

비대면 시스템 구현을 위한 자율주행 서빙 로봇

홍정훈*, 송창의*, 김대관*, 이근호*

*광운대학교 로봇학부

jhhong2501gmail.com, sltidi123@gmail.com, kdggdk7984@naver.com,

rmsg0726@naver.com

Automobile Serving Robot for Untact System

Jeoun-Hun Hong*, Chang-Ui Song*, Dae-Gwan Kim*, Geun-Ho Lee*

*Dept. of Robotics, Kwang-Woon University

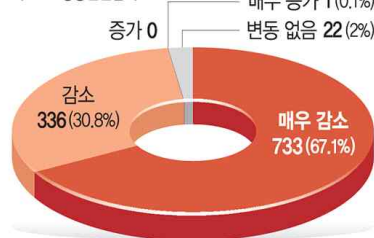
요약

최근 코로나 사태로 인하여 한국에서는 사람 간에 거리를 두는 사회적 거리 두기와 생활 속 거리 두기를 시행하고 있다. 또 식당에서의 지출 중 가장 큰 비율을 차지하고 있다. 인건비를 절감하기 위해 본 연구는 2D Lidar 센서를 사용하여 식당에서의 사람과의 접촉이 가장 큰 서빙의 역할을 로봇이 직접 하는 것을 목표로 하였다. Lidar 센서를 이용해서 SLAM을 구현하였고 안드로이드 앱을 사용하여 주문 결제를, Four Bar Linkage를 사용하여 음식 전달 부분을 구현하였다.

1. 서론

2020년 1월 20일 한국에서 첫 코로나 19 확진자가 나온 이후 감염 예방을 위하여 외출을 자제하는 사회적 거리 두기가 시행되었다.

우한 폐렴 발생 이후 소상공인 매출액 변화
(단위: 명) ※ 소상공인 1092명 답변, () 안은 비율
자료: 소상공인연합회



코로나 19로 인한 사회적 거리 두기가 시행된 이후로 사람과 사람과의 대면이 이루어지는 대부분의 매장에서의 매출은 매우 감소하였고 폐업하는 가게들이 속출하게 되었다.

최저임금이 생존 외식업체에 어떤 충격			업소면적에 따른 폐업률	
2017년	2018년		33㎡ 이하	38.9%
1.7명	종업원 수	1.5명	66㎡ 이하	30.7%
207만원	1인당 인건비	279만원	66㎡ 초과	26.3%
352만원	월평균 총인건비	418만원		
35.4%	영업비용 대비 인건비 비중	40.5%		

*2018년 생존 외식업체 275곳 분석, 자료=한국외식업중앙회 산하 한국외식산업연구원
*최근 1년 새 폐업된 125개 외식업체 분석, 자료=한국외식업중앙회 산하 한국외식산업연구원

MBN 매일경제에서 보도한 한국외식산업연구원의 2018년도 조사에 따르면 가게 운영비에 있어서 인건비가 가장 큰 비중을 차지하고 있고 폐업한 외식업체 125곳의 공통점은 인건비 부담이 높았다고 한다. 2017년 10월 기준으로 살아남은 외식업체 275곳의 2017년 영업비용 대비 인건비 비중은 35.4%, 매출액 대비 인건비 비중은 23.8%에 달했던 반면 2018년도에는 영업비용 대비 인건비 비중은 41.3% 매출액 대비 인건비 비중은 26.4%로 급증함을 확인할 수 있었다.

연도별 최저임금 결정현황



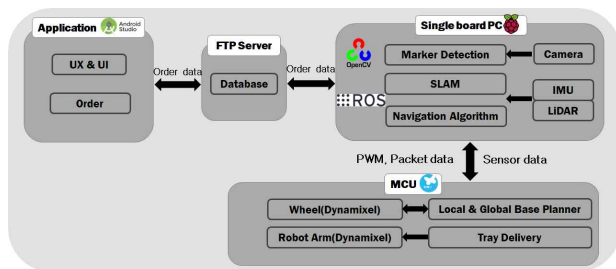
최저임금위원회에서 제공한 연도별 최저임금 그래프를 확인해보면 우상향 그래프를 이루고 있고 정부에서도 최저임금 만 원 이상을 목표로 계속 증가함에 따라 앞으로의 인건비 부담 역시 증가할 것이다.

이에 본 연구는 대인과의 접촉을 최소화하고 인건비의 부담을 경감시킬 수 있도록 Lidar를 사용하여 다양한 매장 구조에 적용할 수 있도록 SLAM 기술을 사용하여 실내

매장의 지도를 생성하고, 안드로이드 앱을 제작하여 비대면 주문 결제 시스템을 구축, 매장 내에서 자율주행으로 음식을 테이블로 이동 후 Four Bar Linkage 메커니즘을 사용하여 로봇이 직접 테이블로 음식을 전달하는 시스템을 목표로 한다.

2. 본론

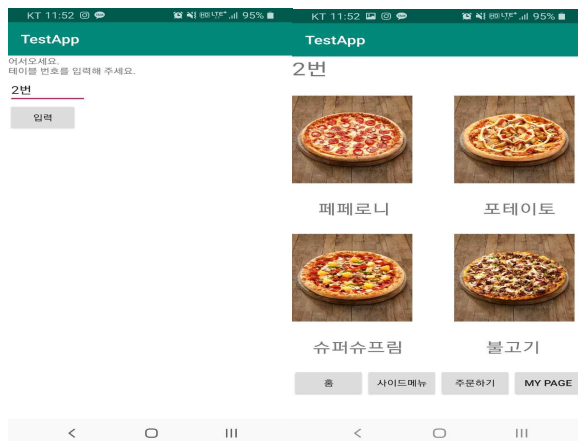
그림 1은 System Architecture이다. 테이블에서 앱을 통하여 음식을 주문하고 서빙 로봇 위에 음식을 놓으면 SLAM 기술로 자율주행하여 테이블까지 장애물을 피하여 도착한 후 음식을 테이블 위에 올려놓고 원래 위치로 돌아오는 것을 목표로 한다.



(그림 1) System Architecture

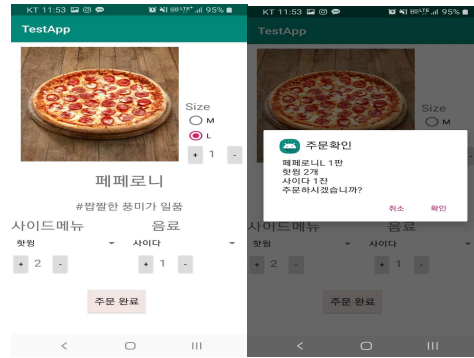
2.1 주문 및 결제 앱 제작 [1]

앱을 사용하여 주문 결제를 하기 위해 Client의 주문처리 및 주문한 내용의 DB기록 그리고 주문내용에 대한 푸시 메시지를 통해 Owner에게의 내용전달을 구현하고자 한다. 앱 개발은 안드로이드 스튜디오를 이용하였고 언어는 자바를 채택하였다.



(그림 2) 테이블 번호 선정 및 메인 메뉴 주문

Client가 주문을 할 때 본인이 앉은 테이블의 번호를 입력하면 메뉴선택 화면으로 진입한다. 메인메뉴를 선택하는 화면이므로 직관적으로 파악할 수 있게 사진과 함께 구성하였다.



(그림 3) 사이즈, 수량, 사이드 메뉴 주문 및 주문 확인

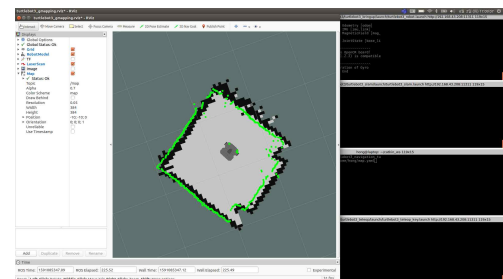
메인메뉴를 선택하면 사이즈 및 수량 그리고 사이드메뉴와 음료를 선택할 수 있는 창으로 진입한다. 주문 할 내용들을 선택한 후에 주문완료 버튼을 선택하여 주문내용을 확인시켜 준 이후에 주문이 완료된다.

SEAT	PIZZA	PIZZAnum	SIDE	SIDEnum	BEV	BEVnum	SIZE
NULL	포테이토	1	핫윙	1	사이다	1	M
NULL	포테이토	1	핫윙	1	사이다	1	M
1번	페페로니	2	핫윙	1	콜라	2	M
NULL	페페로니	2	치즈오브스파게티	2	사이다	1	M
2번	페페로니	2	치즈오브스파게티	2	사이다	1	M
2번	0	2	0	1	0	1	L
2번	페페로니	2	치즈오브스파게티	1	사이다	1	L
2번	페페로니	1	치즈스틱	2	사이다	1	L
3번	불고기	1	핫윙	1	맥주	1	M
4번	슈퍼슈프림	1	치즈오브스파게티	3	사이다	1	M
2번	페페로니	1	핫윙	2	사이다	1	L

(그림 4) 서버 기록

위에서 주문한 내용이 서버에 기록되는 것을 확인할 수 있다.

2.2 SLAM



(그림 5) 작업 공간 Mapping

SLAM(Simultaneous Localization And Mapping)은 동시적으로 위치 추정과 지도 작성하는 것을 의미한다. Lidar 센서로부터 극 좌표계값과 각도 값을 받아 직교 좌표계로 변환하여 2D 지도를 생성한다.

2.3 2D Lidar

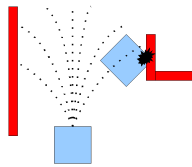
SLAM을 하기 위하여 ROBOTIS 사의 360 LDS-01(Laser Distance Sensor) 제품을 사용하였다. LDS-01은 360도 측정이 가능한 2차원 레이저 스캐너로 로봇 주위의 거리 정보를 수집하여 SLAM을 할 수 있도록 한다. Distance Range는 120~3,500mm를 측정할 수 있고 또한 360도 회전 범위를 갖고 Angular Resolution은 1°를 갖는다. Lidar 센서를 통해 극좌표

(r, θ) 를 얻은 다음 직교 좌표계로 변환하여 (x, y) 값을 직교 좌표계에 그 좌표를 영상으로 출력하여 지도를 만든다.

2.4 Pillar Filtering [2]

쟁반을 전달하기 위해 로봇 상단부에 Four Bar Linkage가 필요하다. SLAM을 하기 위한 LIDAR 센서 주변에 Four Bar Linkage를 연결할 기둥이 맵핑에 방해가 되므로 특정거리에 위치한 장애물들을 인식하지 않는 소스를 추가해야 한다.

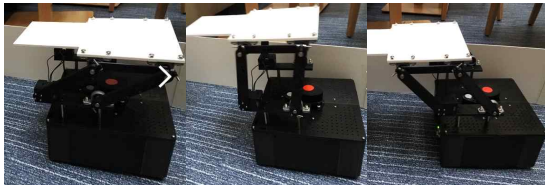
2.5 DWA(Dynamic Window Approach) Algorithm [3]



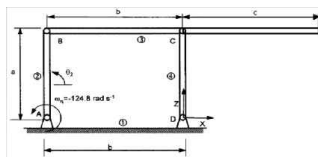
(그림 6) DWA Algorithm Concept picture

DWA 알고리즘의 컨셉은 다음과 같다. 로봇제어공간에서 이산적으로 샘플링하여 $(dx, dy, d\theta)$ 값을 측정한다. 짧은 시간 동안 샘플링한 각각의 속도를 로봇에 적용시켜 현재 로봇의 위치로부터 시뮬레이션을 수행하였을 때 어떠한 결과가 나오는지 예측하고 각각의 궤적에 점수를 매긴다. 장애물이나 벽에 충돌하는 결과값은 제거하고 타겟에 근접한 정도, 장애물에 근접한 정도, 범용 경로에 근접한 정도등을 점수 매겨 가장 높은 점수를 받은 궤적을 선택하여 주행한다. 그리고 이 과정을 반복함으로서 주행한다.

2.6 Parallel Four Bar Linkage [4], [5]



(그림 7) Four Bar Linkage Scenario



(그림 8) Four Bar Linkage Design

쟁반을 테이블과 평행하게 위로 전달하기 위하여 적용한 메커니즘이다. 단순한 동작으로 테이블 위로 쟁반을 놓기 위하여 일반 쟁반과는 다르게 받침으로 살짝 떠 있도록 제작한 쟁반이 사용되었다. 길이 $a = 15\text{cm}$, $b = 12\text{cm}$, $c = 15\text{cm}$ 로 제작하였고 축 D로부터 c에 해당하는 전달부의 작업 반경은 축의 각도의 범위가 $(30^\circ \leq \theta_2 \leq 120^\circ)$ 이므로 $(-10.4\text{cm} \leq x \leq 10.4\text{cm}, 6\text{cm} \leq z \leq 12\text{cm})$ 가 나오게 된다.

2.7 Camera Pose Estimation using ArUco Marker [6]

Camera Calibration은 투영된 영상을 통해 카메라 내부 매개 변수(Intrinsic Parameter: 초점거리, 주점, 비대칭계수)와 영상 왜곡 행렬(Distortion Matrix)을 얻는 과정, 다시 말해 2D 영상 좌표와 3D 공간좌표의 변환관계를 얻는 것을 뜻한다. 본 프로젝트에서는 ROS의 Camera_Calibration 패키지를 이용하여 그림 11과 같이 각 파라미터 행렬들을 얻었다.

ArUco Marker는 패턴별 ID 정보를 가지고 있는 marker이다. marker의 크기 정보와 패턴 ID 정보 그리고 사용하고 있는 카메라의 Calibration 정보를 알고 있을 때, 카메라 영상에서의 marker 위치 및 자세 정보를 얻을 수 있다.

본 프로젝트에서는 marker를 인식하기 위해 OpenCV에서 제공하는 Library를 사용하였다. 먼저, 영상을 GrayScale 영상으로 변환하고 Canny Edge Detection 방법으로 외곽선 추출 영상을 얻는다. 그 후 일정 크기 범위의 사각형을 얻어내고 Homography 함수를 통해 marker의 정면 정사각형 영상을 얻을 수 있다. 그 정사각형 marker 영상을 격자로 나누면 검정색과 흰색 패턴을 얻게 되는데, 그 패턴을 가지고 2진수 ID를 알아낼 수 있다. 그 2진수 ID를 찾으려는 ID 패턴 데이터와 비교해서 영상에서의 marker ID를 알아낼 수 있다.

그리고 해당 패턴 모양을 알기 때문에 Homography 하기 전 영상과 후 영상을 비교해서 카메라 기준 marker의 회전 벡터를 얻을 수 있다. 이와 비슷한 방법으로, 영상에서의 마커 중심의 좌표, marker의 크기를 통해 marker의 위치 벡터를 얻을 수 있다. 얻은 marker의 회전 및 위치 벡터와 Rodrigues 회전 공식을 이용하면 marker 기준 카메라의 위치 벡터도 알 수 있게 된다.

본 프로젝트에서는 이렇게 얻게 된 위치, 회전 벡터를 이용하여 marker에 대한 robot의 정확한 위치 및 자세 제어를 할 수 있게 된다.



(그림 9) Marker & Camera Pose Estimation

3. 실험 결과



(그림 10) 서빙 로봇의 초기 위치, 주행 시작

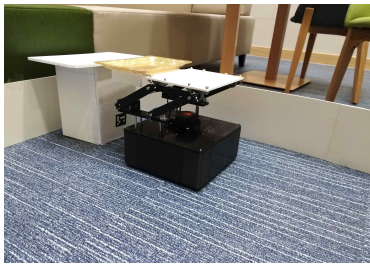
음식이 나오기 전에는 서빙 로봇이 초기 위치에서 대기하고 있다. 쟁반이 Four Bar Linkage 위에 올라가면 앱에서 주문한 위치로 주행을 시작한다.



(그림 11) 테이블 앞 좌표에 정지

그림 11를 보면 로봇이 지정된 좌표 근처인 테이블 앞에 정지하였지만 정확한 쟁반을 내려놓을 수 있을 만큼 정밀하게 이동하지 못했음을 확인할 수 있다.

서빙 로봇을 테이블과 수직 하도록 위치시키기 위해 현재 로봇 위치와 테이블 사이의 x , z , pitch 값을 받아 정밀 조정함으로 그림 11과 같이 Pitch 회전 후 x 만큼 직진 90도 회전 z 만큼 직진하여 테이블과 수직 하게 이동했음을 확인할 수 있었다.



(그림 12) Pose Estimation으로 위치 및 자세 미세 조정

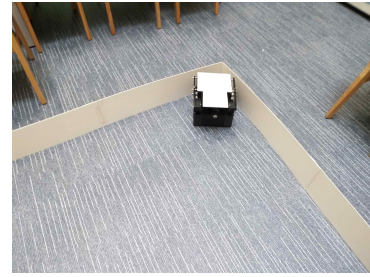


(그림 13) Four Bar Linkage Tray Transfer

그림 13은 테이블과 서빙 로봇 사이의 위치 조정이 끝나면 Four Bar Linkage를 속도제어, 위치제어로 쟁반을 테이블 위에 올려놓고 후진 후 다시 처음 각도로 위치시킨다.



(그림 14) 초기 위치로 귀환



(그림 15) 초기 위치로 귀환

그림 14을 보면 서빙 로봇이 초기 위치로 돌아간 후 다음 음식을 기다리는 것을 확인할 수 있다. 서빙 로봇이 초기 위치로 돌아갔지만 시작할 때와는 다르게 각도가 틀어져 있는 이유는 ArUco Marker를 사용하지 않았기 때문에 시작할 때와는 각도가 다르게 정지해있음을 확인할 수 있다.

4. 결론

본 연구에서는 2차원 Lidar, DWA Algorithm, ArUco Marker, Four Bar Linkage, 안드로이드 어플리케이션을 이용하여, 서빙 로봇 시스템을 구현하였다. 코로나 19로 인해 실험을 진행할 수 있는 장소가 한정적이었고, 협소하였기 때문에 실제 테이블에 맞는 크기가 아닌 작은 스케일로 제작하였다. 로봇을 원격으로 제어하기 위하여 리눅스 운영체제와 서빙 로봇을 ssh 접속을 사용하여 로봇을 제어하였지만 불안정한 네트워크 상태로 인해 로봇의 움직임이 딱딱 끊기는 현상이 있었다. DWA 알고리즘을 사용하여 테이블 근처 좌표까지 이동시켰을 때 테이블 위치까지는 이동을 잘 했으나 각도를 잘 잡지 못하는 현상이 빈번하게 있었지만 ArUco Marker를 사용하여 비교적 정확하게 위치시킬 수 있었다.

[본 논문은 과학기술정보통신부
정보통신창의인재양성사업의 지원을 통해 수행한
ICT멘토링 프로젝트의 결과물입니다]

참고문헌

- [1] 정재곤 “Do it! 안드로이드 앱 프로그래밍 (7판)” pp. 62-67, 2019
- [2] 표윤석, 조한철 외 2인, “ROS 로봇 프로그래밍”, 루비페이퍼, pp. 385-387, 2017.
- [3] ROS WIKI, DWA Local Planner “http://wiki.ros.org/dwa_local_planner”
- [4] 김상훈, Motor Control - DC, AC, BLDC”, 복두, pp. 70-103, 2016
- [5] John J. Craig, “Introduction to Robotics Mechanics and Control 3rd”, Pearson, pp. 41-53, 2009
- [6] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion”, Department of Computing and Numerical Analysis, University of Córdoba, June 2014, pp. 3-7