



국민대학교  
전자정보통신대학  
컴퓨터공학부


# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	OLAF
팀 명	ELSA (Elaborate Localization System Architects)
문서 제목	수행계획서

Version	1.9
Date	2020-MAR-20

팀 원	김 다 훈 (조장)
	김 명 수
	김 선 필
	배 한 울
	윤 찬 우

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 “OLAF”를 수행하는 팀 “ELSA”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “ELSA”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


Filename	수행계획서-OLAF.doc
원안작성자	김선필, 김다훈, 김명수, 배한울, 윤찬우
수정작업자	김선필, 김다훈, 김명수, 배한울, 윤찬우

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-03-15	김선필	1.0	최초 작성	프로젝트 개요, 연구/개발 내용
2020-03-16	윤찬우	1.1	내용 추가	연구/개발 내용 IMU센서 추가
2020-03-18	배한울	1.2	내용 추가	프로젝트 추진 배경 및 필요성 추가
2020-03-19	김다훈	1.3	내용 추가	시스템 기능 요구사항 추가, 배경 기술 추가
2020-03-21	김명수	1.4	내용 추가	시스템 구조 추가, 프로젝트 비용 추가
2020-03-22	김선필	1.5	내용 추가	개발 일정, 산출물 추가
2020-03-23	배한울	1.6	내용 수정	시스템 구조, 비용 수정
2020-03-24	김다훈	1.7	내용 추가	인력자원 추가
2020-03-25	윤찬우	1.8	내용 수정	개발 일정, 참고문헌 수정
2020-03-26	김다훈	1.9	내용 수정	전체 서식 통일


 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 목 차

<b>1. 개요</b>	<b>5</b>
1.1 프로젝트 개요	5
1.2 추진 배경 및 필요성	6
1.2.1 기술의 시장현황	6
1.2.2 개발할 기술의 필요성	8
<b>2. 개발 목표 및 내용</b>	<b>9</b>
2.1 목표	9
2.2 연구/개발 내용	10
2.2.1 하드웨어 제작	10
2.2.1.1 플랫폼 선택	10
2.2.1.2 3D 모델링 및 프린팅	11
2.2.1.3 하드웨어 설계	13
2.2.2 ROS(Robot Operating System) 개발 환경 구축	17
2.2.3 Sensors	18
2.2.3.1 LiDAR(Light Imaging Detection and Ranging)	18
2.2.3.2 OPENCN(Open-source Control Module for ROS) + DYNAMIXEL	19
2.2.3.3 IMU	20
2.2.3.4 CAMERA	21
2.2.4 SLAM(Simultaneous Localization And Mapping)	22
2.2.5 OBJECT DETECTION YOLO	24
2.2.6 Data and Network	25
2.2.6.1 국민대학교 미래관(구 7호관) 데이터셋 구축	25
2.2.6.2 네트워크	26
2.3 개발 결과	28
2.3.1 시스템 기능 요구사항	28
2.3.2 시스템 비기능(품질) 요구사항	28
2.3.2.1 신뢰할 수 있는 실시간 위치추정	28
2.3.2.2 안정적인 서버 통신	29
2.3.2.3 신속한 돌발상황 대처를 위한 Object Detection 알고리즘	29
2.3.3 시스템 구조	30
2.3.3.1 Localization	31
2.3.3.2 SLAM	31
2.3.3.3 Navigation	31
2.3.4 결과물 목록 및 상세 사양	32

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

2.4 기대효과 및 활용방안	33
<b>3. 배경기술</b>	<b>34</b>
3.1 기술적 요구사항	34
3.1.1 개발 환경	34
3.1.2 프로젝트 결과물 확인 환경	34
3.2 현실적 제한 요소 및 그 해결 방안	34
3.2.1 하드웨어 제한 요소	34
3.2.1.1 바퀴	34
3.2.1.2 LiDAR	34
3.2.1.3 로봇의 견고함	35
3.2.1.4 카메라 각도	35
3.2.2 소프트웨어 제한 요소	35
3.2.2.1 Encoder, IMU, Camera, LiDAR 센서들을 이용한 위치추정	35
3.2.3 기타 제한 요소	36
3.2.3.1 웹/앱을 이용한 통신	36
3.2.3.2 다른 층으로의 안내	36
<b>4. 프로젝트 팀 구성 및 역할 분담</b>	<b>37</b>
<b>5. 프로젝트 비용</b>	<b>38</b>
<b>6. 개발 일정 및 자원 관리</b>	<b>39</b>
6.1 개발 일정	39
6.2 일정별 주요 산출물	40
6.3 인력자원 투입계획	42
6.4 비 인적자원 투입계획	43
<b>7. 참고 문헌</b>	<b>44</b>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 1. 개요

### 1.1 프로젝트 개요

매년 들어오는 신입생 및 외부인들은 학과 건물과 같은 대규모 실내 공간에서는 종종 길을 헤매기도 한다. 하지만 GPS가 통하지 않기 때문에 스마트폰도 별다른 도움이 되지 않는 경우가 많다. 마침 실내지도가 있더라도, 지금 내 위치가 어디인지부터가 문제이다. 특히 국민대학교 미래관(구 7호관)의 경우 4층을 기준으로 구관, 신관이 나누어져 있기때문에 처음 방문하는 경우 상당히 혼란스럽다.


OLAF 는 이러한 문제점을 해결하고, 교내 학생 및 교내를 방문하는 외부인에게 도움을 주기 위해서 고안된 프로젝트이다. 국민대학교 각 건물만의 특화된 안내 시스템을 구축하는 것이 이 프로젝트의 목표이다.

주변 환경을 인지하면서 주행하는 실내 이동로봇은 현재 다양한 분야에서 활용되고 있다. 병원 건물 내에서의 물류 이동이나 휠체어 환자 이동, 보안 지키미 로봇, SAVIOKE 호텔 서빙 로봇, Naver의 Around M1 서점 도우미 로봇 등이 활용 되고 있다. 위의 사례들을 보며 OLAF를 통해 안내 시스템이 구축이 된 후 여러가지 분야에 응용을 하여 시스템의 활용 방안을 넓혀 나갈 계획이 있다.

실내 이동로봇은 실내 측위와 주변 환경에 대한 인지를 해야하는데 실내 측위에 대한 여러가지 기술들이 있다. 이러한 측위 기술을 이용한 사례로서 StarGazer 로봇용 측위 실내 센서를 사용하여 천장의 landmark로 사용해서 측위하거나, 실내 GPS나 Wifi, Beacon 등을 이용하여 삼각 측량으로 측위한다.

실내 이동로봇의 경우 가장 필수적인 SLAM(Simultaneous Localization And Mapping)은 동시 측위하면서 지도를 측량하는 기술이다. 이를 위해 특정 패턴으로 빛을 방사해서 되돌아 온 시간으로 주변 장애물의 거리를 측정하는 LiDAR(Light Detection And Ranging) 센서와 IMU 센서를 이용한 측위를 통해 지도를 작성하게 된다. 측위의 불확실성을 극복하기 위해 칼만 필터, 파티클 필터를 활용한 Montecarlo 측위 등의 신호처리 방식을 사용한다. 지도 작성 알고리즘은 파티클 필터에 보정된 격자지도(PGM, Particle Filtered Grid Map)가 있고, 가장 많이 사용되는 것이 점유격자지도(OGM, Occupancy Grid Map)이다.

지도 정보를 바탕으로 출발지와 목적지까지의 경로설정 및 자율주행은 가장 초기에 고려된 Dijkstra의 최단 거리 visibility graph, A\* 알고리즘, D\* 알고리즘, RRT(Rapidly-exploring Random Trees), DWA(Dynamic Window Approach), Potential Field 등의 다양한 알고리즘에 따라 할 수 있다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

로봇 개발을 위해 제공되는 개발환경으로는 ROS(Robot Operating System)를 사용한다. ROS는 로봇을 구성하는 다양한 센서 탐지 및 구동부 제어를 위한 프로그램 패키지 등을 제공한다. 복잡한 로봇 동작들을 위해 다중 프로세스들을 관리하고 연동할 수 있도록 함으로써 여러 개의 로봇 간에 협업도 가능하다. TF 라이브러리를 이용해 여러 좌표 프레임 간 실시간 좌표변환 데이터를 제공함으로써 자세제어를 용이하게 한다.


본 프로젝트에서는 ROBOTIS사의 Turtlebot3 플랫폼을 사용하고 Nvidia사의 Jetson TX2 보드를 주 제어 컴퓨터로 하여, 여기에 LINUX Ubuntu 16.04 버전 운영체제 상에 ROS Kinetic 버전의 메타운영체제를 설치해서 국민대학교 7호관 내에서 사용자가 목적지를 선택하면 해당 목적지로 사용자를 안내해주는 네비게이션 로봇을 구현한다. 하드웨어 제작시 외부 프레임의 경우 일일히 구매를 할시 시간적 비용적인 측면과 OLAF 만의 커스터마이징을 하기에 제약 조건이 많이 생기기 때문에 직접 3D 모델링 및 프린팅을 통해 제작한다.

## 1.2 추진 배경 및 필요성

매년 수천명의 신입생이 입학하고 신입생 뿐만 아니라 학교 외부 관계자, 배달원, 학부모 등 학교에 처음 방문 하는 사람들이 있다. 학교에 방문을 하면 가고자 하는 건물까지는 핸드폰, 학교내부 지도 등을 이용하여 쉽게 찾아 갈 수 있지만 건물 안에서는 가고자 하는 강의실, 장소까지 쉽게 찾아 갈 수 없다. 우리는 이러한 사람들을 위해 건물 내부에서 원하는 목적지까지 안내해주는 로봇을 만들어 보다 쉽고 빠르게 안내해 주는 서비스를 제공하려고 한다.

### 1.2.1 기술의 시장현황

현재 건물 내부에서 목적지까지 안내 해 주는 서비스로 가장 유명한 것이 바로 인천공항에 있는 ‘에어스타’[그림 1] 이다. 에어스타는 탑승권을 스캔하면 체크인 위치, 탑승구 위치를 알려주고 직접 안내해 준다. 에어스타 뿐만 아니라 현재 국내 안내로봇 서비스로는 SAVIOKE 호텔 서빙 로봇, Naver의 Around M1 서점 도우미 로봇 [그림 2], 원익로보틱스의 ‘애디(ADDY)’ 등이 있다. 이러한 추세에 따라 정부는 실내안내로봇 한국산업표준(KS) 규격을 제정하는 방안을 검토하고 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


KS규격이 제정되면 인증을 만들기 위한 논의도 이어질 것으로 예상된다. 실내안내로봇은 비교적 상용화가 활발한 분야이기 때문에 KS인증이 만들어지면 공공기관 등을 중심으로 시장 수요 형성에 기여할 전망이다.



[그림 1] 에어스타



[그림 2] Around M1

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


## 1.2.2 개발할 기술의 필요성

현재 국민대학교 홈페이지에는 캠퍼스 안내란 [그림 3]에 건물의 위치, 건물의 이름 등 나와 있어서 자기 현재 위치만 알면 캠퍼스 내 건물을 찾으러 가는 것은 쉽게 할 수 있다. 하지만 학교에 처음 방문하는 사람이 건물 내부에서 특정 강의실, 강당, 회의실 같은 곳을 가려고 하면 건물 안에서 길을 찾지 못하는 경우가 생긴다. 따라서 우리는 이를 해결하기 위해 캠퍼스 내 건물별로 안내로봇 서비스를 제공하려고 한다. 안내로봇을 통해 현재 건물 내에서 자기 위치가 어디인지, 목적지가 어디 있는지 알 수 있고 안내 로봇이 직접 목적지까지 안내한다.



[그림 3] 캠퍼스 안내 지도



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2. 개발 목표 및 내용

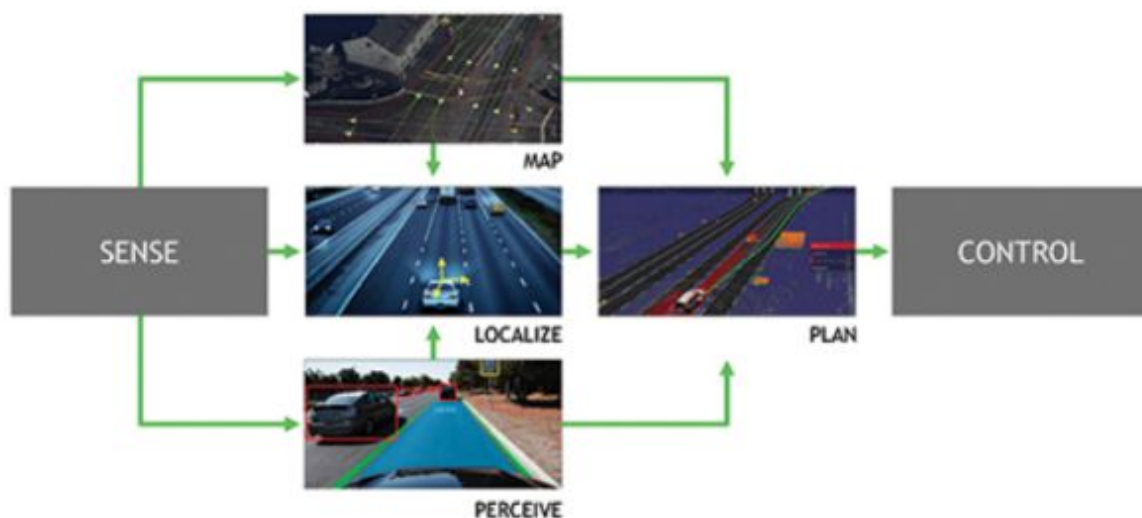
### 2.1 목표

OLAF는 교내 학생 및 교내를 방문하는 외부인에게 도움을 주기 위해서 고안된 프로젝트로서 국민대학교 각 건물만의 특화된 안내 시스템을 구축하는 것이 이 프로젝트의 목적이다.


학과 건물과 같은 대규모 실내 공간에서 매년 들어오는 신입생 및 외부인들과 같은 초행자들에게 OLAF의 에스코트와 최단경로 안내를 해주는 서비스를 제공하여 사람들은 보다 쉽게 길을 찾을 수 있다. 이를 통해 OLAF는 노동력과 사람들에게 삶의 편의를 제공할 수 있다.

교내 네비게이션이 구축된 후, 더 나아가 누구나 스마트 폰으로 촬영만으로 가상의 3차원 실내공간과 실내공간에 적용될 게임, 콘텐츠 제작에도 획기적인 발전이나 새로운 시도가 가능하다. 부가적으로, 실내안의 복지시설 광고나 정보 제공의 새로운 플랫폼으로도 확장 가능하다.

기술적인 측면으로는 SLAM으로 생성된 지도와 IMU, Camera를 통한 정확한 실시간 위치추정, LiDAR를 통한 장애물 인식을 통해 보다 빠른 처리 속도의 최단경로를 계산하는 알고리즘 구현을 목표로 한다.



[그림 4] 자율주행 파이프라인

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

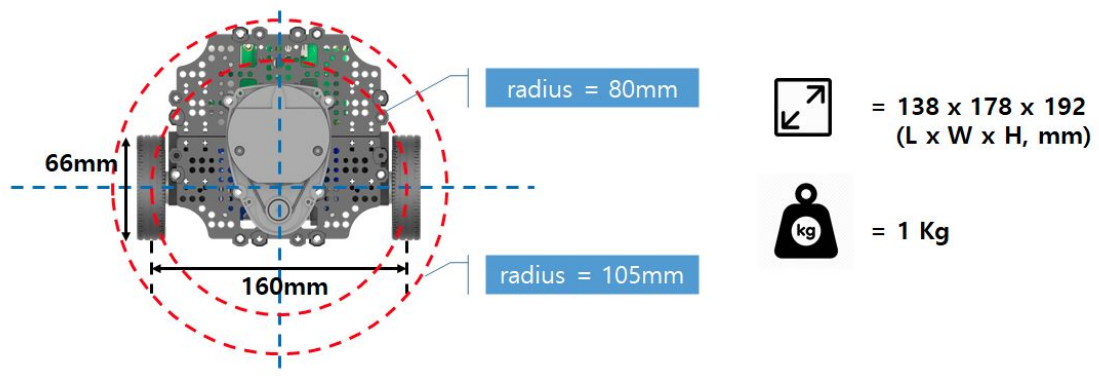
## 2.2 연구/개발 내용

OLAF 프로젝트의 연구 / 개발은 크게 N 부분으로 분리하면 다음과 같다. 먼저, 7호관 내에서 큰 제약 조건없이 돌아다닐 수 있는 플랫폼을 설계 및 제작하는 과정이 필요하다. 플랫폼 제작이 완료되면, 7호관 전체를 돌아다니며 실내 측위 및 주변 환경에 대한 인지를 가능하게 한다. 실내 측위의 경우 2D LiDAR와 IMU를 이용한 SLAM(Simultaneous Localization And Mapping) 기술을 사용하여 국민대학교 7호관의 지도를 구축하는 과정을 거치게 된다. 하지만 구축된 2D 지도만을 가지고 현재 로봇의 위치를 추정하기에는 로봇이 계속 이동하는 상황에서는 실시간으로 발생하는 미세한 오차가 있으므로, 각 교실 문에 붙어있는 방번호판을 인식함으로써 로봇의 현재 위치를 보정한다. 연구 / 개발에 관한 상세한 내용은 다음과 같다.


### 2.2.1 하드웨어 제작

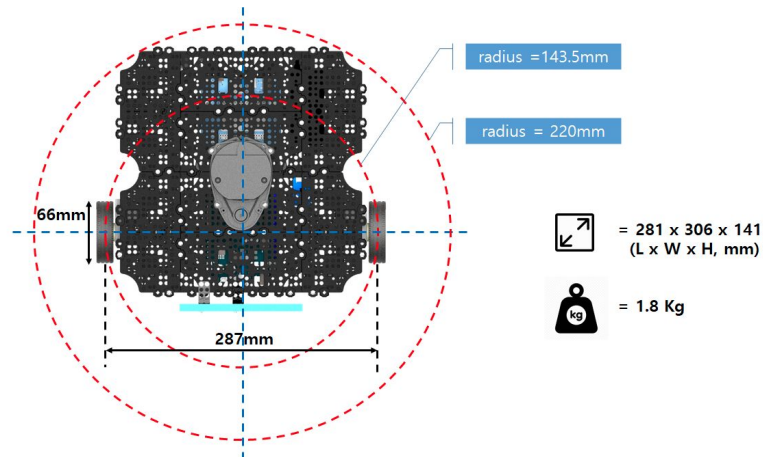
#### 2.2.1.1 플랫폼 선택

OLAF 프로젝트의 출발은 하드웨어 설계 및 제작이다. 비교적 학생들이 접근 할 수 있는 차량 플랫폼 중 XYTRON사의 XYCAR-A2, ROBOTIS사의 TURTLEBOT3 두개의 플랫폼 중 7호관 내부에서 안전하면서 효율적으로 주행할 수 있는 모델을 비교한다. 저렴한 가격, 높은 확장성, 모듈형 구조 및 센서 특화된 부분에 높은 점수를 받은 ROBOTIS사의 TURTLEBOT3을 주 하드웨어 플랫폼으로 정했다. 그러나 기존에 보유하고 있던 [그림 5] TURTLEBOT3 BURGER 모델의 경우 각 센서 및 보드를 부착하기에는 너무 작기 때문에 새로운 설계가 필요하다.



[그림 5] TURTLEBOT3 BURGER 스펙

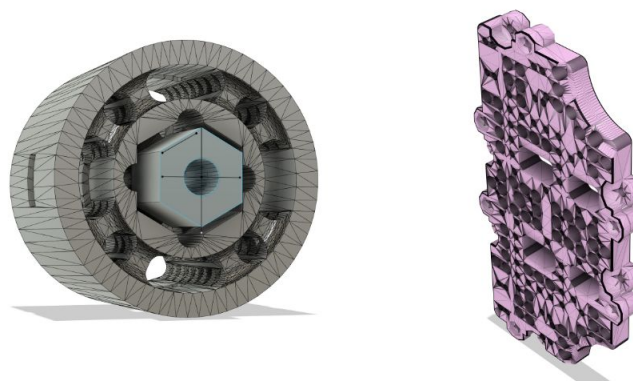
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20




[그림 6] TURTLEBOT3 WAFFLE 스펙

### 2.2.1.2 3D 모델링 및 프린팅

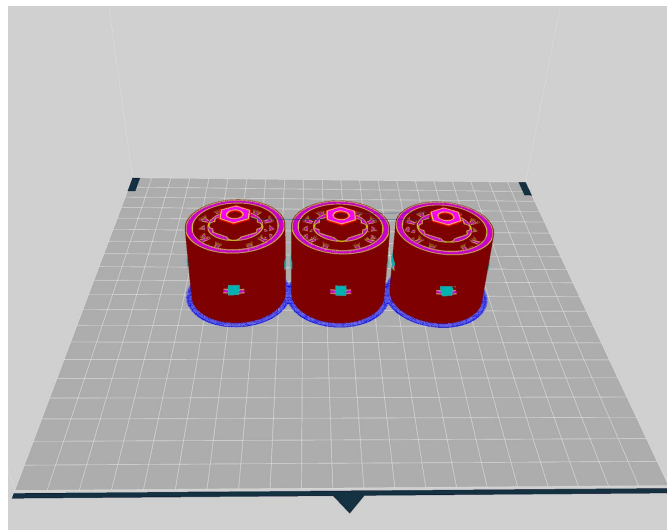
TURTLEBOT3의 강력한 장점중의 하나인 모듈형 구조를 이용하여 우리는 플랫폼의 크기를 키우기 위해 3D 모델링을 진행하여 외부 프레임을 제작한다. [그림 6]의 TURTLEBOT3 WAFFLE 구조를 참고하여 하드웨어 설계 및 부품 모델링을 진행한다. 자체적으로 모델링한 부분은 센서를 지지해줄 수 있는 Plate 및 몬스터 바퀴와 모터를 연결해줄 Motor-Wheel Joint 부분을 모델링 및 프린팅을 진행한다.



[그림 7] Motor-Wheel Joint & Turtlebot3 Plate 모델링


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

Fusion 360 프로그램을 이용하여 3D 모델링을 진행한다. Fusion 360은 오토데스크 에서 출시하고 있는 제품 디자인에 특화된 3D CAD/CAM/CAE 소프트웨어이다. 초심자도 쉽게 사용할 수 있는 프로그램으로 스케치, 바디, 구성요소 및 어셈블리가 모두 동일한 디자인 내에 존재한다. 즉, 파트 파일과 어셈블리 파일과의 구분이 없어 멀티바디 탑다운 모델링이 가능하다.

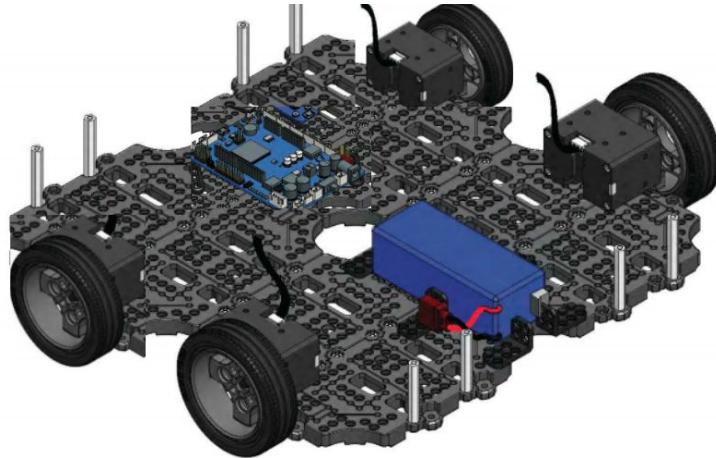


[그림 8] 3D 파일 슬라이싱

모델링한 파일을 3D 프린터 출력을 진행하기 위해서는 몇가지 과정을 거쳐야 한다. 3D 출력에 필요한 세팅 및 stl 파일을 g code로 변환을 해주기 위해서는 슬라이싱 과정을 해야한다. 슬라이싱 과정에서는 모델이 안정적으로 프린팅 될 수 있도록 출력 속도, 서포터의 여부, 프린팅 재료 설정 및 모델의 확장 및 축소 기능등이 있다. 슬라이싱 프로그램으로는 신도리코 사의 3dWox 슬라이싱 프로그램을 사용하여 진행을 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 2.2.1.3 하드웨어 설계



[그림 9] 첫번째 층


OLAF의 첫번째 층은 OPENCN보드 1개, 모터 4개, OPENCN보드에 전원을 넣어줄 배터리 1개를 사용하였다. 아무래도 최하층 및 구동에 필요한 요소들이 많이 들어간 부분이라 안전하게 고정을 해줄 필요가 있다.



[그림 10] 두번째 층

두번째 층은 각종 센서와 Jetson Tx2 보드에 연결을 할 USB 허브 및 전원을 공급할 배터리를 사용한다. 배터리의 경우 19V 출력을 보드에 연결하며 나머지 전원을 USB 허브에 연결한다.

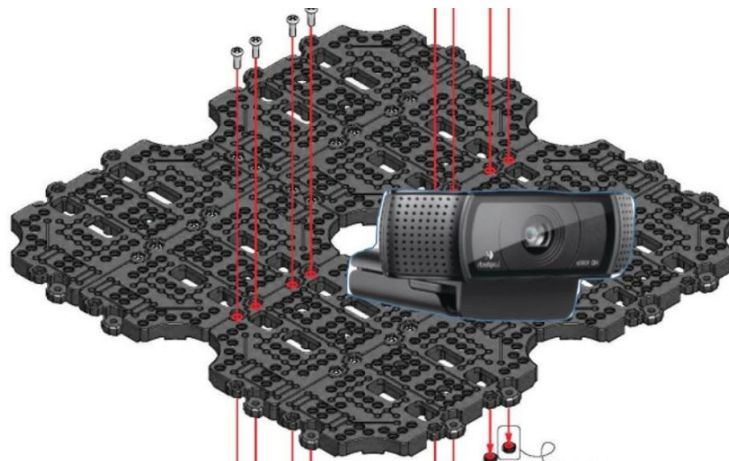


 <div> <p>국민대학교 컴퓨터공학부 캡스톤 디자인 I</p> </div>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20




[그림 11] 세번째 층

세번째 층은 Nvidia Jetson Tx2 보드가 들어간다. 층의 가로,세로 길이는 27cmx27cm 이며 보드의 가로, 세로 길이는 18cmx18cm 로 다른 센서를 장착하기에는 공간적 여유가 없으므로, 보드만 독립적으로 장착하였다.



[그림 12] 4번째 층

네번째 층은 IMU 및 카메라 모듈이 들어갈 층이다. 카메라의 시야각은 천장쪽에 있는 방번호판을 인식하기 용이하게 세로로 장착할 예정이며, IMU의 경우 단단히 고정시키도록 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20



[그림 13] 5번째 층


다섯번째 층은 LiDAR를 위치시키기 위한 층으로, 최상층에 위치한다. 층 높이는 사람의 무릎 높이에 위치하며 장애물 인식 및 Localization 하는데 용이하게 로봇의 정중앙에 위치시키도록 한다.



총 5층으로 구성된 OLAF의 크기는 가로,세로,높이 40cm x 40cm x 60cm의 로봇으로 실내에서 큰 제약없이 주행이 가능하게 설계를 하였다. 주행적인 특이 사항으로는 제자리에서 360도 회전이 가능하며, 최대 속도 시속 3km까지 나올 수 있다. 바퀴의 경우 기존에 터틀봇에서 사용하는 바퀴를 사용한 것이 아닌, Xycar-A2에 있는 몬스터 바퀴를 달았다. 각 층마다 주행시 진동을 줄이기 위해 쇠기둥을 앞,뒤, 양옆으로 8개씩 기본으로 달았으며, 최하층의 경우 중앙에도 2개를 추가함으로써 진동을 최대한 줄일수 있게 설계하였다.

[그림 14] 완성도

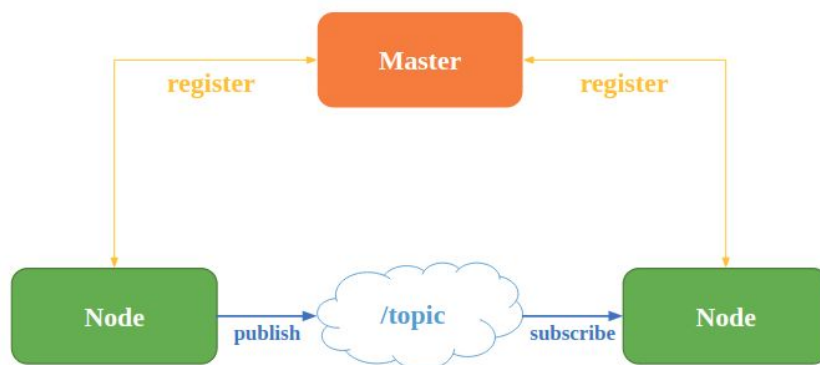


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2.2.2 ROS(Robot Operating System) 개발 환경 구축

ROS 는 로봇 개발을 편리하게 하고 관리 및 운용할 수 있도록 유닉스 기반의 다양한 운영체제에서 사용 가능할 수 있게 설계되어 하드웨어의 가상화, 기기종 디바이스 간의 통신 지원, low level 장치 제어, 로봇을 구성하는 다양한 하드웨어 모듈들의 동작제어에 필요한 응용 프로그램 패키지들을 제공한다. 이 외에 3차원 시뮬레이터인 Gazebo, 시각화 도구인 Rviz, GUI 도구모음 RQT 등의 유틸리티등을 지원한다.

ROS는 topic, service, action service와 같은 분산처리 환경들을 제공하는데 가장 많이 사용되는 topic 메시지를 통해 publisher와 subscriber로 구성된 두 노드 프로세스간에 TCP-IP 소켓 통신을 하도록 함으로써 연동하게 된다. OLAF 에서의 주 topic 메시지 형태는 각 센서 데이터 및 센서 데이터를 처리한 자료형으로 구분이 되며, publisher 로는 IMU, LiDAR, ENCODER, CAMERA 등이 사용되며, 각 알고리즘마다 subscribe 노드를 생성하여 각 센서 데이터를 처리 및 변환하는 과정을 진행한다.



[그림 15] ROS 기본 통신 방법

Nvidia Jetson Tx2는 Linux Ubuntu 16.04 운영체제를 지원하고 있으며, 이에 연동되는 ROS 버전을 구축해야한다. Ubuntu 16.04에 지원되는 ROS Kinetic 버전을 선택하였으며, 각 센서 및 구동에 필요한 패키지들을 설치한다. source.list, key 세팅 및 bashrc 스크립트 셋업을 통해 ROS 패키지를 위한 환경 세팅도 함께 진행되어야한다.

## 2.2.3 Sensors

### 2.2.3.1 LiDAR(Light Imaging Detection and Ranging)




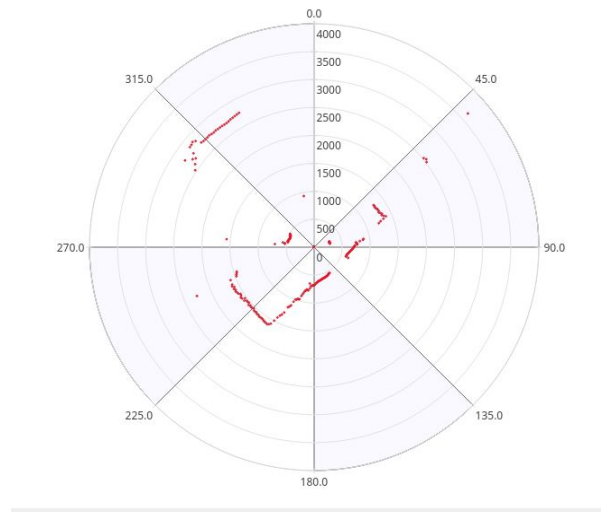
Items	Specifications
Distance Range	120 ~ 3,500mm
Distance Accuracy (120mm ~ 499mm)	±15mm
Distance Accuracy(500mm ~ 3,500mm)	±5.0%
Distance Precision(120mm ~ 499mm)	±10mm
Distance Precision(500mm ~ 3,500mm)	±3.5%
Scan Rate	300±10 rpm
Angular Range	360°
Angular Resolution	1°

[그림 16] 360 Laser Distance Sensor LDS-01 & Specifications

실내 측위 및 장애물 인식을 하기 위해서는 물체의 거리를 탐지하는 센서가 필요하다. 대표적으로 떠오르는 센서는 Radar(Radio Detection and Ranging) 와 LiDAR 로 두 센서간의 성향은 차이가 있다. 간단히 차이를 설명하면 Radar 의 경우 전자기파(마이크로파)를 발생시켜 물체의 위치를 파악하며, LiDAR는 펄스 레이저 즉, 쉽게 말하면 빛을 이용하여 주변을 탐색하는 장치이다. Radar의 경우 넓은 범위를 인식하지만 작은 물체를 감지하기 어려우며, 물체의 정밀한 이미지를 제공하지 못하는 단점이 있다. 반면 LiDAR의 경우 짧은 주파수 영역을 사용하기에 작은 물체도 감지가 가능하며 정밀한 이미지를 제공할 수 있다. 날씨의 영향 및 긴 거리의 영역은 탐지를 못한다는 단점이 존재하지만 실내에서는 이러한 문제점이 많이 나타나지 않아 프로젝트를 수행하기 위해 LiDAR를 선택하였다.

[그림 16]의 LiDAR는 HL Data Storage 사의 LiDAR로 OLAF의 주 센서로 이용될 것이다. [그림 16] 의 Specifications 에서 볼 수 있듯이 360도의 방향을 오차범위 5% 이내로 센싱이 가능하다.

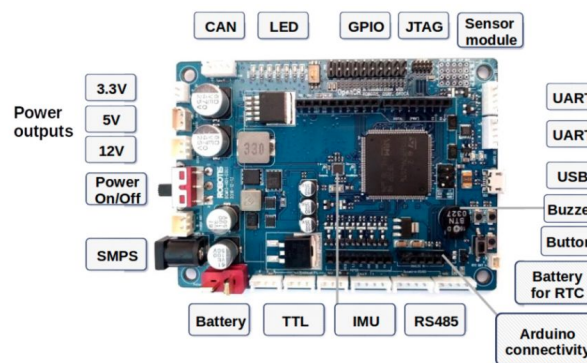
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20



[그림 17] LiDAR 데이터 시각화


[그림 17]은 ros-kinetic-hls-lfcd-lds-driver 오픈소스 패키지를 이용하여 데이터를 간단한 시각화 코드를 작성후 출력된 결과물이다. [그림 17]에서 보는 바와 같이 현재 센서를 기준으로 120mm ~ 3500mm 거리를 센싱을 할 수 있는것을 확인할 수 있다.

### 2.2.3.2 OPENCN(Open-source Control Module for ROS) + DYNAMIXEL



[그림 18] OPENCN

Opencr은 Dynamixel 모터를 구동하기위한 서브 컨트롤러 보드이다. ROS 임베디드 시스템 용으로 개발이 되었으며 오픈 소스 하드웨어 및 소프트웨어를 제공한다. Dynamixel 모터와 직접 연결이 되는 커넥터에 아두이노 지원, UART, RS485, CAN 등의 다양한 통신 지원 및 MPU9250 이라는 IMU 센서를 가지고 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20



[그림 19] Dynamixel

Dynamixel은 디지털 통신으로 제어되는 로봇 전용 액츄에이터이다. 감속기, 제어기, 구동부, 통신부가 하나로 통합된 일체형 구조로 고유의 ID를 가진 Dynamixel들은 하나의 버스상에서 Packet 통신으로 제어되며, 모델에 따라 TTL, RS485 등의 네트워크를 지원한다.


Dynamixel의 가장 큰 장점 중의 하나는 비접촉식 엔코더를 적용하여 내구성, 정밀도, 제어 범위적으로 큰 이점이 있으며, 엔코더의 값으로 선가속도와 각가속도를 얻을 수 있다. 이는 로봇의 위치 추정에 가장 중요한 정보 중의 하나가 되며, 그 값 또한 정밀하게 나오는 것을 확인한다.

### 2.2.3.3 IMU



[그림 20] E2BOX-9DOF-IMU

IMU는 무인비행체(드론), 비행기, 인공위성 등의 기기의 내부에 들어가는 부품으로써 가속도계, 자이로스코프, 지자계를 이용하여 가속도와 회전율, 즉 관성을 측정, 3차원 공간에서의 움직임을 측정하고 방위각을 위해 자기장을 측정할 수 있게 해 주는 통합된 Unit을 말한다. 최근, 터널이나 실내 혹은 전자기 간섭이 있는 공간에서 GPS 신호의 수신에 문제가 있을 때 IMU 에서의 정보를 바탕으로 위치를 추측하는 기술이 네비게이션과 같은 곳에서 유용하게 쓰인다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

IMU에서 측정된 자이로와 가속도, 지자기센서를 이용하여 Roll, Pitch, Yaw 및 선,각속도 데이터를 이용하여 위치를 칼만 필터링 및 연산방식을 통해 데이터 값의 오차를 줄인 후 실내 위치인식을 한다. 즉, 우리의 목표는 IMU센서 데이터 및 Encoder 데이터를 이용하여 SLAM 및 위치 추정의 구현 및 이식이다.

#### 2.2.3.4 CAMERA




[그림 21] Logitech C920 Pro HD

로봇의 위치추정 및 네비게이션시 Encoder 와 IMU의 데이터만을 가지고 실내 주행은 상당히 많은 제약 조건이 따른다.

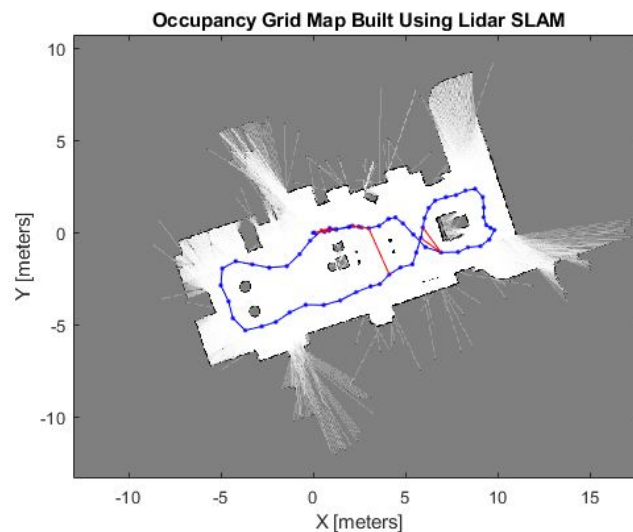
첫번째는 센서 데이터의 오차의 누적이다. 현재 보유하고 있는 센서들의 경우 급격한 제어 및 환경이 주어졌을때 오차가 쌓이게 된다. 이는 결과론적으로 시스템에 치명적인 오류를 범할 수 있으며, 다른 해결방안이 필요하다. OLAF 는 실내 주행시 실시간으로 각 교실 상단에 붙어있는 방번호판을 인식한다. 이로써 OLAF는 현재 자신이 어디쯤에 위치하고 있는지에 대한 정보를 얻으며, 이 정보를 이용해 위치 보정 알고리즘을 수행하게 된다.

두번째는 물체 인식이다. 실내 공간은 비교적 좁은 공간에서 사람이나 물건이 돌아다니는 지역이다. 아무리 정교하게 지도를 작성하여 주행을 한다고 해도 돌발 변수에 대처하는 능력이 필요하다. 사람, 장애물 및 위치 추정에 필요한 정보들을 인식하는 것이 필수 불가결한 요소이다.

따라서 카메라 센서를 이용하여 로봇의 위치 추정 및 물체 인식을 수행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2.2.4 SLAM(Simultaneous Localization And Mapping)




[그림 22] 2D LiDAR based SLAM

SLAM이란 Map도 주어져 있지 않고 Map에서의 로봇의 위치도 알 수 없을때 로봇이 주변 환경을 센서로 감지하면서 Map을 만들고 그 Map에서의 자신의 위치까지 추정하는 작업을 가리킨다.

SLAM 알고리즘은 크게 다음과 같은 2가지로 나뉘어져 있다.

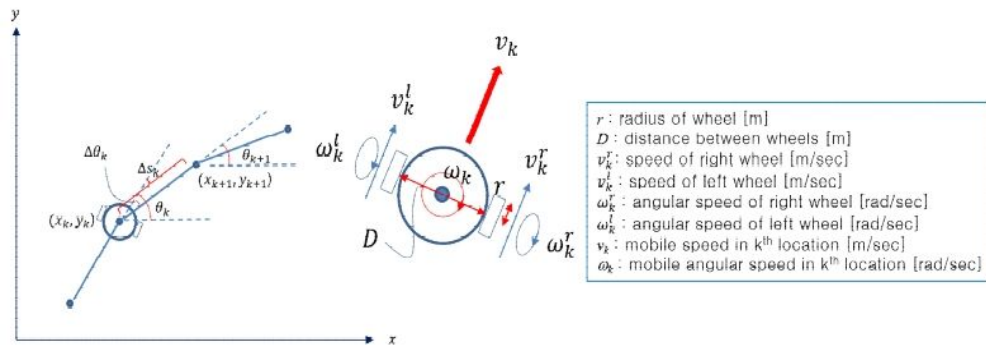
- EKF-SLAM
- Fast-SLAM

EKF-SLAM은 Extended Kalman Filter 알고리즘의 기본 전제인 Gaussian Noise Assumption을 그대로 사용하고 있다. 즉, 로봇의 State와 Map의 Landmark 들을 Mean과 Covariance Matrix로 나타내고 Observation이 일어날 때마다 그것들을 업데이트 시켜주는 방식이다. Covariance Matrix를 유지해야 하고, Matrix 연산이 일어나기 때문에 상당히 느리다. 그리고 Gaussian Assumption은 대부분의 경우 성립하지 않는 단점이 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

FastSLAM은 MonteCarlo Localization에도 사용되는 Particle Filtering 알고리즘을 기반으로 하고 있다. 여기서는 Linear Gaussian Assumption을 사용하지 않으므로 좀 더 일반적인 경우에 적용시킬 수 있다고 볼 수 있다. 각각의 Particle마다 Map과 Pose를 갖고 있고, Observation에 따라 현재 time step에 대해 Sampling을 한다. Sampling한 결과에 대해 Likelihood를 따져 본 다음, 가장 Weight가 높은 Particle을 다음 time step에서도 사용하는 것이다.

FastSLAM의 경우 대표적인 종류로는 Gmapping이 있다. OpenSlam에 공개된 SLAM의 한 종류로 Particle Filtering 기반의 SLAM이다. 필요한 센싱 데이터로는 X,Y, Theta 속도 이동 명령과 주행기록계(Odometry) 및 LiDAR의 scan 값이 필요하다.




[그림 23] Gmapping 동작 원리

Gmapping 동작원리에서 측위 과정에 있어서는 [그림23] 에서 보는 것처럼 k시점에서 로봇이 2차원 x-y 에서의 위치 및 그 위치에서의 방향각으로서  $(x_k, y_k, \theta_k)$ 과 같이 표현할 수 있다.

이 때 로봇의 좌우바퀴 Encoder, 회전속도 센서를 통해 탐지된 좌우바퀴의 회전 각속도와 바퀴의 반경  $r$ , 바퀴사이의 거리  $D$ 로 부터  $k$  시점에서의 로봇의 병진 속도 및 로봇 회전 각속도를 각각 구할 수 있다. 현재 로봇의 병진속도 및 회전각속도로부터  $\Delta t$  시간이 지난  $k+1$  시점에서의 이동거리  $\Delta s_k$  및 방향회전각  $\Delta \theta_k$  을 구할 수 있고, 이로부터  $k+1$  시점에서의 위치 및 방향각을 구함으로써 측위한다.

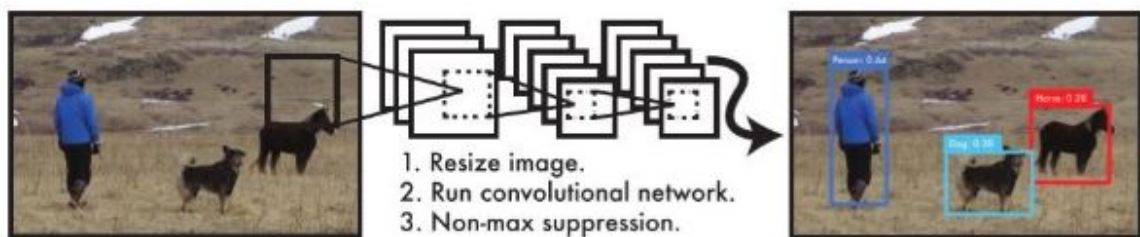
OLAF는 위 두가지 방법의 SLAM을 적용하여 둘 중의 성능이 안정적인 모델을 선택하여 Mapping을 진행한다. Mapping의 경우 시간적인 비용보다는 정확도가 우선시 되기 때문에 좀 더 정밀한 정보가 생성되는 방법을 채택하여 사용한다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2.2.5 OBJECT DETECTION YOLO

Object Detection이란 머신러닝에서 한 사진안에 어떤 물체들이 등장하는지에 대한 것을 판별해주는 Task를 말한다. Object Detection을 위한 네트워크는 Task의 특징에 따라 다양한 네트워크들이 존재하는데 예를 들면 R-CNN, SAS 딥러닝, YOLO 등이 대표적이다. 우리는 실시간 환경에서 빠른 시간안에 인식을 진행해야 하기때문에 세 모델중 YOLO를 네트워크 모델로 선정했다.




[그림 24] YOLO Detection System

YOLO(You Only Look Once)는 이미지 내의 bounding box와 class probability를 single regression problem으로 간주하여, 이미지를 한 번 보는 것으로 object의 종류와 위치를 추측한다. [그림 24]과 같이 single convolution network를 통해 multiple bounding box에 대한 class probability를 계산하는 방식이다.

기존의 Object detection method와 비교했을 때, YOLO가 보여주는 상대적인 장점과 단점은 다음과 같다.

- **장점**
  - 간단한 처리과정으로 속도가 매우 빠르다. 또한 기존의 다른 real-time detection system들과 비교할 때, 2배 정도 높은 mAP를 보인다.
  - 이미지 전체를 한 번에 바라보는 방식으로 class에 대한 맥락적 이해도가 높다. 이로인해 낮은 background error를 보인다.
  - Object에 대한 좀 더 일반화된 특징을 학습한다. 다른 Detection System들에 비해 훨씬 높은 성능을 보여준다.
- **단점**
  - 상대적으로 작은 object들에 대해 낮은 정확도를 나타낸다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


## 2.2.6 Data and Network

### 2.2.6.1 국민대학교 미래관(구 7호관) 데이터셋 구축

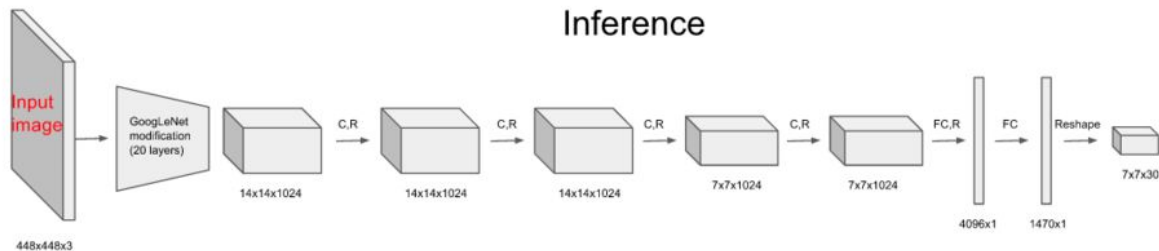


[그림 25] 국민대학교 미래관 데이터 수집 및 라벨링

[그림 25]의 왼쪽 사진은 국민대학교 미래관 7층에서 로봇에 카메라를 부착하여 일정 속도로 이동하는 영상을 1초단위로 끊어서 저장한 이미지이다. 로봇 테스트를 할 시 동영상 저장 기능을 추가하여 데이터를 수집하며, 수집된 데이터는 팀 ELSA에서 직접 labeling tool을 이용하여 제작한다. 현재 초기 설정된 클래스로는 번호판, 엘리베이터, 사람, 장애물이 있으며, 실험을 거쳐 클래스를 추가 혹은 삭제할 계획이다. 목표 데이터의 개수는 2000장이며, 모두 국민대학교 미래관 4층에 대한 데이터를 기준으로 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2.2.6.2 네트워크




[그림 26] 네트워크 구조

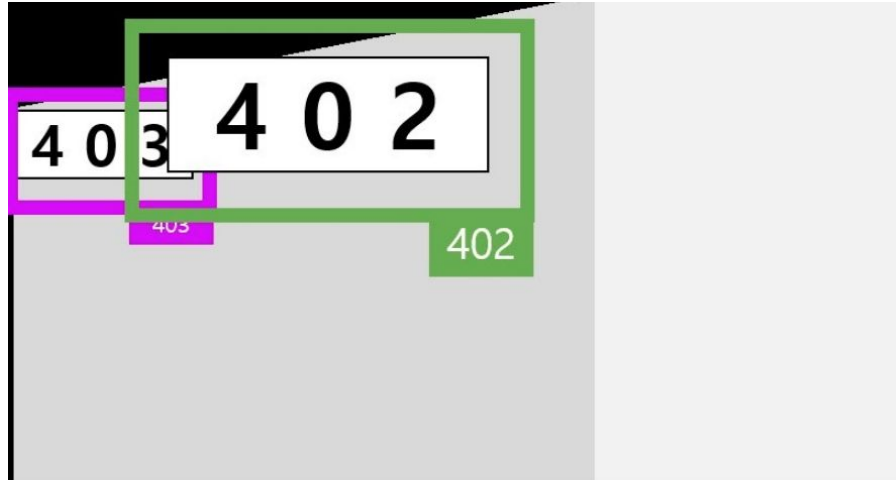
네트워크의 큰 특징 중 하나는 이미지로부터 features를 뽑아서 각 bounding box를 예측한다는 것이다. 시스템은 input image를  $S \times S$  grid로 나눈다. 그리고 만약 어떤 object의 한 가운데가 grid cell에 놓인다면, grid cell은 그 object를 탐지할 의무가 생긴다는 설정이다. 각 grid cell은 B개의 bounding box를 예측하고 그 box의 예측에 대한 confidence score를 예측한다. confidence score은 얼마나 박스안에 실제로 object가 존재하는지, 그리고 그 class를 얼마나 잘 반영했는지에 대한 것이다. Confidence는 형식적으로  $\text{Object} \times \text{IOU}(\text{Intersection Over Union})$  라고 정의를 한다. 예를 들면 object가 box안에 아무것도 없을 때에는 confidence가 0이 되어야 한다. 또, predicted box와 ground truth의 IOU가 얼마나 일치하는지가 confidence가 되도록 맞추고 싶다. 각 bounding box는 5가지 예측을 하게된다.  $x, y, w, h, \text{confidence}$  이다.  $(x,y)$  는 grid cell의 영역에 관련된 box의 center를 의미한다. width, height ( $w,h$ )는 박스의 높이와 너비로서, 이미지 전부와 비례하게 예측된다. 마지막으로 confidence는 ground truth box와 예측된 box의 IOU를 예측한다. 각 grid cell은 C개의 조건부 확률을 예측한다.

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

### [식] Unified Detection


[식]은 얼마나 box를 잘 예측했는지, 그리고 object의 class를 잘 맞추었는지를 나타내는 score가 된다. 결국 최종적인 예측은  $7 \times 7 \times 30$ 의 텐서가 되며 우리는 클래스의 개수에 따라 마지막 channel 수를 조정하는 과정을 거친다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20



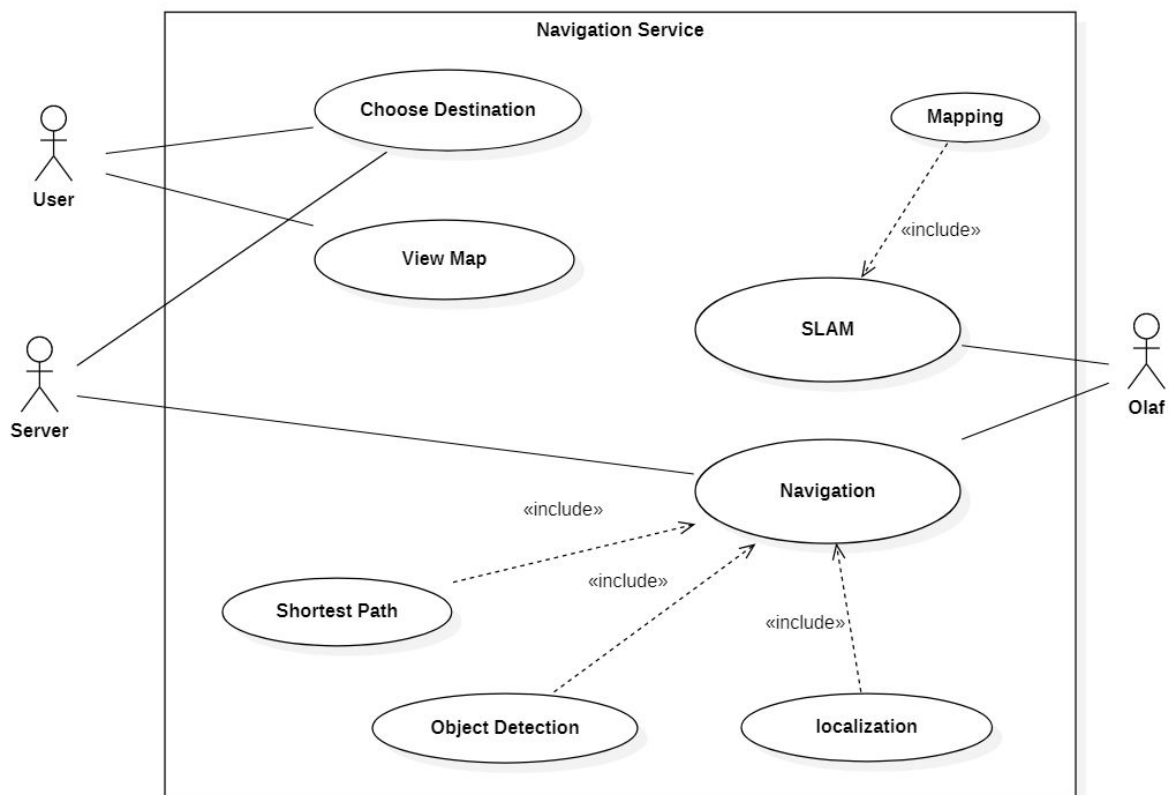
[그림 27] 방번호 인식 및 번호 추출

Object Detection시 방 번호판이 인식되면, 그 번호판이 어느 교실의 번호판인지를 판단하는 알고리즘이 필요하다. 번호판 영역을 따로 추출한 뒤, 윤곽선 기반 인식(Contour)을 하여 번호가 적힌 영역을 추출해낸뒤, 미리 mnist 데이터가 트레이닝된 모델에 적용하여 번호를 추출한다. 추출된 번호를 기반으로 OLAF는 현재 자신이 어느 위치에 도달했는지 다시 확인할 수 있으며, 이를 토대로 위치를 보정하게 된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 2.3 개발 결과


### 2.3.1 시스템 기능 요구사항



### 2.3.2 시스템 비기능(품질) 요구사항

#### 2.3.2.1 신뢰할 수 있는 실시간 위치추정

실시간적으로 정확한 위치추정과정을 위해 전자파와 카메라 인식을 위한 조명, 사람의 동선을 제한하여 안정적인 환경을 구성해야 한다. 직접적으로 받는 자기장이 20mG보다 심한 경우는 지자기 센서(IMU)에 영향을 끼쳐 방향추출에 있어서 어렵다. 주변에 자기장이 20mG미만의 환경이라 가정한다. 밤이나 어두운 날씨임에도 조명을 없을 경우, 위치 보정에 사용하는 카메라가 무의미하게 된다. 하지만 실내공간의 밝기가 일정 수준의 이상이라는 것을 가정한다. 동적 물체(사람, 장애물 등)들이 1분이상 정지해있지 않는다고 가정한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


### 2.3.2.2 안정적인 서버 통신

앞에서 제한한 환경에서 확실한 위치 추정을 하고, IEEE 802.11 무선 근거리 통신망 네트워크 주파수 2.4GHz, 대역폭 20MHz 상황에서 통신을 기본으로 한다.

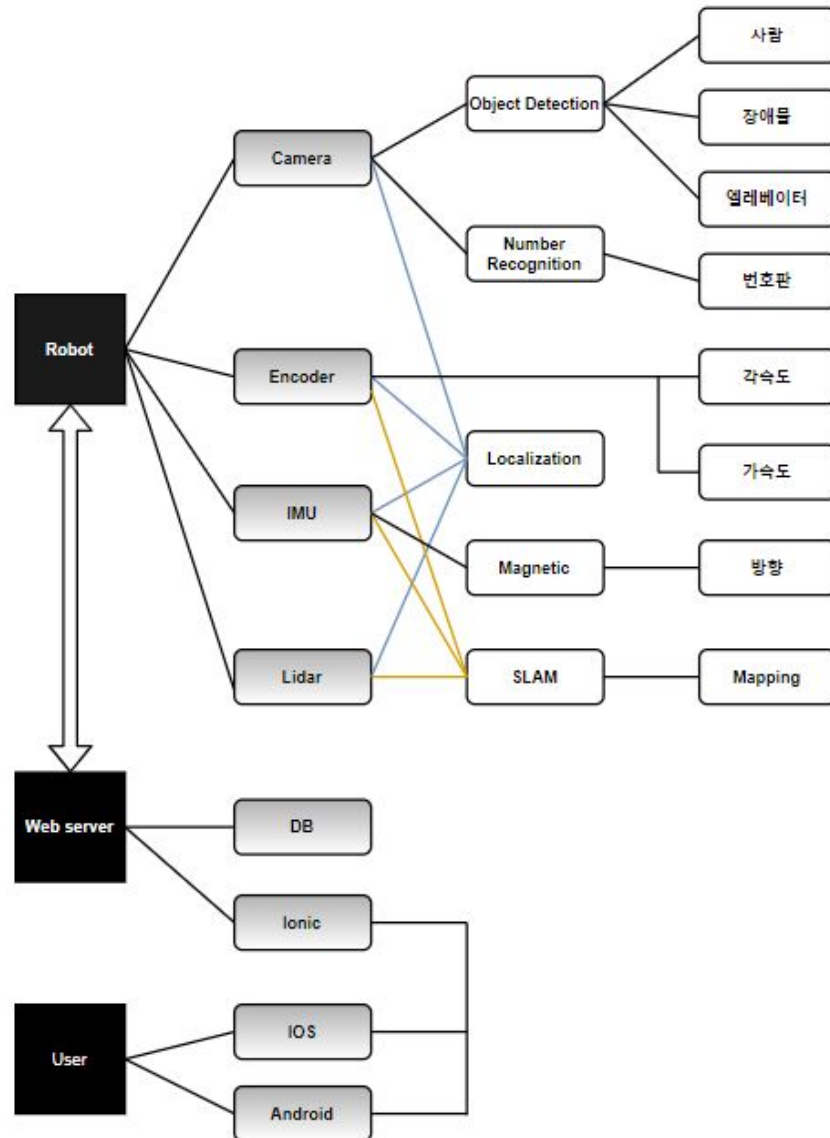
사용자가 앱을 통해 목적지를 입력받아 확실한 최단경로를 제공하고, 목적지까지 얼마나 왔는지에 대한 서비스를 제공하고 도착을 알려주기 까지 각각은 서버와에 통신을 3초 이내를 목표로 하고 로봇과 서버간에 현재 위치 데이터 통신 또한 3초 이내를 목표로 한다.

### 2.3.2.3 신속한 돌발상황 대처를 위한 Object Detection 알고리즘


로봇이 길을 안내하는 도중 실내공간에서 동적 장애물이 갑자기 나오거나 사람이 지나가는 돌발상황에서도 카메라를 통한 Object를 인식하고 대처를 위한 처리 속도가 1초 이내를 목표로 한다. 카메라 및 LiDAR를 이용하여 긴급 정지 모듈을 구축함으로써 안정성을 확실히 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 2.3.3 시스템 구조



[그림 28] 시스템 아키텍처

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 2.3.3.1 Localization


먼저 SLAM을 통해 local map을 구한다. 그리고 나서 LiDAR를 통해 실시간으로 측정되는 값을 local map과 비교하여 자기 위치를 판단한다. LiDAR 센서 하나 만으로는 안전성을 확보 할 수 없으므로 IMU 센서를 통해 방향을 정하고 Encoder 값으로 각속도, 가속도 값을 계산하여 이동할 때마다 자신의 위치 값을 수정하게 된다. 또한 정확성을 더욱 높이기 위해 Camera를 통해 강의실 번호를 인식하여 자신의 위치를 계산하는데 보완 할 수 있다.

### 2.3.3.2 SLAM

SLAM 은 ‘Simultaneous Localization and Mapping’ 의 약자로 동시적 위치 추정과 지도를 작성하는 기술이다. SLAM은 IMU로 부터 방향, Encoder로 부터 가속도와 각속도, LiDAR를 통해 주변 환경의 값을 받아 데이터를 수집 하고 자신의 위치를 파악하며 지도를 작성한다.

### 2.3.3.3 Navigation


SLAM을 통해 생성된 local map과 Localization을 통해 나오는 자신의 위치를 이용한다. local map에 자신의 위치와 목적지가 주어진다면 자신의 위치와 목적지 사이에 waypoint를 생성하고 최단 경로 알고리즘을 통해 목적지까지 이동한다. Navigation 중 Camera를 통해 장애물을 발견하게 되면 회피 기동을 하여 목적지까지 이동한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 2.3.4 결과물 목록 및 상세 사양

대분류	소분류	기능	형식	비고
하드웨어	NVIDIA Jetson TX2	Localization, SLAM 수행 센서 데이터 처리	Python ROS C++ Cuda	
	E2BOX-EBIMU 9DOFV5	각속도, 가속도, 방향 데이터	Serial	
	360 Laser Distance Sensor LDS-01	Scanning 데이터	Serial	
	Logitech C920 Pro HD	영상 데이터	Serial	
	OpenCR 1.0	모터 제어	ROS	
	DYNAMIXEL AX-12A * 4	Encoder 데이터	Serial	
하드웨어 출력	신도리코 DP 200	3D 모델링 출력	3D Printing	
분석 데이터	MNIST	숫자 인식용 데이터	CSV	
네트워크 모델	YOLO, Darknet	Object Detection	Pretrained Model	
프로그램 코드	ROS 패키지	로봇 제어 및 센서 데이터 처리	ROS Python C++	



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


## 2.4 기대효과 및 활용방안

실외 공간은 GPS를 통해 현재 위치를 알고 경로를 탐색해주지만 실내 공간은 GPS로 하기에 어려움이 있어 표지판을 통해서 길을 알거나 내부인들에게 물어봐서 길을 찾기 마련이다. 이렇듯 우리는 학교뿐만이 아닌 코엑스와 같은 넓은 실내공간에서 외부인, 외국인과 같은 초행자들에게 길을 안내 해주는 로봇을 통해 보다 빠르고 짧은 경로를 안내할 수 있다. 그리고 이 로봇을 통해 사람들의 삶의 질을 높이고 상업적으로도 수익을 얻을 수 있다.

첫째로, ‘이마트’, ‘롯데 마트’ 와 같은 대형 실내 장소에서 원하는 물품을 입력하면 그 물품의 정보를 제공해주고 물품이 있는 실내공간의 최단 경로를 안내해주는 방법을 논의 해볼 수 있다.

둘째로, 기차 및 열차와 같은 장시간 이용하는 실내장소에서는 사용자들에게 게임, 콘텐츠를 제공합니다. 그리고 기차 및 열차의 도착예정 기차 및 열차칸들의 지도를 제공하여 복지시설의 위치, 좌석의 자리 안내 더불어 간단한 음료판매까지도 확장할 수 있다.

이러한 것들을 제공하여 주고, 상업적으로는 광고를 추가하여 수익을 얻을 수 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 3. 배경기술

#### 3.1 기술적 요구사항

##### 3.1.1 개발 환경

- 운영체제 : Linux Ubuntu 16.04
- SBC : Nvidia Jetson TX2
- 메타 운영체제 : ROS
- 개발 언어 : Python, C++, Cuda, Java, js, html
- 프레임워크 : Pytorch

##### 3.1.2 프로젝트 결과물 확인 환경

- 실제 환경
- 안드로이드, IOS 모바일 기기

#### 3.2 현실적 제한 요소 및 그 해결 방안


##### 3.2.1 하드웨어 제한 요소

###### 3.2.1.1 바퀴

기존 Turtlebot3 에서 제공되는 바퀴는 바퀴 휠이 원형 모양이 아닌 톱니 모양이다. 따라서 로봇이 움직일때 마다 로봇에 떨림이 발생하여 센서값에 오차가 발생하게 된다. 이를 해결하기 위해 바퀴 휠을 3D 모델링을 통해 다이나믹셀에 맞는 완전한 원형으로 구현 하였고 바퀴 타이어도 기존 타이어에서 안전성을 위해 더 크고 쿠션이 들어간 타이어로 교체하였다.

###### 3.2.1.2 LiDAR

현재 프로젝트에서 사용하고 있는 라이더는 2D 1채널 라이더로 벽을 인식하는 것에는 문제가 없다. 하지만 유리, 빛이 반사되는 벽면, 계단 같은 경우에는 인식을 하지 못하여 Mapping 하는 부분에서 오차가 발생 하게 된다. 따라서 Mapping 과정에서 계단과 같이 절대 가지 말아야 할 곳은 따로 처리를 해주고 유리나 빛이 반사되는 곳은 빛이 반사가 안되도록 포맥스 같은 판으로 가리고 난 후 local map을 mapping 하는 과정을 진행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 3.2.1.3 로봇의 견고함

로봇의 형태가 일체형이 아닌 와플 플레이트를 적층형으로 쌓아 서포터로 고정하는 형식이다. 각 층마다 센서를 탑재하는데 작은 충격으로도 로봇의 떨림이 발생하여 센서에 오차값이 생긴다. 로봇의 일체형 형태의 하드웨어를 구하는데 한계가 있기 때문에 와플 플레이트를 견고하게 하기 위해서는 서포터를 구하여 더욱 더 많이 연결하였다.

### 3.2.1.4 카메라 각도

현재까지 구상 된 로봇의 높이는 약 60cm 로 카메라의 위치가 현저히 낮은 위치에 존재한다. 이에 따른 문제점으로 카메라가 아래에서 위를 향하는 각도로 부착을 하게 되는데 실제 동작시 건물 내부에 사람이 많이 존재하게 되면 Localization의 정확도를 위해 강의실 번호를 인식하는데 어려움을 겪게 된다. 또한 카메라위치보다 낮은 위치에 장애물이 존재하면 카메라로 장애물을 인지하지 못해 회피기동을 하는데 어려움을 겪는다. 카메라 위치를 조절하기 위해 카메라 암을 달아 카메라 위치를 높이는 방법을 구상중이며 장애물판단 하는데에 카메라 뿐만 아니라 라이다 센서도 같이 활용 하는 방법을 구상중이다.

## 3.2.2 소프트웨어 제한 요소


### 3.2.2.1 Encoder, IMU, Camera, LiDAR 센서들을 이용한 위치추정

Encoder센서에서 속도와 각속도를 알 수 있고 IMU를 통해 방향을 알 수 있다. 그리고 LiDAR를 통해 벽과의 거리를 유지하며 카메라의 호실의 숫자를 인식하여 위치를 보완한다. 이렇게 각종 센서들을 이용하는 이유는 최소의 오차를 만들기 위해서이다. 그러나 센서들에서 측정값들의 오차값이 누적될 수 있다. 이러한 오차도 줄이기 위해 칼만필터를 사용한다. 그럼에도 불구하고 더 적은 오차를 만들 수 있는 필터들을 연구해야 할 것으로 보인다.

## 3.2.3 기타 제한 요소


### 3.2.3.1 웹/앱을 이용한 통신

현재 상용되고 있는 안내로봇인 에어스타를 살펴보면 사용자는 아무런 제약 없이 에어스타를 이용할 수 있다. 즉 에어스타에 음성인식, 터치스크린 등이 있어 사용자가 바로 에어스타와 소통 할 수 있다. 하지만 OLAF에는 사용자와 직접 소통 할 수 있는 수단이 없다. 사용자와 소통 하기 위해서는 사용자가 웹을 통해야만 한다. 따라서 사용자는 핸드폰을 소지해야만 안내로봇 서비스를 이용 할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


### 3.2.3.2 다른 층으로의 안내

만약 사용자의 원하는 목적지가 현재 위치한 층이 아닌 다른 층으로 안내를 원하는 경우에는 상당한 제약이 따른다. 로봇으로 계단을 이용하거나 엘리베이터를 이용하는 것은 하드웨어적으로 한계가 분명하다. 따라서 이 문제를 해결하기 위해 각 층마다 안내 로봇이 있다는 가정 하에 진행 하려고 한다. 안내 로봇이 엘리베이터 까지만 안내를 하고 사용자가 목적지 층으로 이동을 하면 목적지 층에 있는 안내로봇에게 신호를 보내 탑승한 엘리베이터 위치로 이동하게 한다. 그 뒤 목적지 층에 있는 로봇이 안내를 시작하는 방식으로 진행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


#### 4. 프로젝트 팀 구성 및 역할 분담

이름	역할
김다훈	<ul style="list-style-type: none"> <li>- Software Project Leader</li> <li>- IMU 캘리브레이션 및 데이터 처리</li> <li>- LiDAR 센서 데이터 처리 및 SLAM 알고리즘 수정</li> <li>- 사용자에게 제공할 최단경로 알고리즘 개발</li> <li>- 시스템 구조 설계 및 Nvidia Jetson Tx2 커널 수정</li> <li>- 회의록 및 문서 관리</li> <li>- Dataset 라벨링</li> </ul>
김선필	<ul style="list-style-type: none"> <li>- Hardware 설계 및 관리</li> <li>- 3D 모델링 및 프린팅</li> <li>- LiDAR 센서 데이터 처리 및 SLAM 알고리즘 수정</li> <li>- Object Detection 개발 및 관리</li> <li>- Dataset 라벨링 툴 개발</li> <li>- 시스템 구조 설계</li> </ul>
김명수	<ul style="list-style-type: none"> <li>- 사용자에게 길안내 서비스(UI) 개발 및 관리</li> <li>- AWS EC2 를 이용한 웹 서버 작업</li> <li>- ionic 빌드를 통한 application 빌드 (android, ios)</li> <li>- Serial 통신 설정</li> <li>- 시스템 구조 설계 및 Nvidia Jetson Tx2 설정</li> <li>- Dataset 라벨링</li> </ul>
배한울	<ul style="list-style-type: none"> <li>- IMU 캘리브레이션 및 데이터 처리</li> <li>- 각 센서와의 Nvidia Jetson Tx2 연동 시스템 구축</li> <li>- 모터 컨트롤러 개발 및 관리</li> <li>- Navigation 시스템 개발 및 관리</li> <li>- Dataset 라벨링</li> <li>- Hardware 설계 및 관리</li> </ul>
윤찬우	<ul style="list-style-type: none"> <li>- IMU 캘리브레이션 및 데이터 처리</li> <li>- 번호 추출 알고리즘 개발 및 관리</li> <li>- Navigation 시스템 개발 및 관리</li> <li>- Dataset 라벨링</li> <li>- Hardware 설계 및 관리</li> <li>- LiDAR 센서 데이터 처리 및 SLAM 알고리즘 수정</li> </ul>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 5. 프로젝트 비용


항목	예산치(MD)
하드웨어 설계	50
소프트웨어 설계	50
Nvidia Jetson Tx2 개발 환경 구축	20
SLAM을 통한 지도 구축	30
카메라를 이용한 Object detection 및 강의실 번호 인식 구현 및 개발	50
IMU, Encoder, LiDAR를 통한 위치추정 알고리즘 개발	50
최단 경로 알고리즘 개발	20
로봇과 사용자 통신 및 UI 개발	30
AWS 서버 DB 개발	30
장애물 대처 알고리즘 개발	30
Application 개발	20
시스템 테스트	10
보고서 작성	10
합	370

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 6. 개발 일정 및 자원 관리

### 6.1 개발 일정


항목	세부내용	12월	1월	2월	3월	4월	5월	비고
요구사항 분석	주제 결정 및 시나리오 작성							
	시나리오에 따른 역할 분담							
	SRS 작성							
	결과에 따른 시나리오 수정							
관련분야 연구	ROS 학습							
	머신러닝과 딥러닝 학습							
	LiDAR 센서 학습							
	GPS, IMU 센서 학습							
	알고리즘 학습 YOLO / 위치추정 / SLAM							
	3D 모델링 및 프린팅							
	ionic 학습							
설계	소프트웨어 설계							
	하드웨어 설계							
	시스템 설계							
	알고리즘 설계							
구현	Localization 개발							
	web 개발							
	aws ec2 웹 서버 구축							
	ionic 빌드를 이용한 앱 개발							
	코딩 및 모듈 테스트							
테스트	전체 프로세스 및 시스템 테스트							

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20


## 6.2 일정별 주요 산출물

마일스톤	개요	시작일	종료일
계획서 발표	<ul style="list-style-type: none"> <li>프로젝트 시나리오 계획 및 구체화               <ul style="list-style-type: none"> <li>시나리오 구상</li> <li>시나리오를 수행하면서 시나리오 수정 및 개선</li> </ul> </li> <li>GitHub repository 생성               <ul style="list-style-type: none"> <li>캡스톤 양식에 따른 git 폴더 생성</li> <li>README.md 파일 작성</li> </ul> </li> <li>계획서 발표에 필요한 자료               <ul style="list-style-type: none"> <li>발표 슬라이드 제작</li> <li>발표 슬라이드에 따른 동영상 제작</li> <li>수행 계획서 작성</li> </ul> </li> <li>산출물               <ul style="list-style-type: none"> <li>프로젝트 수행 계획서</li> <li>프로젝트 발표 ppt</li> <li>프로젝트 발표 동영상</li> </ul> </li> </ul>	2020-01-10	2020-03-26
설계 완료	<ul style="list-style-type: none"> <li>하드웨어 &amp; 소프트웨어 시스템 설계 완료</li> <li>개발 환경 구축</li> <li>산출물               <ul style="list-style-type: none"> <li>시스템 설계 사양서</li> <li>하드웨어 부품</li> <li>센서 구매</li> </ul> </li> </ul>	2020-03-09	2020-03-26
중간 보고	<ul style="list-style-type: none"> <li>Hardware(OLAF) 설계 및 동작 테스트 완료</li> <li>Tx2 Board 개발환경 및 ROS를 통한 센서간의 연동 환경 구축</li> <li>Encoder센서 테스트</li> <li>IMU센서 테스트 및 캘리브레이션 완료</li> <li>LiDAR 센서 테스트 및 미래관 지도 구축 완료</li> <li>Camera 테스트 완료 및 YOLO Dataset 구축</li> <li>AWS EC2 Web Server 및 UI, API 구현</li> <li>산출물 :               <ul style="list-style-type: none"> <li>프로젝트 중간 보고서</li> <li>프로젝트 중간 발표 ppt</li> <li>프로젝트 중간 발표 동영상</li> </ul> </li> </ul>	2020-03-27	2020-04-23




 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

	<ul style="list-style-type: none"> <li>- 각 기능 별 구현 소스 코드</li> </ul>		
구현 완료	<ul style="list-style-type: none"> <li>- 앱과 로봇간 통신</li> <li>- Navigation 구현</li> <li>- 하이브리드 앱 구현 완료</li> <li>- 최단경로 알고리즘 구현</li> <li>- Localization 최적화</li> <li>- SLAM 최적화</li> <li>- Camera를 통한 Object Detection 구현</li> <li>- Camera를 통한 Number Recognition 구현</li> <li>- 회피 기동 구현</li> <li>- Waypoints 생성</li> <li>- Ionic 을 통한 하이브리드 앱 빌드 테스트 완료</li> <li>- AWS EC2 Server 최적화</li> <li>● 산출물: <ul style="list-style-type: none"> <li>- 각 기능 별 구현 소스 코드</li> </ul> </li> </ul>	2020-04-24	2020-06-04
테스트	<ul style="list-style-type: none"> <li>- 시스템 통합 테스트</li> <li>● 산출물: <ul style="list-style-type: none"> <li>- 구현 소스 코드</li> <li>- 전시용 자료</li> <li>- 온라인 평가용 자료</li> </ul> </li> </ul>	2020-06-05	2020-06-11
최종 보고서	<ul style="list-style-type: none"> <li>- 최종 보고</li> <li>● 산출물: <ul style="list-style-type: none"> <li>- 최종 결과 보고서</li> </ul> </li> </ul>	2020-06-12	2020-06-19

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

### 6.3 인력자원 투입계획


이름	개발항목	시작일	종료일	총개발일(MD)
김선필 윤찬우 배한울 김다훈	<ul style="list-style-type: none"> <li>하드웨어 설계</li> <li>3D 모델링 및 프린팅</li> <li>구조 설계</li> <li>부품 조달</li> </ul>	2020-01-10	2020-03-02	40
김선필 윤찬우	<ul style="list-style-type: none"> <li>IMU</li> <li>칼만필터 적용</li> <li>센서 값 추출</li> </ul>	2020-01-10	2020-02-18	30
김명수	<ul style="list-style-type: none"> <li>길 안내 서비스(web) 개발</li> </ul>	2020-01-10	2020-02-21	20
김다훈 배한울	<ul style="list-style-type: none"> <li>Nvidia Jetson Tx2 보드 환경 설정 및 관리</li> </ul>	2020-02-21	2020-03-02	10
김선필 김다훈	<ul style="list-style-type: none"> <li>SLAM 연구 및 개발</li> <li>LiDAR 센서 활용</li> <li>IMU Calibration 코드 작성</li> </ul>	2020-02-18	2020-03-21	30
배한울 윤찬우	<ul style="list-style-type: none"> <li>IMU</li> <li>노드 간 통신</li> <li>데이터 시각화</li> </ul>	2020-03-02	2020-04-03	20
김명수	<ul style="list-style-type: none"> <li>AWS EC2를 이용한 웹 서버 구축</li> </ul>	2020-03-02	2020-04-14	40
김선필 윤찬우	<ul style="list-style-type: none"> <li>Camera</li> <li>Object Detection</li> <li>Number Recognition</li> <li>Navigation System 개발 및 관리</li> </ul>	2020-04-03	2020-05-31	60
배한울 김다훈	<ul style="list-style-type: none"> <li>SLAM</li> <li>최단경로 알고리즘</li> <li>localization</li> <li>센서 최적화</li> <li>Navigation System 개발 및 관리</li> </ul>	2020-04-03	2020-05-31	60

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

김명수	<ul style="list-style-type: none"> <li>Application</li> <li>- ionic 빌드</li> <li>- ios 환경 구축</li> <li>- android 환경 구축</li> </ul>	2020-04-03	2020-05-31	60
-----	---------------------------------------------------------------------------------------------------------------------------------	------------	------------	----


## 6.4 비 인적자원 투입계획

항목	Provider	시작일	종료일	Required Options
컴파일러	GNU, Clang	2020-03-01	2009-05-31	
개발용 PC 5대	Samsung, Apple, Lenovo	2020-03-01	2020-05-31	
임베디드 보드(Tx2)	NVIDIA	2020-03-01	2020-05-31	
임베디드 보드(Open CR)	ROBOTIS	2020-03-01	2020-05-31	
3D 프린터	신도리코	2020-03-01	2020-05-31	
Turtlebot3	ROBOTIS	2020-03-01	2020-05-31	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

## 7. 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	Web	Turtlebot3 e-Manual	ROBOTIS		표윤석	
2	기사	정부, 안내로봇 KS규격 제정 검토...수요 형성 '기대반 우려반'	<a href="https://m.etnews.com/20200323000169">https://m.etnews.com/20200323000169</a>	2020.03.23	변상근	
3	기사	인천국제공항, 안내 로봇 '에어스타' 본격 운영	로봇신문사	2018.07.11	정원영	
4	Web	OpenSlam	OpenSlam			
5	기사	라이다의 원리와 장단점, 구현 방식에 따른 종류	TechWorld	2019.01.24	양대규	
6	논문	YOLO:Real-Time Object Detection	University of Washington	2016.05.09	Joseph Redmon	
7	논문	ROS를 활용한 서빙 이동로봇 구현	배재대학교	2019.02.10	김의선	
8	Web	딥러닝을 이용한 숫자 이미지 인식	조대협 블로그	2017.01.09	조대협	
9	Web	ros-sensor-fusion	methyDragon	2019.03.16	methyDragon	
10	Web	Fusion 360 tutorial for absolute beginners	autodesk	2019.09.06	Product Design Online	
11	세미나	오픈 로보틱스 세미나	ROBOTIS	2014.12.21	표윤석	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	수행계획서		
	프로젝트 명	OLAF	
	팀 명	ELSA	
	Confidential Restricted	Version 1.9	2020-MAR-20

12	서적	칼만필터는 어렵지 않아	한빛아카데미	2019.05.20	김성필	
13	Web	Xycar-A2 pdf	Xytron		허성민	
14	Web	Naver Labs' On road Intelligence	Naver			
15	논문	실내 환경에서의 이동로봇의 위치추정을 위한 카메라 센서 네트워크 기반의 실내 위치 확인 시스템	University of Tokyo	2016.04.16	지용훈	