

# 시각장애인 촉각기반네비게이션에 관한 연구

강민석\*, 박상은\*\*, 차건환\*\*\*, 최승현\*\*\*\*, 한동윤\*\*\*\*\*,  
 김수동\*\*\*\*\*, 박진호\*\*\*\*\*  
 \*~\*\*\*\*\* 숭실대학교 소프트웨어학부  
 e-mail:qkrguswndk@naver.com

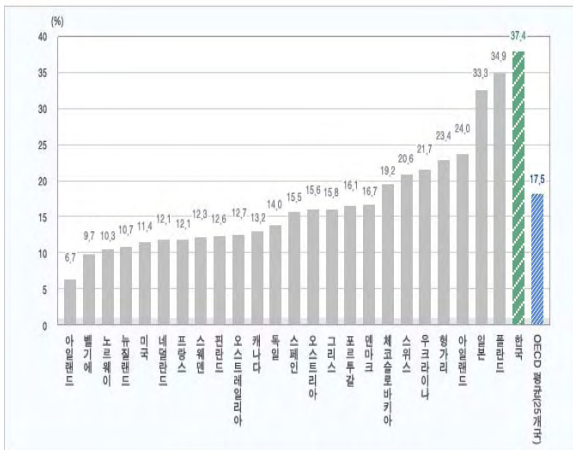
## An Empirical Study on Tactile-Based Navigation for Securing Walking Safety of the Visually Impaired

Min-Suk Kang\*, Sang-Eun Park\*\*, Kun-Hwan Cha\*\*\*, Seong-Hyeon Choi\*\*\*\*, Dong-Yun Han\*\*\*\*\*, Soo-Dong Kim\*\*\*\*\*, Jin-Ho Park\*\*\*\*\*  
 \*~\*\*\*\*\* Dept of Software, Soong-Sil University

### 요 약

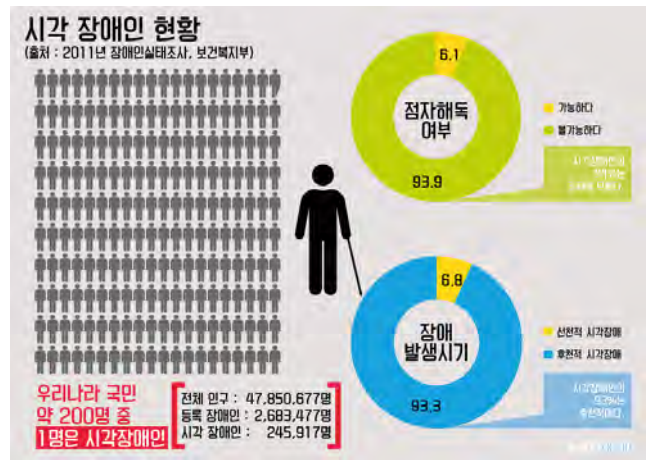
현재, 시각장애인들이 보행 시 경로 안내를 제공받는 방법은 청각을 사용하는 것이다. 이로 인해 시각장애인의 보행시 안전이 매우 위험한 상황이다. 비장애인은 시각과 청각을 모두 활용해 길 찾기가 수월하지만, 현 시점에서 시각장애인은 오직 청각에만 의존해야 한다. 보행 시 교통사고 사망률이 OECD 평균의 두 배를 웃도는 대한민국에서 시각장애인들이 거리를 도보하기에는 더더욱 위험한 현실이다. 이에 대한 대책으로 본 논문에서는 아두이노를 사용해 촉각기반 네비게이션을 만들어 시각장애인의 교통사고 발생율을 낮추려고 한다.

### 1. 서론



(그림 1) OECD 국가의 보행 중 교통사고 사망률

OECD 국가의 보행 중 교통사고 사망률 평균은 17.5%이다. 그 중 대한민국은 37.4%이며, OECD 평균을 훨씬 웃도는 수준이다. 다시 말해서, 대한민국의 보행자의 안전은 OECD 다른 국가의 수준에 미치지 못한다는 것이다. 이런 상황 속에서 시각장애인은 비장애인들보다 도로에서 더 많은 위험을 겪고 있을 것이다.



(그림 2) 시각 장애인 현황

현재 대한민국 시각 장애인 수는 거의 245,917 명이다. 즉 대한민국 국민 200명 중 1명은 시각장애인이라는 의미이다. 이 사람들은 모르는 길을 도보로 찾아갈 때 청각을 활용하여 안내를 받아야 한다. 하지만 이 방법은 사용자의 청각을 차단 할 수 있다는 치명적인 문제점이 있다.

\*\*\*\*\* 교신저자,

※ 이 논문은 서울어코드 활성화 사업에서 지원되었음.



(그림 3) 이어폰 사용 시 경적소리 인지 거리 감소

스마트폰과 이어폰을 사용한다면, 경적소리 인지 거리가 40~75%가량 감소한다. 시각장애인들이 경로 안내를 받을 때 청각을 사용한다면, 외부 소리 인지 능력이 감소하고 이는 교통사고로 이어질 수 있다.

본 논문에서는 이를 위한 대책으로 촉각을 사용하는 네비게이션을 제시한다. 길을 찾을 때 자주 사용되지 않는 촉각이라는 감각을 사용함으로써 남겨진 시각을 자유롭게 사용할 수 있는 여지를 남겨 두는 것이다.

## 2. 본론



(그림 4) 에어나비 형태

에어나비는 그림4와 같은 형태로 손 안에 들어오는 형태로 만들어졌다. 방향침이 튀어나와 손가락으로 방향을 느낄 수 있도록 하는 형태이다.

사용자는 스마트폰에서 에어나비 앱을 실행하고, 목적지를 음성으로 입력한다. 그 후, 방향침을 만지며 안내 받은 진행방향으로 이동하면 된다. 만약 횡단보도 등과 마주친다면, 진동을 울려서 사용자에게 주의를 줄 수 있다.

이를 구현하기 위해서, 먼저 그래프를 만든 후 Dijkstra-Algorithm을 사용해서 지도에서 모든 경로에 대해서 최단 경로를 구하여 이 결과를 활용해 안내를 한다.



(그림 5) 숭실대 부근 지도를 추상화한 그래프

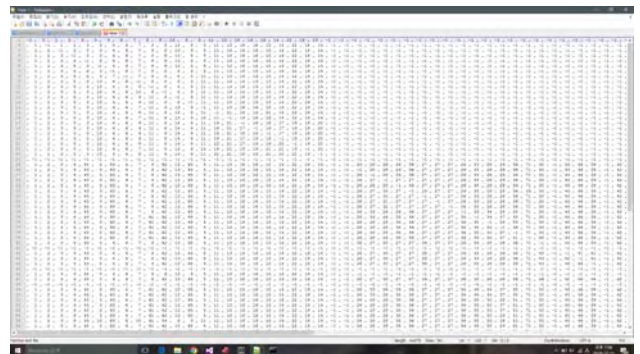
먼저, 그래프를 만들기 위해 먼저 건물과 도로에 정점을 만들고, 서로 인접한 것들을 간선으로 연결한다. 모든 정점들은 자신의 GPS 좌표값을 가지고 있다.

```
priority_queue<node, vector<node>, greater<node>> heap;
heap.push({ 0, k });

while (!heap.empty()) {
    int idx = heap.top().second;
    heap.pop();
    if (check[idx])
        continue;
    check[idx] = true;
    for (auto i : gnext[idx]) {
        if (dist[i.first] > dist[idx] + i.second)
        {
            dist[i.first] = dist[idx] + i.second;
            heap.push({ dist[i.first], i.first });
        }
    }
}
```

Dijkstra Algorithm 구현 (C++11사용)

만들어진 그래프를 Dijkstra Algorithm을 사용하여 최단경로를 구했다. priority\_queue를 사용하였기에 최단경로를 구하는 시간복잡도는  $O(n \log n)$ 이다.



(그림 6) 최단경로를 저장한 이차원 배열

이렇게 구한 최단경로를 위와 같이 이차원 배열의 형태로 저장을 하여 아두이노에 데이터를 저장하였다. 배열의 size가 96 X 96 이어서 아두이노 램에 이 모든 정보를 저장할 수 없었다. 그래서 아두이노 플래쉬 메모리에 값을 저장할 수 있게해주는 PROGMEM 키워드를 사용하였다. 배열에 데이터를 path[now][destination] = next\_vertex\_idx 로 저장을 해놓았기 때문에, 현재 정점의 index와 목적지로 지정한 정점의 index를 안다면 경로를 찾아갈 수 있다. 현재 정점의 index와 목적지의 index가 같다면 목적지의 도착한 것으로 판단한다. 아두이노에서 작동하는 pseudo code는 다음과 같다

```
while(now != destination) {
    int next_idx = path[now][destination]
    NavigatePath(now, next_idx)
    now = next_idx
}
```

에어나비는 시각장애인들이 목적지까지 도달하기 위해 어떤 방향으로 움직여야 할지 안내해 주는 기기이므로, 스텝모터를 얼마나 회전시킬지 그 각도를 정하는 알고리즘이 중요하다. 현재 정점과 다음 정점의 GPS 좌표를 이용하여 두 정점간의 각도를 구할 수 있지만, 에어나비 사용자 신체의 방향이 경로를 이동하면서 변하는 경우가 있다. 이에 대한 솔루션으로, 회전변환과 평행이동을 사용하였다. 처음에 두 정점 사이의 각도만큼 회전 변환을 취해준 다음, 그 이전 각도만큼 다시 평행이동을 해준다면, 경로를 따라 이동하는 사용자에게 적합한 방향으로 안내를 해 줄 수 있다. 다음은 이 솔루션에 대한 구현이다.

```
float GetDegree() {
    float now_x = float(vertex_x[now]) / 100000; // 현재위치 받아와야함
    float now_y = float(vertex_y[now]) / 100000; // 현재위치 받아와야함
    float next_x = float(vertex_x[next]) / 100000;
    float next_y = float(vertex_y[next]) / 100000;
    //회전이동
    float tmp_now_x = cos(theta) * now_x + sin(theta) * now_y;
    float tmp_now_y = -1 * sin(theta) * now_x + cos(theta) * now_y;
    float tmp_next_x = cos(theta) * next_x + sin(theta) * next_y;
    float tmp_next_y = -1 * sin(theta) * next_x + cos(theta) * next_y;

    //평행이동
    tmp_next_x -= tmp_now_x;
    tmp_next_y -= tmp_now_y;
    tmp_now_x = 0, tmp_now_y = 0;

    //거리계산
    float distance = sqrt(pow(tmp_next_x, 2) + pow(tmp_next_y, 2));
    //prev_degree = degree;

    //사인역함수로 각도구하기
    degree = asin(distance / tmp_next_x); // 평행이동한후에 각도 계산임
}
```

```
//사분면 계산
if (tmp_next_x > 0 && tmp_next_y < 0) // 4사분면
    degree += 270;
else if (tmp_next_x < 0 && tmp_next_y < 0) // 3사분면
    degree += 180;
else if (tmp_next_x < 0 && tmp_next_y > 0) // 2사분면
    degree += 90;

return degree;
}
```

위 함수에서 반환한 각도만큼 스텝모터를 회전하며, 현재 정점을 갱신하는 것을 반복하면 사용자에게 목적지까지 경로안내를 할 수 있다.

**3. 결론**

모르는 길을 찾아갈 때 네비게이션 만큼 효율적인 방법은 없을 것이다. 하지만 현재, 시각장애인들을 위한 네비게이션은 경로는 안내할 수 있지만 거리에서 그들의 안전까지 보장해 주지는 못한다.

이 문제에 대해, 에어나비는 경로 안내에서 잘 사용하지 않는 촉각을 이용하여 시각장애인들이 자유롭게 청각을 사용할 수 있게 한다. 이는 시각장애인들의 안전을 보장하며 경로를 안내하는 솔루션이 될 수 있다.

**참고문헌**

- [1] 김상연, 조성만, "시각장애인을 위한 햅틱 네비게이션 시스템", 멀티미디어학회 논문지, Vol.14, issue 1, 2011.
- [2] 김창기, 서정민, "스마트 기기를 이용한 시각장애인 네비게이션 시스템", 한국콘텐츠학회논문지, Vol.15, issue 1, 2015.
- [3] 윤창호, 박재권, 강중훈, 정주은, 정민아, "시각장애인을 위한 보행보조 지팡이와 네비게이션", 한국정보과학회 학술대회, 2015.