

카트 부착 스마트 모듈형 로봇

정희철*, 손영우*, 김은호*, 김탁윤*, 문재현**

*광운대학교 로봇학부, **한국기술거래사회

7855hckka@naver.com, omlijy9603@gmail.com, dmsghdmstj@gmail.com, nicehappykty@naver.com,
smjhoon@gmail.com

**교신저자(Corresponding Author)

Smart modular robot with cart attached using AI algorithm

Hee-cheol Jeong*, Young-woo Son*, Eun-Ho Kim*, Tak-Yun Kim*, Jae-Hyun Moon**
*Dept. of Robotics, Kwangwoon University, **Korea Technology Transfer Agent Association

요 약

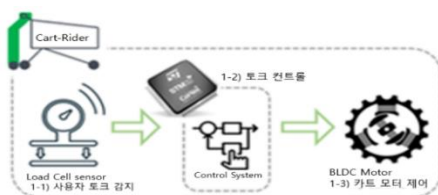
쇼핑카트 부착 모듈형 로봇 ‘Cart-Rider’는 어드미턴스 제어를 통한 사용자의 힘 보조 기능, 딥러닝을 활용한 네비게이션 기능, GPS를 활용한 도난 방지 기능을 제공하는 로봇으로 대형 마트에서 발생하는 안전사고 및 쇼핑카트 도난을 예방하는 동시에 사용자에게 편의성을 제공하는 로봇이다. 또한 여러 대를 겹쳐서 보관하는 기존의 카트 시스템을 유지하고 탈부착이 용이하도록 하드웨어를 제작하여 환경에 영향을 주지 않고 유지 및 보수가 용이하도록 제작했다.

1. 서론

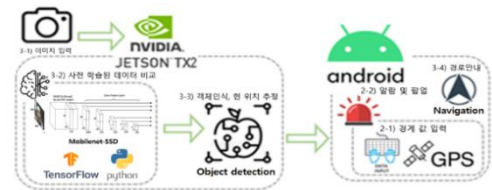
최근, 코로나가 지속되면서 언택트에 가장 맞는 AI와 로봇의 시장 점유율이 증가함에 따라 쇼핑에도 도움을 줄 수 있는 스마트 카트들이 개발되기 시작했다. 몇 대형마트들의 스마트 카트들의 기능은 일반적인 소비자들을 위해 개발되었고, 이를 보완하기 위해 새로운 카트 부착형 스마트 모듈을 고안하게 됐다. 첫째, 많은 적재량은 하중을 늘게 하여 장보기 약자가 카트 운전을 어렵게 만든다. 힘 보조 기능으로 손쉽게 누구나 밀 수 있도록 하려고 한다. 마트의 큰 경제적 손실로 이어지는 카트 도난 문제를 도난 방지 기능을 통해 해결하고자 한다. 마지막으로 고객들의 편의성을 증진시키기 위하여 태블릿을 통해 현 위치 파악 및 목적지까지 경로를 안내하는 네비게이션 기능을 통해 편리한 쇼핑을 도모한다. 이를 바탕으로 쇼핑관련 서비스를 제공할 모듈형 로봇을 제작했다.

2. 제안시스템

2.1 시스템구성도



(그림 1) 시스템 구성도 1



(그림 2) 시스템 구성도 2

하드웨어는 로봇으로 인한 추가적인 인프라 구축을 최소화하는 방향으로 설계했다. 3d 프린터로 제작된 2개의 손잡이 커버로 인해 기존의 카트 손잡이에 탈부착이 가능하며 사용하지 않을 시에는 손쉬운 보관 및 운송이 가능하다. 또한 기존의 쇼핑카트는 공간적 효율성을 위해 카트의 뒷부분을 유동적으로 설계하여 서로 겹쳐서 보관할 수 있다는 장점을 계속 유지할 수 있도록 부착형 모듈로 제작하되, 카트와 같이 겹쳐도 기존의 장점을 없애지 않는 방향으로 제작했다.

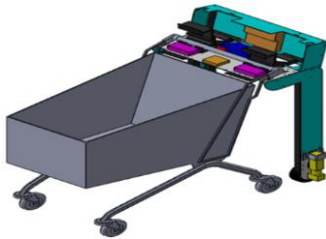
(그림 1)의 시스템 구성도와 같이 사용자의 토크를 감지하여 사용자의 힘을 보조해줄 수 있도록 토크 컨트롤 및 모터제어를 한다. 이 과정을 통해 사용자는 큰 힘을 들이지 않고도 카트를 쉽게 움직일 수 있다.

(그림 2)의 시스템 구성도에서 카메라로부터 입력 받은 이미지로 딥러닝 모델과 앱(안드로이드)을 거쳐 사용자의 현 위치를 추정하며 동시에 사용자가 입력한 목적지까지의 경로를 나타낸다. 이를 통해 사용자가 효율적인 쇼핑을 할 수 있도록 돕는다. 또한, GPS

를 이용하여 사전에 등록된 경계 값과 실시간 비교를 통하여 카트의 도난을 방지한다.

2.2 하드웨어 설계

하드웨어는 여러 대를 겹쳐서 보관하는 현재의 카트 보관 시스템과 주변 환경을 고려하고 탈부착이 가능한 구조로 (그림 3)과 같이 고안됐다. 로봇으로 인한 추가적인 인프라 구축을 최소화하는 방향으로 설계했다. 제작된 2개의 손잡이 커버로 인해 쇼핑카트 손잡이에 탈부착이 가능하며 사용하지 않을 시에는 손쉬운 보관 및 운송이 가능하다. 또한 기존의 쇼핑카트는 공간적 효율성을 위해 겹쳐서 보관하는데 이와 같은 카트 보관 시스템에 자연스럽게 융화될 수 있도록 로봇의 하드웨어는 디글자 형태로 설계했다. 기본 프레임은 프로파일 20x20과 알루미늄으로 틀을 잡고 아크릴로 외관을 형성했다. 로드셀을 통해 사용자의 주행 의도를 측정하기 위한 손잡이는 기존의 바 형태에서 개별 제어를 하기 위해 양쪽의 두개로 분리하여 설치했다. 또한 모터가 연결되어 있는 로봇의 하부와 쇼핑카트 사이에 추가적인 지지대를 설치해 바닥의 마찰력이나 외력에도 안정적으로 주행할 수 있도록 설계했다.



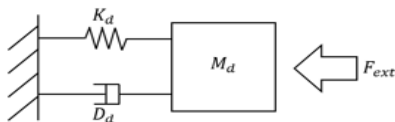
(그림 3) 하드웨어 사진

2.3 제어기 설계

1) 힘 보조기능

힘 보조 기능을 위해 사용자와 로봇의 상호작용 제어 기술을 사용했다. 상호간에 적용되는 힘을 제어하는 기술이며 물체와 대상과의 동적 관계인 임피던스를 고려하여 위치 기반 임피던스 제어 기법인 어드미턴스 제어를 적용했다.

로봇 시스템의 어드미턴스 모델링을 구하기 위해서 시스템이 갖고 있는 역학적인 파라미터들의 표현을 (그림 4)로 표현했다. 로봇과 외부 환경 사이 관계를 free space에서의 어드미턴스 모델로 나타냈다.



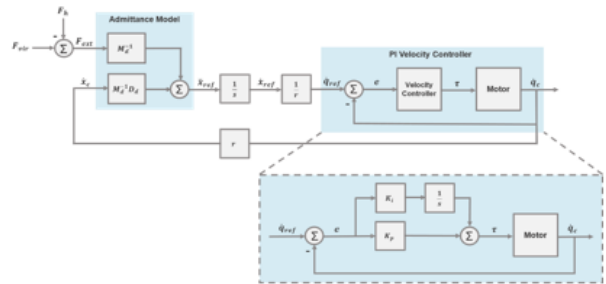
(그림 4) 어드미턴스 모델

위와 같은 모델을 운동방정식으로 나타내면 다음과 같다.

$$-F_h = M_d(\ddot{x}_0 - \ddot{x}) + D_d(\dot{x}_0 - \dot{x}) + K_d(x_0 - x) \quad (1)$$

손잡이와 사용자의 손이 항상 접촉 상태에 있기 때문에 목표 가속도, 속도, 위치는 0으로 두고 사용자가 미는 힘을 최소화 하기 위하여 목표 힘 역시 0으로 두어 contact space에서의 어드미턴스 환경을 만들어 로봇이 원하는 질량 댐핑 성질을 갖도록 하는 가속도를 다음 식으로 도출할 수 있다.

$$\ddot{x}_{ref} = (F_h - D_d \dot{x}) / M_d \quad (2)$$



(그림 5) 어드미턴스 제어기

어드미턴스 제어기는 (그림 5)을 통해서 흐름을 보여준다. 사용자가 손잡이를 밀거나 당기면 손잡이에 부착된 로드셀 센서를 통해 힘을 측정한다. 측정된 힘은 어드미턴스 모델을 통해서 로봇이 구현해야 할 목표 가속도를 도출한다. 목표 가속도는 속도로 변환 후 속도 제어기를 통해서 로봇이 목표 속도로 움직이도록 제어한다.

2.4 소프트웨어 구성 및 기능

1) 도난 방지 기능

Convex Hull algorithm이란 다양한 객체에 볼록 껍질을 만드는 알고리즘이다. [2]에서 설명된 알고리즘의 총체적인 내용과 이해를 통하여 GPS를 통하여 저장한 현 위치 값과, 사용자가 저장한 좌표 값들을 비교하여 볼록껍질을 제작한 후, 현 위치가 볼록껍질의 대상군에 들어가 있는지, 없는지를 비교하여 현 위치가 지정 구역 내부에 있는지 외부에 있는지 판단한다.

Cart-Rider의 외부 유출을 막기 위한 기능이므로, 유출된 실외에서 작동하기 위하여 GPS를 사용했다. 저장된 경계 값과 GPS를 통해 수신 받은 좌표를 배열에 저장, Convex Hull Algorithm을 이용하여 껍질 대상이 될 수 있는 점 후보군을 추출하여 현 위치 좌표 값과 비교, 현 위치가 지정구역 내부인지 외부인지 판별한다. 지정구역 외부로 판단할 시, 경고음과 경고 다이얼로그를 반복하여 호출한다. 지정 구역 내로 판단 시, 경고음과 경고 다이얼로그는 더 이상 호출되지 않는다.

사용 전, 사용자가 입력 및 관리할 데이터를 위한 class를 별도로 제작한 후, DB(Database)를 cursor

에 담아 저장한다. SQLite를 이용하여 데이터 관리 및 버전관리를 내부적으로 시행했다.

2) 네비게이션 기능

실내 네비게이션 기능은 객체 인식 모델 중 Object detection과 안드로이드 앱 개발도구인 Android Studio와 게임 라이브러리인 libGDX를 활용하여 구현했다.

사용자가 네비게이션 시작 메뉴에서 목적지를 정한 이후 네비게이션 실행버튼을 누르면 추정된 현위치가 자동적으로 찍히고 선택된 목적지까지의 경로를 제시해준다. A* 알고리즘을 활용하여 현재 사용자의 위치값과 목적지를 입력하면 앱 내부에서 Priority Queue로 노드 간의 cost를 비교함으로써 최단경로를 사용자에게 제공한다. 사용자의 현 위치 추정은 카메라와 object detection을 통하여 진행이 된다.

기존 [3]에서 얘기하고 있는 ssd모델을 발전시킨 [4]Mobilenet SSD 모델이 실시간성에도 높은 fps(frame per second)를 보이는 객체 인식 모델이었기에 이에 대한 논문을 살펴본 후 이 작품에 최적이라 판단하여 이를 사용했다. 또한, 딥러닝 모델을 었을 Jetson Board위에서도 가볍게 모델이 실행될 수 있도록 최적화 라이브러리인 TensorRT를 사용하여 초당 약 50fps정도의 빠른 detection을 구현할 수 있었다.

3) Jetson 과 앱 간의 통신

TCP/IP 소켓통신을 통하여 통신을 진행한다. 소켓과 입력 받은 주소정보를 사용하여 flag값에 따른 통신 및 socket 상태를 제어할 수 있도록 한다.

앱과 jetson 모두 소켓을 열어주어 통신을 진행한다. 소켓이 close되어 통신이 끊기는 경우를 방지하기 위해 timer를 열어주어 지속적으로 소켓을 생성할 수 있도록 했다.

인식된 객체의 Class id값을 넘겨주면 앱에서 받아 좌표를 찍을 수 있도록 한다. 미리 앱 내부에 각 Class id값 마다 좌표 값을 전부 임의로 생성한 매트릭스에 따라 초기화를 시켜주어 객체 인식이 되었을 시에 좌표 값들이 잘 찍힐 수 있도록 한다.

3. 시스템 구현 효과



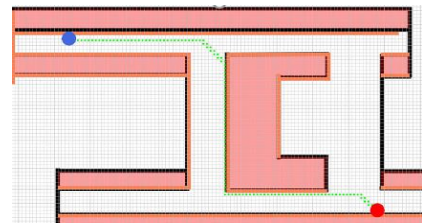
(그림 6) 로봇 장착 전 후 힘 측정 그래프

카트에 신는 Load를 45, 70kg으로 두고 실험을 진행했다. (그림 6)에서 알 수 있듯이, 로봇을 장착하지 않았을 경우에는 카트를 미는데 필요한 힘이 최대 61.7, 70.3N이 필요한 것으로 확인됐다. 반면 로봇을 장착한 후에는 평균 10N의 힘으로 무거운 카트를 밀 수 있었다. 즉 설계한 어드미턴스 controller를 통한 힘 보조 기능이 잘 작동하는 것을 알 수 있었다. 특히 cornering 시에 힘 보조 기능이 우수한 것을 확인할 수 있었다.



(그림 7) 객체 인식 과정 및 정확도

(그림 7)에서 볼 수 있듯이 객체들이 카메라와 가까이 위치할 때 최대 81.8%의 인식률을 보여준다. 정확도가 다소 떨어지지만, 50fps라는 빠른 처리속도를 보여준다.



(그림 8) 네비게이션 실행결과

주황색으로 장애물의 테두리를 표시했다. 파란색 점이 시작점이고 빨간색 점이 도착점이다. 두 점에 대한 path planning을 진행하였을 때, 초록색 점으로 경로를 보여줄 수 있도록했다. Object detection으로 현 위치를 파악한 후, 목적지를 선택함으로써 (그림 8)와 같이 초록색으로 최단 경로를 나타내는 것을 확인할 수 있다.

구역 지정서비스나 평면 선형 기하학에 대한 객체 모델을 제공하는 오픈 소스인 JTS 라이브러리를 이용하는 대신 Convex Hull Algorithm을 사용함으로써 앱의 퍼포먼스를 올렸다. GPS값을 읽음과 동시에 DB와 비

교함으로써 (그림 9)과 같은 3초이내에 응답하는 실시간성 확인이 가능하다.

```
hAcc=5 et=+10h22m22s138ms
hAcc=5 et=+10h22m25s137ms
hAcc=5 et=+10h22m27s149ms
hAcc=5 et=+10h22m30s138ms
hAcc=5 et=+10h22m32s139ms
```

(그림 9) 경계값 비교 시간 log

GPS값을 읽음과 동시에 비교함으로써 실시간 경계 값 확인. 이를 통하여 최초 정량적 목표인 경계를 넘어 간지 3000ms 내에 알람 호출을 확인할 수 있다. (그림 10)은 앱을 통해 알람 호출 화면이다.



(그림 10) 앱의 경고 다이얼 로그

```
D/MainActivity: onLocationResult: Location[fused 37.620252,127.054737
D/MainActivity: onLocationResult: Location[fused 37.620252,127.054737
```

(그림 11) GPS 값 확인 log

(그림 11)은 목표영역과 실제영역은 GPS의 소수점 5째자리까지의 분해능을 구현한 모습이다.

4. 결론

Cart-Rider를 통하여 노약자들의 힘을 보조해줌으로써 적재량이 많은 카트도 방향전환이나 손쉽게 카트를 끌 수 있도록 도와줄 수 있으며, 기존에 지정된 좌표들을 통하여 카트의 도난 현상을 방지할 수 있으며 마지막으로 사용자의 편의를 위하여 사용자의 현 위치로부터 목적지까지의 최단거리를 띄워주는 직관적인 앱으로 현 대형마트의 모든 소비자들이 조금 더 효율적인 쇼핑을 할 수 있도록 돕는다. 이 Cart-Rider를 통하여 한국의 615개의 대형마트들에 최대한 적용할 수 있도록 가장 많이 쓰이는 카트 크기를 기준으로 모듈형 로봇을 구현했다.

대형마트에 초점을 두고 로봇을 개발하였지만, 대형마트뿐 아니라 물류업계에서도 사용이 가능하다. 물류업계에서는 팔레티너와 같은 기구를 사용하여 대부

분의 물품들을 운반할 수 있다. 이런 경우 역시 동일한 메커니즘을 사용하기 때문에 Cart-Rider가 적용될 수 있는 분야이다. Cart-Rider를 위와 같은 제품에 부착함으로써 물건 운반에 필요한 노동을 도와줄 수 있으며, 물류업계마다의 스타일에 맞추어 기능의 추가 또는 제거를 통해 기능들을 새로이 적용할 수 있다. 더 나아가 호텔, 병원과 같은 실질적인 노동이 필요하면서 동시에 안전을 중요시하는 모든 분야에도 Cart-Rider의 적용 범주가 될 수 있다.

5. 참고논문

- [1] 부하 무게에 따른 힘 증강 로봇의 제어 안정성 확보를 위한 가변 어드미턴스 제어 기법 - 김효민, 박현석, 양우성.
- [2] Convex Hulls of Finite Sets of Points in Two and Three Dimensions, Commun. ACM, vol. 20, no. 2, pp. 87-93, 1977. - Franco P. Preparata, S.J. Hong.
- [3] SSD: Single Shot MultiBox Detector - Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg
- [4] Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems - Yu-Chen Chiu; Chi-Yi Tsai; Mind-Da Ruan; Guan-Yu Shen; Tsu-Tian Lee

본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다.