

식당 홀의 완전 자동화를 위한 테이블 로봇 시스템

Table Robot System for Complete Restaurant Hall Automation

○양 지 윤¹, 권 근 우¹, 김 다 현¹, 김 정 목¹, 류 태 훈¹, 배 승 학¹, 김 완 수^{2*}

¹⁾ 한양대학교 ERICA 로봇공학과 (TEL: [REDACTED]; E-mail: wldbs2206@gmail.com)

Abstract This paper introduces a Table Robot System aimed at fully automating restaurant service. Utilizing ceiling-mounted cameras, the system enhances object detection and dynamic path planning algorithm, thus reducing labor costs and improving efficiency. Initial findings show significant increases in service speed and reductions in human server dependency, highlighting the system's potential to revolutionize restaurant operations.

Keywords Restaurant automation, Table robot system, Object detection, Obstacle recognition, Dynamic path planning.

1. 서론

세계 로봇 연맹(IFR)이 발표한 보고서(World Robotics 2023)에 따르면 전문 서비스 로봇의 수요는 15.8만 대로 전년 대비 48% 증가하였다 [1]. 서비스 로봇 중 최근 푸드테크 트렌드가 주목을 받으며 레스토랑 내의 서빙 로봇이 널리 운용되고 있다.

하지만 현재 서비스 중인 서빙 로봇은 음식을 서빙하는 주 업무만을 가질 뿐, 테이블 위 식기를 치우거나, 테이블을 옮기는 등의 작업은 할 수 없다. 결국, 서빙의 작업으로 인해 홀인원의 고용이 필수적이다. 이 경우, 인건비 감축에 대한 효율성이 감소하고, 로봇과 사람이 한 공간에서 작업이 진행되면서 불가피 하게 동선이 겹치는 문제가 발생한다. 따라서 홀인원이 필요하지 않은 서빙 로봇 시스템에 대한 필요성을 인지하였고 기존 시스템의 개선 방안을 제시하려고 한다.

본 논문은 식당 홀의 완전 자동화 시스템인 ‘테이블 로봇 시스템’ 제안 및 검증에 위한 시뮬레이션 구현을 중점적으로 작성하였다. 해당 시스템은 테이블 자체가 로봇이 되어 음식 서빙, 테이블 자리 제공, 청소 공간으로의 이동이 가능하게 한 시스템이다. 또한, 테이블 로봇은 원가 절감을 위해 자체 내장 센서를 줄이고, 식당 천장에 부착된 카메라를 통해 로봇 주변을 인식하여 제어하는 것에 목표를 둔다.

2. 본문

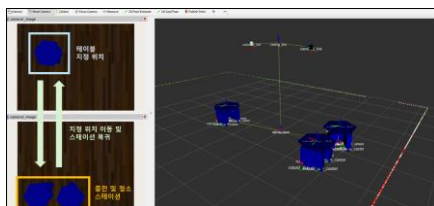


그림 1. 테이블 로봇 시스템

‘테이블 로봇 시스템’(그림 1)의 기본 프로세스는 로

봇이 충전 및 청소 스테이션에서 손님이 지정한 위치로 이동한 후 복귀하는 과정이 필요하다. 이를 위해 로봇은 자신의 위치를 파악하고 주변 장애물을 인식하여 회피 및 이동하는 제어가 필요하다.

해당 프로세스의 시뮬레이션은 테이블 로봇 3대를 이용하여 진행한다. RGB 정보로 로봇을 인식하고, Depth 정보로 장애물을 인식하기 위해 RGBD 카메라를 일정 높이에서 바닥을 바라보는 방향으로 설치한다. 카메라가 바라본 바닥 면을 x, y 평면으로 가정하여 일정한 크기의 그리드를 가진 가상의 맵을 생성한다. 해당 그리드에는 인식한 장애물과 로봇별 위치를 나타내어, 목표 지점으로 이동할 수 있는 맵 정보를 제공한다. 맵의 각 그리드 크기는, Depth 이미지 프레임을 640*480으로 나누어 설정한다.

카메라 인식을 활용하여 도출된 그리드 맵을 Global Planner인 D* Lite에 사용한다. D* Lite에서 Path를 생성하고 Local Planner인 Admittance 제어기에 전달하여 경로 추종을 수행한다. 이때 Admittance 제어기를 사용하는 이유는 식당 환경이라는 동적 환경에서 순간적인 회피 기동 가능해야 하므로 Admittance 제어를 추가하여 Path를 추종하면서 동시에 동적 장애물에 대해 순간적인 대응도 가능하도록 하였다.

1. Robot Detection

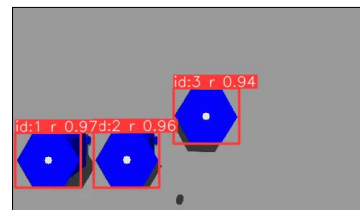
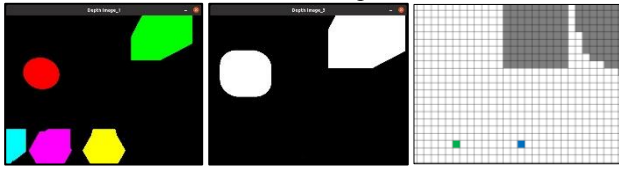


그림 2. 테이블 로봇 Object Detection.

카메라로부터 로봇을 감지하는 기술은 Object Detection을 사용한다. 로봇의 커스텀 데이터를 생성한

후 yolov8n.pt를 통해 학습한다. 학습한 데이터를 이용하여 로봇을 Object Detection하고, Bounding Box의 중심 좌표값을 (x, y) 좌표로 추출한다. 각 로봇을 구별하기 위해 bytetrack.yaml 사용하여 Multi Object Tracking으로 로봇별 ID를 지정한다. 위치 좌표값의 정확도를 높이기 위해 로봇 자체적으로 Localization 작업을 진행한다. 해당 작업은 로봇에 내장된 IMU, LiDAR, Encoder 센서에서 받아온 데이터를 ICP(Iterative Closest Point) 알고리즘[2]과 EKF(Extended Kalman Filter)를 사용하여 수행된다. Localization을 통해 얻은 좌표와 비전 인식을 통해 얻은 좌표값을 Linear Interpolation 하여 각 로봇별 좌표 도출한다. 그 후, 각 로봇의 좌표에 해당하는 그리드의 행과 열 번호를 반환한다.

2. Obstacle Detection & Grid Map Generate



(a) 전 (b) 후 (c) 시각화 그림
그림 3. 그리드 화 (a)전 (b)후 (c)시각화 그림

그리드 맵에서는 장애물의 위치를 1, 비장애물을 0으로 저장한다. 장애물 인식을 위해 카메라를 통해 받아온 Depth 값으로 바닥면과 카메라의 거리보다 짧은 거리인 행렬 인덱스 번호를 파악한다. Depth 행렬과 그리드 맵의 크기는 같으므로 해당 인덱스에 1 값을 저장한다. 그 후, 장애물 별로 그룹화를 진행한다. 그룹화된 장애물 중 실제 장애물이 아닌 로봇도 포함되므로, 이를 구별시키는 작업이 필요하다. 따라서 로봇 인식에서 반환한 로봇 위치 좌표를 포함하는 그룹 그리드를 0으로 변환하여 장애물만을 표현한 그리드 맵을 도출한다. 줄어든 로봇 사이즈를 고려하여 장애물 크기를 모폴로지 연산을 통해 형태학적으로 확장하는 작업을 추가로 진행한다. (그림3-b)

위의 과정으로 도출된 장애물이 표현된 그리드 맵에 로봇 위치를 ID 별로 2,3,4로 나타낸다. 마지막으로 장애물과 로봇이 모두 표현된 그리드 맵을 반환한다.

3. Global Planner

식당 내부에서 손님이 원하는 위치에 테이블을 배치하기 위해 글로벌 경로 계획을 진행한다. 이때 동적 환경에 적합한 D* Lite[3]를 활용하여 노드 단위 경로를 생성한다. 로봇별로 경로를 생성해야 하므로 비전 작업을 통해 전달받은 Grid Map에서 제어하고자 하는 로봇을 제외한 다른 로봇 또한 장애물로 판단하도록 맵을 수정하여 로봇별로 경로를 생성한다.

4. Local Planner

글로벌 경로 계획을 통해 생성된 경로를 추종하기 위해 Admittance 제어를 활용하였다. 또한, 장애물이 갑자기 나타났을 경우 비전 작업에서 그리드 맵이 업

데이트되는 동안 D* Lite에서 경로 수정이 없으므로 D* Lite만으로 부족한 순간적인 회피 기동이 가능하다. 일반적인 Admittance 제어는 외부 힘에 순응하도록 로봇에게 원하는 위치, 속도, 가속도를 가상 객체의 운동 방정식(1)을 사용하여 계산하는 제어 방식이다.

$$M_d(\ddot{x}_d - \ddot{x}_0) + D_d(\dot{x}_d - \dot{x}_0) + K_d(x_d - x_0) = F_{ext} \quad (1)$$

이때 현 시스템에서는 장애물에 대한 외부 힘 F_{ext} 은 Lidar에서 측정된 장애물이 일정한 거리에 들어왔을 때 수식(2)를 활용하여 구한다.

$$F_{ext} = \begin{bmatrix} F_{ext}[x] = -F_{size} \cdot \cos(obstacle_{degree}) \\ F_{ext}[y] = -F_{size} \cdot \sin(obstacle_{degree}) \\ 0 \end{bmatrix} \quad (2)$$

수식(2)의 $obstacle_{degree}$ 는 로봇과 가장 가까운 장애물의 각도이고, F_{size} 는 Lidar로 측정된 로봇과 장애물까지의 거리와 Min-Max 정규화 알고리즘을 활용해 구한 힘의 크기이다. 수식(2)를 활용해 계산한 F_{ext} 을 가상의 Mass-Spring-Damper 시스템 수식(3)을 활용하여 로봇의 가속도를 구한다. 따라서 F_{ext} 가 없으면 경로를 추종하고 일정 거리 안에 장애물이 들어와 F_{ext} 가 발생하면 회피기동 수행한다.

$$\ddot{x} = M_d^{-1} \cdot \{K_d(x_d - x) - D_d\dot{x} + F_{ext}\} \quad (3)$$

$$\dot{x} = \dot{x} + \ddot{x} \cdot \Delta t$$

앞의 기술을 통해 스테이션에 위치한 3대의 로봇이 랜덤하게 장애물이 생성되는 환경에서 지정 위치로 이동 및 스테이션으로의 복귀를 확인하였다.

3. 결론

본 논문에서는 홀인원이 필요치 않은, 식당 홀의 완전 자동화를 위해 테이블 자체가 로봇이 되어 홀을 관장하는 ‘테이블 로봇 시스템’을 제안 및 시뮬레이션 상의 구현을 작성하였다. 시뮬레이션을 통해 로봇이 서빙 외에도 테이블을 형성, 스테이션으로의 이동 및 복귀함으로써 식당 홀의 완전 자동화 가능성을 확인하였다.

하지만 해당 시스템이 현실에서 식당 회전을, 인건비 감축에 얼마큼의 효율을 상승시키는데 대한 추가적인 실험을 통해 수치적인 검증이 필요하다. 따라서 본 연구는 완전 자동화 식당의 실사 제작에 토대를 마련하는데 기여할 것 기대하며, 향후 추가적인 연구와 실험을 통해 자동화 식당 시스템 구현 및 실효성을 파악할 예정이다.

참고문헌

- [1] S. International Federation of Robotics, “World Robotics 2023”, pages 37, 2023.
- [2] Zhang, Z. “Iterative Closest Point (ICP)”. In: Ikeuchi, K. (eds) Computer Vision. Springer, Boston, MA. 2014.
- [3] S. Koenig and M. Likhachev. “D* Lite” in Proceedings of the AAAI Conference of Artificial Intelligence(AAAI), pages 476-483, 2002.