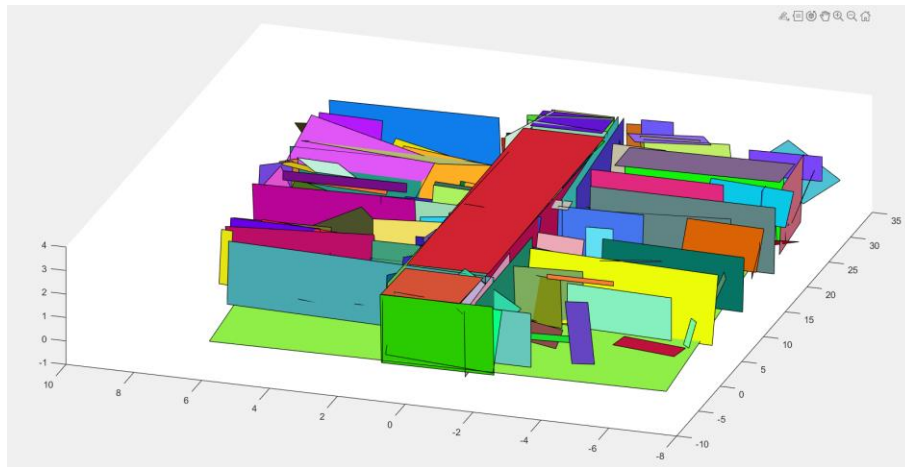


그림 11 Rectangle 변환 단계 결과

- Rectangle 합병 단계 결과

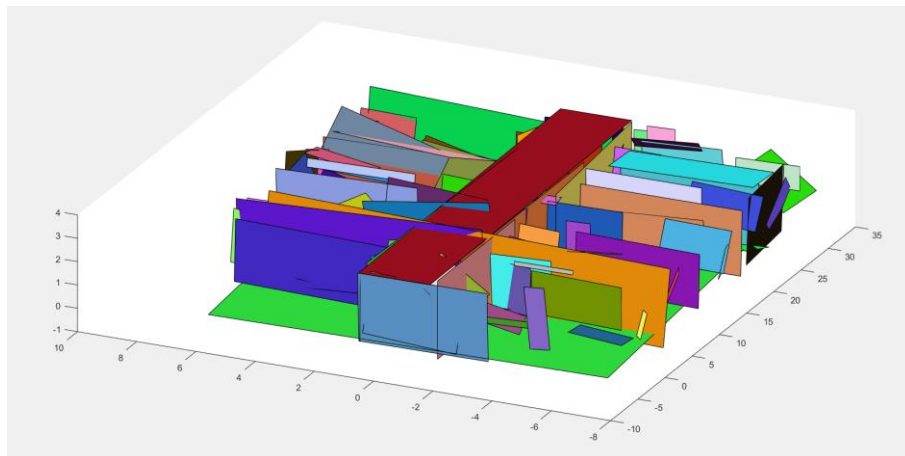


그림 12 Rectangle 합병 단계 결과

### 5.3 바닥, 벽, 장애물 구분 및 3 차원 공간 구현

- 입력되는 Rectangle 점 데이터 형식
  - 텍스트 파일의 첫 줄은 바닥 면의 높이를 나타냄
  - 텍스트 파일의 둘째 줄부터 하나의 면을 이루는 3 차원 상의 4 개 정점이 콤마로 구분되어 나열 ( $x_0, y_0, z_0, x_1, y_1, z_1, \dots, z_3$ )

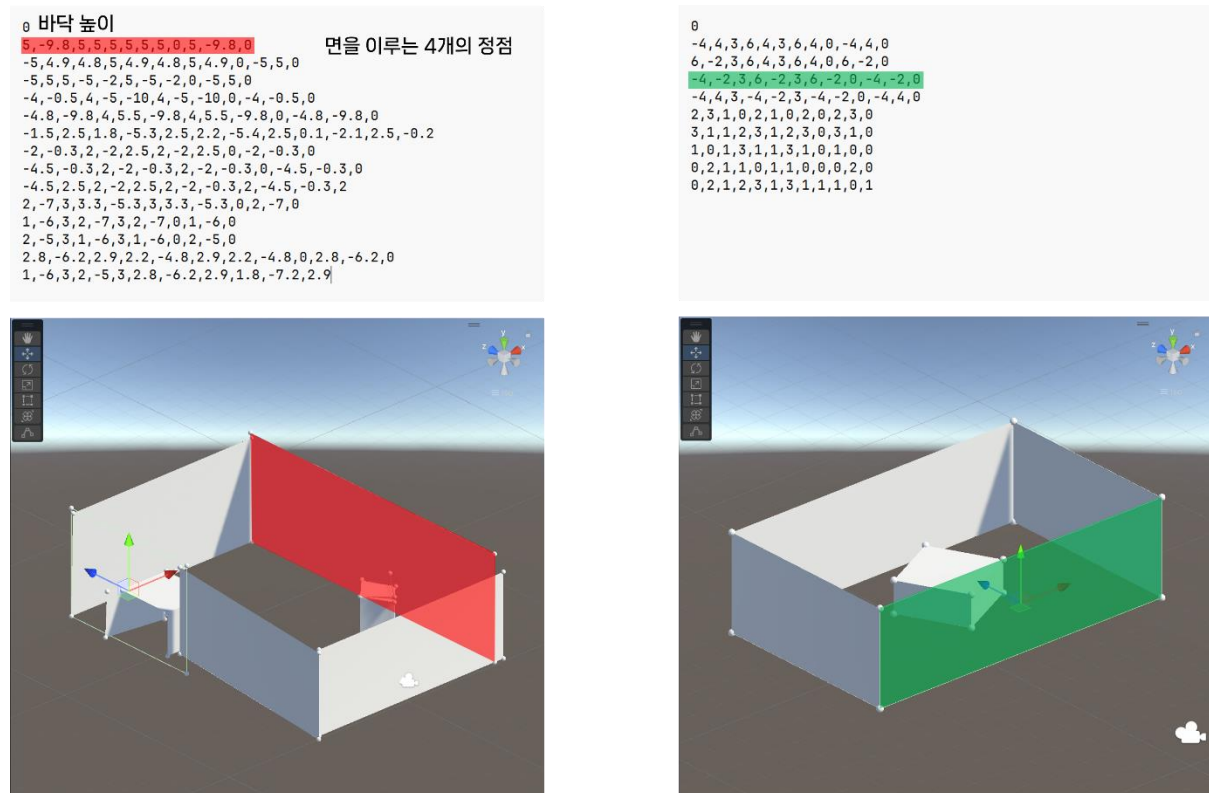


그림 18 (위) 사용할 텍스트 파일 형식 예시

(아래) 예시 데이터를 3 차원 공간에 표현한 그림 (시야의 편의를 위해 장애물을 가리는 면의 mesh renderer 를 off 한 상태)

- 3 차원의 Rectangle 정점을 2 차원으로 변환

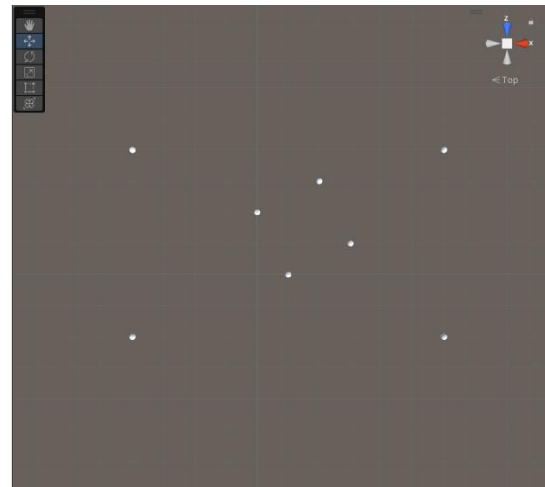
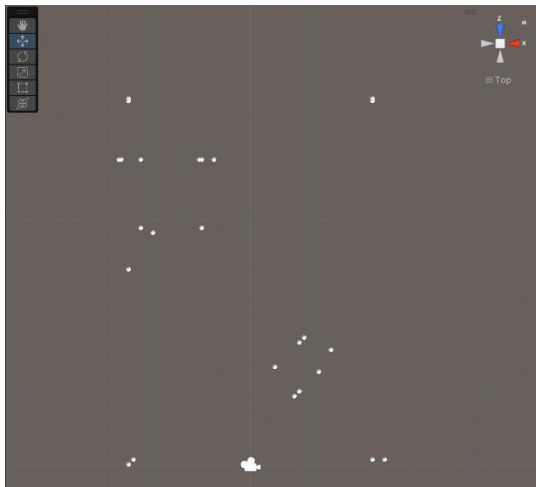


그림 19 2 차원으로 변환한 Rectangle 정점

- 2 차원 변환한 점을 바탕으로 Convex Hull 을 활용하여 벽을 이루는 점 추출



그림 20 파란색 구는 Convex Hull 로 찾아낸 볼록 다각형을 이루는 정점들로부터 구한 4 개의 점을 나타냄

- 벽을 이루는 4 개의 점을 이용하여 3 차원 공간에 벽과 바닥 구현

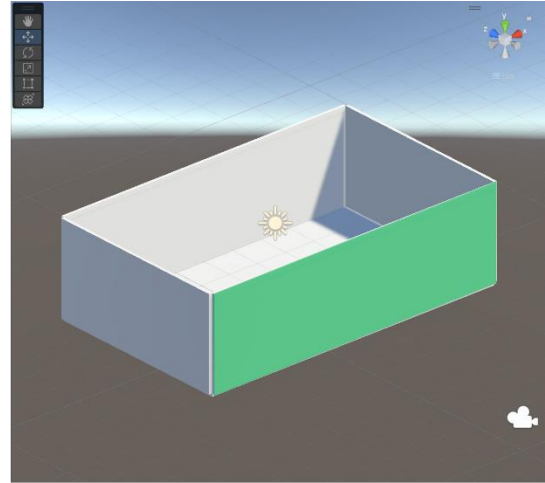
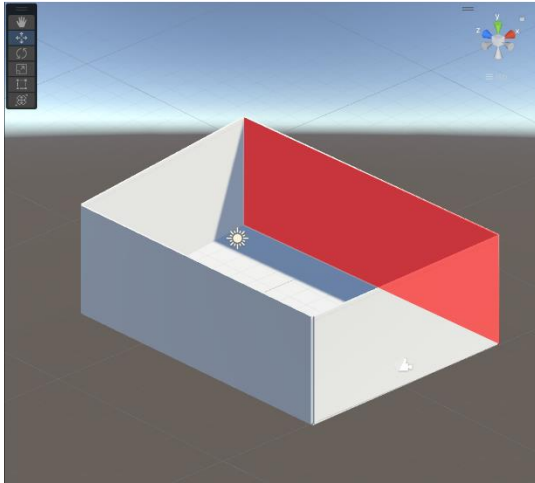


그림 21 [그림 20]의 4 개 점을 이용하여 벽과 바닥 생성

## 6 참고 자료

[1] 독일 프라이브루크 대학 OctoMap 3D scan dataset : FR-079 corridor (5.2.2)

<http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/>