

시각 장애인을 위한 보행 보조 기기 Black shield

최동영, 방민호, 지도교수 박용완 교수님

영남대학교 기계IT대학 정보통신공학과

(우) 38541 경북 경산시 대학로 280

21720972@ynu.ac.kr ktminho@naver.com, ywpark@ynu.ac.kr

요 약

시각이 불편한 장애인의 실내에서의 보행의 안내는 단순히 점자 블록으로 가르쳐주는 것이 전부이다. 현재 시각 장애인의 보행 보조 방법으로는 흰 지팡이, 스마트 슈즈 등이 보편적인데, 해당 제품들은 주변의 위험을 확인하는 기능만 한다. 이와 다르게 실내에서 태그를 인식해 방향에 맞게 길 안내를 하여 실내에서의 안내를 도와줄 수 있다. 실내의 경우는 자신의 위치 정보를 받아오기 힘들다. 따라서 경로를 설정하고 방향을 바꾸는 지점마다 음성 안내로 따라가게 하여 실내에서의 안내가 가능하다. 게다가 'Black shield'는 장애물 회피 및 경로 안내까지 가능하게 하는데 단순히 출발지와 목적지를 한정 짓지 않고, 목적에 따라서 출발지/목적지를 변경하여 실내에서의 자유로운 이동이 가능하다는 점을 가장 큰 이점이라고 할 수 있다.

1. 서 론

인간의 감각 중 가장 큰 부분을 차지하는 시각이 불편한 시각 장애인들을 위하여 일상생활을 위한 필수적인 인간의 보행에 도움을 주고자 프로젝트를 기획하였다.

시각이 제한된 보행은 장애물의 거리에 따른 위치와 정확히 가고 있는지에 대한 의문 때문에 답답하고 두려움을 느낄 수 있으며, 따라서 크고 작은 사고 발생 가능성이 매우 높아서 해당 문제점을 인지하고 회피할 필요성이 매우 크다.

위와 같이 시각 장애인들을 위한 문제를 해결하기 위해 보행자 지팡이로 잘 알려진 흰 지팡이, GPS를 활용한 스마트 슈즈 등이 있다. 하지만, 이러한 제품들은 길 안내가 아닌 장애물 인식이 주 용도이고, 건물 내부에서는 GPS가 작동하지 않는다는 문제점이 있다는 점에서 해당 주제를 선정하였다.

장애물 감지 기능을 포함하고, 길 안내 기능도 추가된 프로젝트를 기획하게 되었다. 게다가, GPS 위치기반 시스템이 아니라, 태그(일종의 QR코드와 같은 패턴 코드) 기술을 사용하여 사용자의 위치와 출발·목적지를 설정할 수 있어, 실내에서 위치 인식 없이 사용이 가능한 기술로 고안하였다.

위 기술은 임베디드와 신호처리 기술을 주요 기술로 설정하였고, 아두이노 보드에 초음파 센서, 부저, 음성 모듈, 진동장치를 연결하여 임베디드 기술을 수반하고, 태그 인식을 통해 출발지와 목적지를 x, y 좌표로 배열을 설정하여 Dijkstra 알고리즘을 통해 최적의 경로로 이동을 할 수 있도록 하는 신호처리 기술을 수행한다.

이 논문은 본론부터의 구성 요소들에 대해서 알아보면, 2절에서는 해당 주제와 연관된 논문과의 비교/차별점에 대해서 설명하였고, 3절에서는 주제의 기능 블록도, 기능 순서도를 통해 어떻게 동작하는지, 또한 pseudo code를 통해 전체적인 그림을 보여줄 예정이다. 4절에서는 사용한 센서들을 명시, 실제 구현한 함수들의 동작 방법을 설명하고자 하였고, 5절에서는 실제 만든 후 성능 측정을 통해 정확도를 체크하였다. 6절에서는 해당 주제를 선정하고 만든 결과, 마지막으로 참고 문헌 순으로 해당 논문이 진행된다.

2. 관련 연구

2.1 시각장애인용 길 안내 서비스 시스템(장영건, 차주현, 시각장애인용 길안내 서비스 시스템에 대한 연구, Journal of Rehabilitation Welfare Engineering & Assistive Technology, 2017. vol. 11, no. 4, 315-321)

해당 논문에서는 실내가 아닌 실외지만, 해당 논문에서도 GPS의 오차가 발생할 수 있다는 점을 경고하고 있다. 실내가 아닌 실외의 공간에서도 오차가 발생할 수 있는 요소(전파가 잘 통하지 않는 공간 ex) 터널)과 같은 장소 때문에 GPS의 오차가 발생한다고 되어있다. 실내의 경우는 오히려 적은 이동 때문에 오히려 자신의 위치를 알아내기가 힘들다. 따라서 위치 감지가 아닌 방향 변화에 중점을 뒀 이를 바탕으로 길 안내를 진행하였다.

2.2 시각장애인을 위한 네비게이션 시스템 설계 및 구현(장수민, 황동교, 강수, 김은주, 박준호, 장기훈, 유재수, "시각장애인을 위한 네비게이션 시스템 설계 및 구현", 한국콘텐츠학회논문지 '12 Vol. 12 No. 1, 2011. 12. 1.)

해당 논문에서는 실외에서 시각장애인들의 활동 범위를 확대하기 위해 주변 시설물 검색, 길 안내 서비스를 하도록 하였다. 음성을 통해 안내해준다는 것은 일치하지만, 결국에는 본인의 위치를 API를 통해 받아와 실외에서 본인 근처에 있는 장소를 추천하는 것으로 위치기반 서비스가 기반이 된다. 위의 시각장애인용 길 안내 서비스 시스템과 마찬가지로 실외에서의 안내를 중점으로 하였다. 따라서 실내에서 점자 블록과 함께 태그를 인식하는 것을 통하여 안내를 시작하여서 목적지까지 음성안내를 하는 'Black

shield'가 실내에서는 더욱 효과적일 것이다.

3. 제안된 기술/기능의 제목

3.1 제안된 기능 구조

본 논문에서 사용되는 주요 기술은 센서에서의 2 가지 다른 센서를 통해 3단계의 다른 출력을 나타내는 임베디드 분야와 비전 센서를 통해 필요한 사물을 학습시키고 인식시켜 동작했을 때 오류 없이 인식하는 신호처리 2가지 트랙을 만족하는 것을 목표로 하였다.

[그림 1]을 보면, 우선 사용되는 중심이 되는 아두이노를 통해 각 센서들이 연결되고 아두이노에서 센서로부터 받아들이어서 해당 정보를 처리하고 각 센서들을 통해 원하는 신호로 출력하게 된다.

전반적인 동작을 살펴보면, [그림 2]처럼 Black shield에서는 메인 프로그램에서 우선적으로 2차원의 좌표값들을 설정하여 건물의 특정 층 도면을 설정하는데 이때, 해당되는 좌표값들은 특정 층 도면에서 방향이 꺾일 수 있는 장소들을 말한다. 해당 좌표값들과 매칭이 되기 위해서 태그라는 것을 사용한다.

태그는 비전 센서를 통해 인식하는데, QR코드와 비슷한 형태의 그림으로, 이를 기존에 비전 센서를 통해 인식시켜 해당 그림들이 인식될 경우 아두이노 메인 시스템에서 이를 받아들이고 이제 다른 출력을 나타낼 수 있도록 설정하는 역할을 한다.

해당 인식시켜 둔 태그들과 좌표 값들을 매칭시켜 메인 Spot과 서브 Spot을 설정하는데, 메인 스팟은 목적지로 할 수 있는 공간(예를 들면 화장실, 강의실, 엘리베이터 등)으로 설정하고, 서브 Spot에서는 방향 전환을 안내해야 하는 장소들로 설정을 한다.

좌표값과 태그들의 매칭 값이 끝난다면, 사용자가 출발지와 목적지를 설정할 수 있는데, 이때 아두이노 내에서 출발지, 목적지가 설정된다면 Dijkstra 알고리즘을 통해 최적의 경로를 설정한다. 이때, 경로에 해당되는 각 서브 스팟에서는 현재 위치에 따른 방향 전환을 각 다르게 설정하여 현재 있는 상황에 맞추어 정확하게 안내를 하게 된다.

길 안내를 하는 도중 경고, 지시 등을 나타내는 센서들이 존재하는데, 센서들은 진동장치, 음향 장치, 초음파 센서, 비전 센서를 사용한다.

먼저 초음파 센서의 경우 측정되는 거리를 아두이노로 전송하고, 아두이노에서는 인간의 걸음에 따라 장애물이 가까워질 때마다 서로 다른 주파수/세기로 부저/진동 장치로 경고를 한다.

비전 센서의 경우에는 앞에서 말했던 것처럼 걸음 도중 학습된 태그가 인식된다면 해당 정보를 아두이노에 전송하여 이를 바탕으로 방향 전환을 음성으로 출력하여 사람이 더욱 받아들이기 쉽게 해준다.

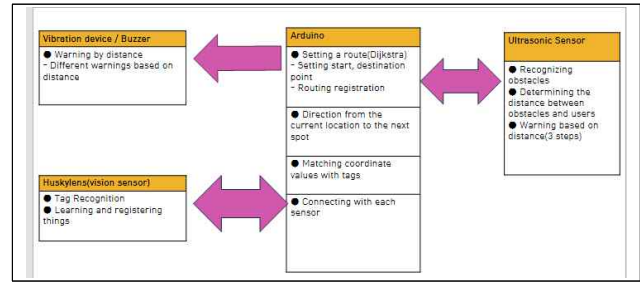


그림 1. 기능 블록도 (Functional Block Diagram)

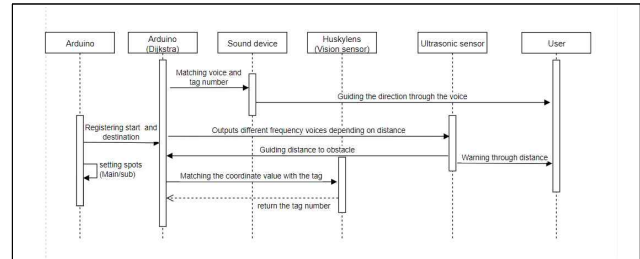


그림 2. 기능 순서도 (Sequence Diagram)

3.2 제안된 기능 구조의 알고리즘 (pseudo code)

아두이노 프로그램으로 정하고 난 후 작성한 수도 코드이다. 해당 수도 코드를 작성하기 전에는 실제 환경과 가장 비슷하게 하기 위하여 영남대학교 IT관 3층을 실거리를 측정하여 최대한 안내에 문제점이 없도록 하였으며, 그 중 학생들이 가장 많이 다니는 공간들을 메인 Spot(화장실, 강의실, 엘리베이터 등)으로 설정하였고, 방향을 바꿔야 하는 지점 혹은 방향의 안내가 필요한 여러 갈래 길로 갈수 있는 지점을 서브 Spot으로 설정하였다. 이때 2차원 행렬 (x,y)에 대해서는 반드시 인접한 다른 좌표 값이 x나 y의 값 둘중 하나는 일치하여 일직선에 있도록 설정하였다.

setup에서는 main이 실행되는 부분으로, 실제 Dijkstra 알고리즘이 해당 위치에서 실행된다. 또한 사용되는 핀번호들의 설정 및 변수 선언을 통해 아두이노에 연결된 센서 및 장치를 명시하였다.

loop 함수에서는 반복하며 장애물을 인식하고 계속해서 비전 센서에 인식되는 것을 대기하여야 하기 때문에 해당 내용들은 loop 함수에 들어가 있다. 장애물 인식의 경우 초음파 센서로부터 측정된 값에 따라 3단계로 나누어서 경고를 한다. 경고는 사람의 보폭에 따른 거리에 장애물이 존재한 경우 서로 다른 주파수를 통해서 3단계에 따른 부저와 진동 장치로 경고를 하도록 설정하였다. 그 3단계 외의 경우에는 부저와 진동장치를 꺼놓는 방법으로 해당 센서, 장치들이 동작을 하지 않도록 설정하였다.

Dijkstra 알고리즘은 메인 Spot 중에서 출발지와 목적지를 설정, 그 이후 알고리즘이 동작하면서 가중치 행렬(거리 행렬)에 따라서 테이블에 출발지와 목적지 사이에 거쳐야 하는 Spot을 설정하고, 해당 테이블에서 정보를 불러온다. 따라서 해당 테이블에

다른 방향 이동을 지시하며, 밑의 direction 함수와 함께 사용되면서 테이블에 등록되어 있는 데로 음성을 통해 안내하며, 최종적으로 도착지에 도착해서는 음성으로 종료를 알리게 된다.

마지막으로 음성을 통해 방향을 지시하는 경우 사람이 어디에 서 있느냐에 따라 왼쪽 오른쪽 구분이 정확하게 달라야 하기 때문에 이때 3가지 좌표(현재 위치, 목적지, 서브 Spot)를 상황에 따라 맞는 방향에 따라 계속해서 직진, 좌회전, 우회전을 알려주도록 설정하였다.

```

Arduino_Main

M1~M6 : Main spot, S1~S8 : Sub spot
int weight[VERTICES][VERTICES] //matrix of distances between spots
int x[14][2] //Coordinate matrix based on real distance

uint 8_t road1~17 // Voice for arrival and directional instructions

setting pin buzzer, trigPin, echoPin, sww1, sw2, vib //pin_num
setting point(start, end, temp) // Using dijkstra

void setup()
{
    Dijkstra(start, end)
}

void loop()
{
    Calculate distance by using Ultrasonic sensor
    Three-step output representations that vary with distance
    {
        if (one-step) // a person within a step
            Sound and vibration warning

        else if(two-step) // a person within a two or three step
            Sound and vibration warning

        else // a person within over three step
            No warning
    }
}
Dijkstra function
{
    path[][] = inf
    for (i = 0; i < n; i++)
        Initialization

    Comparison of lengths between adjacent spots
    (= Store self-renewed paths in the final distance array)

    distance[w] = distance[u] + weight[u][w];
}

direction function
{
    Values based on orientation based on location
}

```

그림 4. 제안하는 알고리즘의 유사코드 (pseudo code)

4. 기능 구현

4.1 하드웨어 및 소프트웨어 플랫폼

표 1. 플랫폼 정보

플랫폼	주요 내용	작성 요령 및 주의 사항
하드웨어 플랫폼	Arduino Mega 2560 R3	용량이 큰 음성처리 데이터 값과 Dijkstra 알고리즘을 수행하기 위해서 사용되는 재귀함수는 상당한 동적메모리를 사용하므로, 일반 Arduino 보드로는 메모리를 충족하지 못하기 때문에, Arduino Mega 버전을 사용함.
	Arduino UltraSensor (HC-SR04)	초음파 센서의 유효 인식 거리가 3m 이내이기 때문에, 설정 각도에 맞게 정확한 부착이 필요함.
	Arduino Speaker (ZAS-YMJ-00094)	호환 보드별 pin 번호가 다르므로, 보드 설명에 맞는 pin에 연결 필요함.
	Arduino Vibrator module (ZAS-YMJ-00031)	
	Vision Sensor (Gravity HUSKEYlens Pro)	태그를 학습 후, 각 태그에 Spot (좌표) 정보를 입력하여 사용함. *좌표는 실거리를 측정하여 축약시킨 좌표.
소프트웨어 플랫폼, 펌웨어	Arduino IDE	C/C++ 언어를 기반으로 사용되지만, 추가적인 Arduino 전용 함수와 명령어도 사용해야 함.
	processing IDE	측정한 실거리를 바탕으로 processing 동작을 통해 가상맵을 구현하여 태그 위치를 지정함.

4.2 Spot 설정

영남대학교 IT관 3층을 기준으로 건물 내부의 출발지와 도착지로 설정할 수 있는 공간(화장실, 강의실, 엘리베이터, 계단 등)을 메인 Spot으로 설정하고, 방향을 바꾸거나 계속 직진해야 하는 통로 및 코너를 서브 Spot으로 설정하여 건물 내의 실거리를 바탕으로 좌표를 설정하였다.

Spot 위치를 processing 프로그램을 사용하여 가상맵 형태로 간단하게 도식화 작업을 하였다. 아래의 그림에서 보이는 것처럼 메인 spot은 빨간 점으로, 서브 Spot은 초록 점으로 표시를 하여 구분 지었다. Spot 설정 후 Processing을 통해 좌표값을 넣은 대로

아두이노 프로그램에서 해당 좌표값들을 가진 배열을 생성하였고, 해당 배열에서는 Dijkstra를 통해 최종적으로 출발지와 목적지를 벽과 같은 장애물들을 피하면서 길 안내를 하도록 하는 작업을 진행하였다.

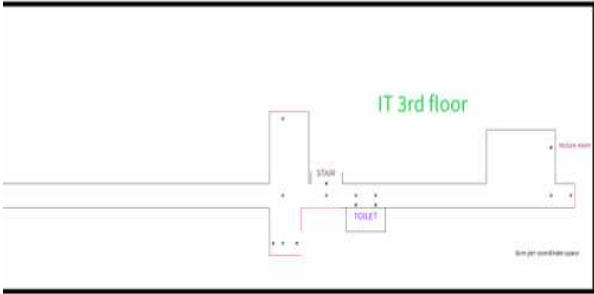


그림 . processing을 통한 좌표 도식화 작업

4.2 최적의 경로 설정(Spot Path Finding Algorithm - Dijkstra)

Dijkstra 알고리즘은 위의 도식화 작업이 끝난 뒤 지정된 Spot의 좌표 행렬과 해당 Spot 사이의 거리를 나타내는 행렬을 받아온다. 먼저 각 Spot 사이의 거리 행렬을 무한대로 초기화 시키고 출발지와 목적지를 설정한다.

출발지와 목적지를 정해지면 초기화된 거리 행렬을 바꾸면서 행렬을 갱신하는데 출발지부터 시작하여 방문하지 않은 여러 Spot 중에서 가장 짧은 거리를 가진 Spot에 연결하여 경로 행렬을 갱신한다. 그 후 이어진 Spot에서 다음 방문하지 않은 행렬들을 검색하며 해당 과정을 마지막 목적지에 도착할 때까지 반복해 해당 출발지와 목적지까지의 최단 거리를 갱신하고 해당 갱신된 것에 따라 최단 거리에 맞게 방향을 음성을 통해 알려주며 정확하게 안내하도록 설정한다.

5. 성능 측정 및 분석

5.1 임베디드 트랙, 핵심 모듈 성능 측정

단계별, 사용자와 물체 사이의 기준 거리 설정(거리:d(m))

트랙	핵심 모듈	설계 목표	현재 성능 및 달성도
임베디드	초음파센서	보행을 방해하는 장애물을 관측 시, 센싱을 통해 사용자와 해당 장애물 간의 거리를 전송한다.(2초내 전송)	센서를 통해 실 거리를 측정했을 때 Serial Monitor를 통해 거리 값이 올바르게 전송됨을 확인 (100%)
	부저 및 진동장치	초음파 센서를 통한 거리 정보를 받아 장애물과의 거리가 2m 이하가 되면, 세가지 기준(보폭 수)에 따라 경고 음성 및 진동을 출력하여 위험을 경고한다.(단계별 출력)	사람의 보폭을 고려하여 1단계 2m, 2단계 1.3m, 3단계 0.7m로 3단계 위험신호를 알림(100%)

3단계(0m<d<0.7m), 2단계(0.7m<d<=1.3m), 3단계(1.3m<d<2m)

모듈	초음파센서 (Serial Monitor)	음성 모듈	진동장치
실제거리			
0	0	작동X	작동X
0.69(m)	69(cm)	3단계 음성	3단계 진동
0.71(m)	71(cm)	2단계 음성	2단계 진동
1.29(m)	129(cm)	2단계 음성	2단계 진동
1.31(m)	131(cm)	1단계 음성	1단계 진동
1.99(m)	199(cm)	1단계 음성	1단계 진동
2.01(m)	201(cm)	작동X	작동X

그림 6. 성능 측정 결과 표

초음파가 측정된 거리를 실거리로 변환하는 식을 통해 변환하여(trigpin의 동작을 low, high로 측정한 시간이 왕복한 시간이기 때문에 반을 나눈다. 그 이후 소리의 속도 0.034cm/us를 곱하여 실제 우리가 사용하는 거리 cm, m를 측정할 수 있게 하여 해당 거리를 실제거리와 초음파 센서간의 거리로 비교하여본 결과 실제 거리와 일치하는 것을 확인하였다. 또한 실거리를 바탕으로 나누는 구간 값에 따라서 음성 모듈(부저)와 진동장치의 세기가 달라지는 것을 확인할 수 있었다.

5.2 신호처리 트랙, 핵심 모듈 성능 측정

트랙	핵심 모듈	설계 목표	현재 성능 및 달성도
신호처리	비전센서	- 2초 내에 태그를 인식하여 Pass or Fail을 판별하고 올바른 경로의 태그를 인식 시 다음 방향을 제시한다. - 실 측정거리를 바탕으로 좌표값을 설정하여 길찾기 알고리즘을 통해 실시간 길안내 음성을 지원한다.	태그를 학습시키고 좌표를 매칭시켜 해당 태그를 인식 시에 태그의 정보에 따라 방향안내 및 실시간 음성신호를 출력함을 확인 (100%)

비전센서 태그인식 정확도(%)

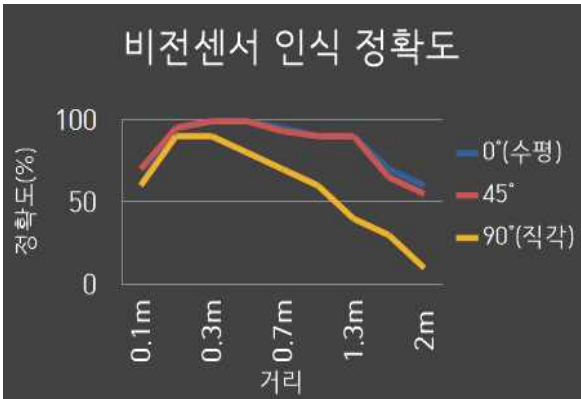


그림 7. 성능 측정 결과 분석 (꺾은선 그래프)

비전 센서의 경우 태그라는 QR코드와 비슷한 형태의 그림을 미리 학습시켜 놓았는데, 해당 그림들을 메인 Spot, 서브 Spot으로 구분 지어서 메인 스팟에서의 도착의 경우 해당 목적지에 도착한 것을 음성으로 출력이 가능하게 하였다. 실제로 음성뿐만 아니라 아두이노 모니터를 통해 해당 태그가 인식되는 것을 해당 좌표값과 함께 출력하는 것을 통해 청

각뿐만 아니라 시각으로까지 완벽하게 동작하는 것을 확인하였고, 출발지와 목적지가 바뀔에 따라 태그는 그대로지만 음성이 계속 변화하며 정확한 길 안내를 하는 것을 확인하였다.

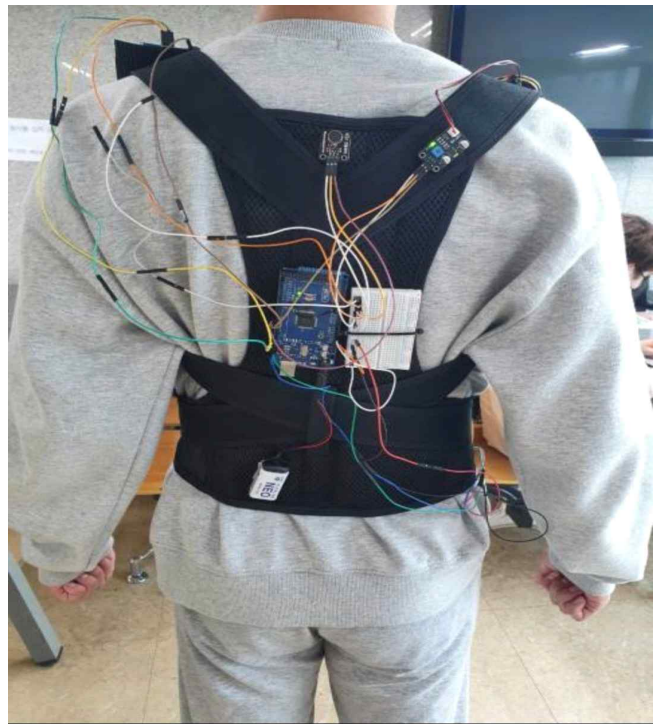
6. 결 론

결론에는 다음 항목들이 반드시 포함되도록 할 것: 시각은 인간의 오감중 80%를 차지하여, 시각 장애인들의 일상생활 중 실내에서의 보행이 많이 불편하다는 점을 고려하여 이를 극복하고자 해당 기술을 고안하게 되었다. 해당 기술은 장애물 인식뿐만 아니라 실내에서 태그라는 것을 인식하여 방향 안내가 가능하도록 하였다. 기존의 기술은 실내보다는 실외, 또한 길안내보다는 장애물 인식을 중점으로 하였다. 실내의 경우에는 위치 인식이 어렵다는 점에서 단순히 바닥에 점자 블록을 통해서 멈추는 지점을 표시해 놓았지만, 이는 해당 도착 장소가 목적지인지 판별하는 것이 불가능하다. 하지만 Black shield를 사용한다면, 목적지에 따라서 방향이 꺾이는 지점마다 태그를 인식하여 정확하게 갈 수 있다. 실제 완성 후 눈을 가리고 측정한 경우 느리게 걷긴 하지만, 음성을 통해 정확히 이동하는 것을 알 수 있었으며, 장애물 인식 역시 모니터를 통해 거리를 확인하며 걷는 결과 100%로 서로 다른 거리에 따른 다른 출력을 나타내는 것을 확인할 수 있었다. 실제 현재 다른 논문들을 비교해 본 결과 실내보다는 실외의 길 안내를 가능하게 하던가, 실내의 큰 공간에서의 위치를 알아보는 것이 대다수였다. 하지만 Black shield는 실내의 그렇게 크지 않는 공간이더라도 실거리를 측정하고 이에 대해서 각 spot에서의 좌표를 설정을 원활하게 한다면, 실내에서 정확하게 목적지로의 안내가 가능하다는 점이 가장 큰 장점이 라고 할 수 있다.

마지막으로 해당 기술은 현재 IT관 3층에 한정되어 있지만, 다른 건물이나 다른 층에서 실거리를 측정하고 해당 비율을 정확하게 설정한다면, 해당 측정이 끝난 어느 위치에서든지 이를 바탕으로 시각 장애인들에게 정확한 안내를 통해 편의성이 매우 높아질 것이다.



정면 사진



후면 사진

참고문헌

7.1 저널 및 칼럼 (Journal & Column)

- [1] 고재현, “눈과 시각”, *빛으로 보는 세상*(한국일보 공동기획), 사이언스타임즈, 2005. 9. 7
- [2] 유경민, “눈 뜬 자들의 도시에서 길 잃은 시각장애인의 하루”, *시사저널*, 1504호, 2018. 8. 10
- [3] ITN, 국제섬유신문, “시각장애인 위한 스마트슈즈 개발”, 2014. 3. 3
- [4] 양대규, “자율주행차의 눈이 된 비전 AI”, *Ai타임즈*, 스페셜리포트④, 2020. 10. 30
- [5] Industrial Communication Network, “자율주행차와 스마트카를 위한 라이더(LiDAR) 기술”, *ICN*, 2017. 4. 21

7.2 논문지 (Paper)

- [6] 장영건, 차주현, *시각장애인을 위한 길안내 서비스 시스템에 대한 연구*, *Journal of Rehabilitation Welfare Engineering & Assistive Technology*, 2017. vol. 11, no. 4, 315-321
- [7] 정일홍, 김은지, *가상공간지도 기반의 3차원 가상공간 저작도구의 개발*, *한국컴퓨터정보학회논문지*, 2006, vol.11, no.2, pp. 177-186
- [8] 장수민, 황동교, 강수, 김은주, 박준호, 장기훈, 유재수, “시각장애인을 위한 네비게이션 시스템 설계 및 구현”, *한국콘텐츠학회논문지* ‘12 Vol. 12 No. 1, 2011. 12. 1.

7.3 국제 표준 및 상용 제품의 매뉴얼 자료

- [9] DFROBOT, “SEN0305 – Gravity: HUSKYLENS – An Easy-to-use AI Machine Vision Sensor”
- [10] GitHub, “Talkie” (<https://github.com/going-digital/Talkie>)
- “Arduino Mega 2560(R3)”