

위치기반 서비스에서 이동 패턴 정보 추출을 위한 실험 데이터 수집

임재걸*, 이강재**, 정승환***

Experimental Data Collection for Moving Pattern Information Extraction in Location Based Service

Jaegel Yim*, Kangjai Lee**, Seunghwan Jeong ***

요약

위치기반 서비스(LBS: Location Based Service)는 사용자의 현재 위치를 파악하여, 사용자의 위치를 고려한 유용한 정보를 제공하는 서비스를 일컫는다. 위치기반 서비스에는 디렉토리 서비스, 게이트웨이 서비스, 유틸리티 서비스, 표현 서비스, 경로 서비스 등이 있는데, 이러한 서비스를 개발 하려면 필수적으로 사용자의 위치를 파악해야 한다. 본 논문에서는 옥내 무선근거리통신망 환경에서 Fingerprint 방식으로 의사결정트리를 이용한 옥내 측위 방법을 소개한다. 또한 이 측위 방법으로 실험 데이터를 수집하고, 이를 이용한 이동 패턴 정보 추출에 대하여 살펴본다. 실험을 위해 의사결정트리를 생성하는 알고리즘과 현재 위치를 판정하는 알고리즘을 소개하고, 이 알고리즘을 적용한 옥내 측위 프로그램을 이용한다.

▶ Keyword : 위치기반 서비스(Location Based Service), Fingerprint, 의사결정트리(Decision Tree)

• 제1저자 : 임재걸

* 동국대학교(경주) 컴퓨터멀티미디어학과 ** 수원과학대학 컴퓨터정보과 *** 동국대학교(경주) 전자계산학과

I. 서론

위치기반 서비스(LBS: Location Based Service)는 사용자의 현재 위치를 파악하여, 사용자의 위치를 고려한 유용한 정보를 제공하는 서비스를 일컫는다. 위치기반 서비스에는 첫째, 사용자가 지정한 혹은 사용자의 위치에서 가장 가까운 장소, 생산물, 서비스를 찾아주는 디렉터리 서비스, 둘째, 사용자의 위치를 찾아주는 게이트웨이 서비스, 셋째, 주어진 장소의 이름, 거리와 번지, 혹은 우편번호에 대한 지리적 위치를 결정하여 주는 지오코더(geocoder)와 반대로 지리적 위치를 입력받아 완벽한 주소를 찾아주는 역지오코더(reverse geocoder)를 포괄적으로 지칭하는 위치 유틸리티 서비스, 넷째, 모바일 단말기 상에 지리적인 정보를 표현하는 서비스를 일컫는 표현 서비스, 다섯째, 사용자에게 적당한 경로를 제공하는 경로 서비스 등이 있다[1].

위치기반 서비스를 개발하려면 필수적으로 사용자의 위치를 파악해야 한다. 사용자의 위치를 파악하는 측위 방법은 옥외용과 옥내용으로 구분되며, 옥외용은 GPS 방법[2]이 일반적으로 사용되고 있다. 옥내용으로 크리켓(Cricket)[3], 액티브 배지(Active Badge)[4] 등 다양한 시스템이 소개된 바 있는데, 기존의 이러한 방법들은 실내의 일정 영역에 적외선 센서를 부착하거나, 천정에 초음파와 RF 신호 발생기를 부착하는 등 특수한 환경을 꾸며주어야 작동한다.

특수 장비를 이용하는 방법은 범용성과 경제성이 미약하므로, 이미 설치된 무선근거리 통신망을 이용한 옥내측위 방법이 근래에 활발히 소개되었다. 무선근거리통신망을 이용하는 옥내측위 방법은 수신신호의 세기를 이용하는 방법[5, 6, 7, 8]과 신호의 도착 시각(TOA: Time of Arrival)을 이용하는 방법[9]이 있다.

기존에 무선근거리통신망 환경의 옥내 측위 방법들 중 가장 정확도가 높은 방법은 RADAR[10]에 소개된 Fingerprint 방식의 K-NN 방법으로 알려져 있다. Fingerprint 방식은 준비 단계와 실시간 단계로 구성된다. 준비 단계에서는 후보 지점별로 특징 데이터를 미리 측정하여 Training Data를 생성하며, 실제 실시간 단계에서는 현재 위치에서 측정된 특징 데이터를 가지고 Training Data를 참조하여 현재 위치를 추정한다. RADAR의 K-NN의 경우 특징 데이터로 UDP 패킷의 신호 세기를 사용하여, 실시간 단계에서 측정된 특징 데이터를 준비 단계 때 후보 지점별로 측정했던 Training Data들과 비교하여 가장 유사한 K 개의 후보 지점을 선택하고, 이들의 평균 좌표를 현재 위치로 판정

하게 된다.

본 논문에서는 옥내 무선근거리통신망 환경에서 Fingerprint 방식으로 의사결정트리를 이용한 옥내 측위 방법을 소개한다. 또한 이 측위 방법을 이용하여 실험 데이터를 수집하고, 수집된 실험 데이터를 통해 사용자가 이동할 때의 이동 패턴 정보 추출에 대하여 살펴본다. 실험을 통해 측위 방법의 정확도를 보이며, 실험을 위해 의사결정트리를 생성하는 알고리즘과 현재 위치를 판정하는 알고리즘을 소개하고, 이 알고리즘들이 적용된 옥내 측위 프로그램을 작성한다. 여기서 Fingerprint 방식의 특징 데이터는 무선랜 카드가 장착된 Lab-Top 컴퓨터가 주변에 설치된 AP들의 신호 세기를 측정하여 이용한다.

II. 관련 연구

본 장에서는 기존의 다양한 옥내 측위 시스템에 대하여 알아보고, 본 논문에서 보이는 의사결정트리 방법을 설명한다.

1. 기존의 연구

기존의 옥내 측위 시스템의 예로 크리켓[3] 시스템에서는 고정 지점에 부착된 비컨(beacon)들이 고주파 신호와 초음파 신호들을 동시에 송신하면 이동물체에 부착된 수신기가 이 신호들을 수신하고, 신호들의 도착 시각의 차이를 바탕으로 거리들을 구하고, 비컨들의 좌표와 거리를 바탕으로 자신의 좌표를 구한다.

액티브 배지[4] 시스템에서는 이동물체들에 적외선을 발사하는 송신기가 부착되어 있고, 방마다 여러 개의 수신기가 고정 지점에 설치되어 있다. 적외선은 벽을 통과하지 못하기 때문에 이동물체가 어느 방에 들어오면 그 방에 설치된 고정 수신기들만 그 이동물체의 적외선을 감지할 수 있다. 수신기는 중앙컴퓨터에 연결되어 있어서 수신기에 감지된 적외선을 바탕으로 이동물체가 어느 방 안에 있다는 것을 판정한다. 이 상에서 본 바와 같이, 크리켓이나 액티브 배지 시스템에서는 측위를 위한 특수 장비가 필요함을 알 수 있다. 이외에도 특수 장비를 이용하여 사용자의 옥내 위치를 파악하는 다양한 연구 결과가 발표된 바 있다.

사용자가 어디에 있는지 인터넷을 사용할 수 있도록 하기 위하여 무선근거리통신망이 거의 어느 곳에서든 제공된다. 따라서 무선근거리통신망을 이용하여 사용자의 위치를 판정할 수 있다면 옥내 측위를 위한 특수 장비가 필요 없으므로 경제적이고 구현하기에 편리하다[5, 6, 7, 8, 9, 10]. [5, 6, 7, 8]은 AP(Access Point)의 신호의 세기를 바탕으로 이동단

말기와 AP간의 거리를 구하고, 이 거리를 바탕으로 이동단말기의 위치를 구한다. [5, 6, 7]은 정확도를 제고하기 위한 방법을 소개하는데 반하여, [8]은 처리 속도를 개선하는 방법을 소개한다. AP 신호는 간섭과 반사 등의 영향을 받음으로 정확도가 비교적 낮다는 단점이 있다. 그래서 신호 세기가 아닌 신호가 전송되는데 걸리는 시간으로 거리를 구하는 방법이 [9]에 소개되어 있다.

[9]는 AP 신호가 도착하는데 걸리는 시간(TOA)을 이용하여 거리를 구하는 방법을 소개한다. 이 방법을 이용하려면 정확도가 높은 시계가 장착되어야 하고 물리계층에 가까운 계층의 통신을 해야 한다.

2. 의사결정트리 방법

아래 [표 1]과 같은 Training Data를 이산화한 Table이 있다고 가정하고 이 테이블을 Root로 하여 의사결정트리를 생성하는 과정은 다음과 같다.

다섯 개의 AP 중, Gain이 가장 큰 AP_i 를 현재 노드의 레이블로 하고, AP_i 의 도메인이 $interval_1, \dots, interval_k$ 라면, 이에 속하는 각 $interval_k$ 에 대하여 $interval_k$ 를 레이블로 하는 가지를 만들고 이 가지에 AP_i 의 값이 $interval_k$ 인 모든 행으로 구성된 테이블에서 AP_i 열을 삭제하여 작성한 테이블을 자식노드로 생성한다.

[표 1]에서 AP_i 요소에 대한 Gain은 식 1처럼 테이블의 Information(정보이득)에서 AP_i 의 Entropy를 뺀 값이다.

$$Gain(AP_i) = I(S_1, S_2, \dots, S_m) - E(AP_i) \dots\dots [식 1]$$

여기서 $I(S_1, S_2, \dots, S_m)$ 는 테이블을 분류하는데 필요한 정보의 양으로써, 1 부터 m 까지의 Class에 대하여 S_i 는 i 번째 Class에 속하는 샘플의 개수를 뜻한다. P_i 는 i 번째 Class가 나올 확률이라고 하면 Information은 [식 2]처럼 된다.

$$(P_i = \frac{S_i}{|S|})$$

$$I(S_1, S_2, \dots, S_m) = - \sum_{i=1}^m P_i \log_2(P_i) \dots\dots\dots [식 2]$$

그리고 $E(AP_i)$ 는 현재 테이블을 AP_i 를 기준으로 분류하였을 때, 분류 결과를 분류하는데 필요한 정보의 양으로 [식 3]으로 구한다. 여기서 j 는 선택된 AP_i 의 동일한 속성들을 의미한다.

$$E(AP_i) = \sum_{j=1}^v \frac{S_{1j} + S_{2j} + \dots + S_{mj}}{S} I(S_{1j}, S_{2j}, \dots, S_{mj}) \dots [식 3]$$

생성된 각각의 자식 테이블들은 다시 위에 설명한 방법과 동일하게 자식 테이블을 생성한다. 더 이상 자식 테이블을 생성할 수 없거나, 자식 테이블을 구성하는 모든 샘플이 동일한 Class에 속하면 현재 테이블은 의사결정트리의 단말노드에 해당한다. [그림 1]은 [표 1]에 대해 의사결정트리를 생성한 예이다.

표 1. Training Data를 이산화한 Table 예

AP1	AP2	AP3	AP4	AP5	Class
interval1	interval2	interval2	interval3	interval1	지점1
interval1	interval1	interval2	interval2	interval3	지점1
interval2	interval3	interval1	interval3	interval1	지점1
interval3	interval3	interval2	interval2	interval2	지점2
interval2	interval3	interval1	interval2	interval2	지점2
interval1	interval2	interval3	interval1	interval1	지점2
interval1	interval1	interval1	interval3	interval1	지점3
interval2	interval2	interval2	interval2	interval3	지점3
interval3	interval1	interval1	interval3	interval3	지점3

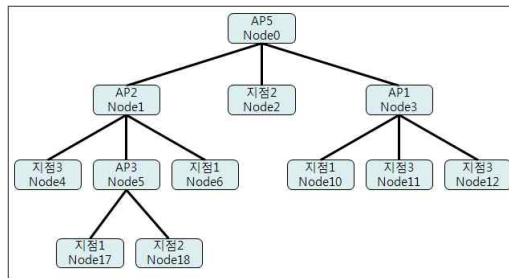


그림 1. [표 1]에 대한 의사결정트리

III. 옥내 측위 프로그램 개발

본 장에서는 의사결정트리를 이용한 옥내 측위 프로그램의 구현 환경과 프로그램의 구성에 대하여 설명한다.

1. 개발 환경

결정 트리를 이용한 측위 프로그램을 Lab-Top 컴퓨터에서 구현하였다. 프로그램 개발 도구로는 Microsoft Visual Studio 2005의 C# 언어를 이용하였다. 실험에 사용된 Lab-Top 컴퓨터에는 무선랜카드(Intel(R) PRO/Wireless 2200BG Network Connection)가 장착되어 있다. 주변에 설치된 AP들이 SSID, BSSID, RSSI, Network Type 등의 정보를 브로드캐스트하면, 이 무선랜카드를 통해 정보들을 캐치하여 실험에 필요한 특정 데이터(신호 세기)를 얻을 수 있다. [11]에는 신호 세기를 측정할 수 있는 RSSI 판독 라이브러리 함수에 대하여 자세히 설명되어 있고, 본 연구에서는 이 라이브러리 함수를 이용한다.

2. 의사결정트리를 이용한 측위 프로그램

다음 [표 2]는 준비 단계 때 작성한 Training Data를 가지고 의사결정트리를 생성하는 알고리즘이며, [표 3]은 실시간 단계에서 새로 측정된 신호 세기를 가지고 앞서 생성한 의사결정트리를 참조하여 현재 위치를 판정하는 알고리즘이다.

그리고 [그림 2]는 의사결정트리를 이용한 측위 프로그램의 이벤트 흐름도이다. 준비 단계 때 Training Data를 작성하기 위해서는 실제 후보 지점으로 이동한 후 지도정보가 출력된 Box에서 해당 위치를 클릭하면 된다. 이 경우 [그림 2]의 이벤트 흐름도에 보이는 것처럼 현재 위치에서 특정데이터를 측정하여 Table에 추가하게 된다. 후보 지점별로 측정된 특정 데이터들이 모두 Table에 적재되면 준비 단계는 완료된 것이며, 이 Table이 Training Data가 되는 것이다.

표 2 의사결정트리를 생성하는 알고리즘

```

algorithm CreatDecisionTree(int[][]
    localTable,
    MacListType localMacList)
사전조건: localTable은 ROOT 테이블, 또는 부모
테이블이다. localMacList는 localTable의 AP
필드명(MacAddress)이 저장되어 있다.
사후조건: subTable, subMacList 생성하여
재귀호출.
트리의 단말노드에 도달하기까지 노드를 구하여
반환한다.

1. 새로운 노드(node) 생성
2. if(localTable의 레코드 수 == 0)
    
```

```

        2.1 node = null
        2.2 return (node)
3. else if(localTable의 레코드 수 == 1)
3.1 node.label 에 Class 필드의 값(지점) 저장
3.2 return (node)
4. else if(localTable의 필드 수 == 1)
4.1 node.label 에 필드의 값들을 비율로 저장
4.2 return (node)
5. else if(localTable의 Class 필드의 값들이
동일한 경우)
5.1 node.label 에 Class 필드의 값(지점) 저장
5.2 return (node)
6. end if
7. localTable의 Information을 구함
8. Table의 AP 필드마다 Entropy를 구함
9. Gain 값이 가장 큰 AP 필드를 구함
10. localMacList에서 Line 8에 해당하는
필드명(MacAddress)을 node.label 에 저장
11. localMacList에서 Line 8에 해당하는
필드명이 삭제된 subMacList 생성
12. loop(i=0; i<interval 수; i++)
12.1 subTable 생성 // Gain 값이 가장 큰
AP 필드의 해당 interval 별로 생성
12.2 if(subTable의 레코드 수 == 0)
12.2.1 새로운 노드(terminal) 생성
12.2.2 terminal.label 에 localTable의
Class 필드의 값(지점)들을 비율로 저장
12.2.3 node.next[i] = terminal
12.3 else
12.3.1 node.next[i] = // 재귀호출
CreatDecisionTree(subTable, subMacList)
12.4 end if
13. end loop
14. return (node)
end CreatDecisionTree
    
```

표 3. 현재 위치를 판정하는 알고리즘

```

algorithm Locate(NODE node,
    APListType apList)
//의사결정트리를 이용하여 판정된 위치를 반환
사전조건 : node는 의사결정트리이다. apList에는
실시간 단계 시 측정된 AP들의 MacAddress와
신호 세기가 저장되어 있다.
사후조건 : 판정된 현재 위치를 반환한다.

1. loop(i=0; i<=apList의 필드 수; i++)
1.1 if(node.label 이 MacAddress일 경우)
1.1.1 rssi = apList에서 node.label에
해당하는 신호 세기 저장
1.1.2 interval = rssi가 속한 interval
구함
1.1.3 node = node.next[interval]
1.2 else if(node.label 이 단일 지점일 경우)
1.2.1 return (node.label)
    
```

```

1.3 else
//node.label이 지점에 대한 비율일 경우
1.3.1 return (node.label)
1.4 end if
2. end loop
end Locate
    
```

실시간 단계 때 현재 위치를 판단하기 위해서는 그림 2의 이벤트 흐름도에 나타난 것처럼 'CreatTree' 버튼을 클릭하여 의사결정트리를 생성한다. 이 작업은 실시간 단계를 수행할 때 최초 한번만 수행하면 된다. 이후 'Position' 버튼을 클릭하게 되면 현재 위치에서 특정 데이터를 측정하고, [표 3]의 의사결정트리를 참조한 위치 판정 알고리즘을 이용함으로써 현재 위치를 판정하게 된다.

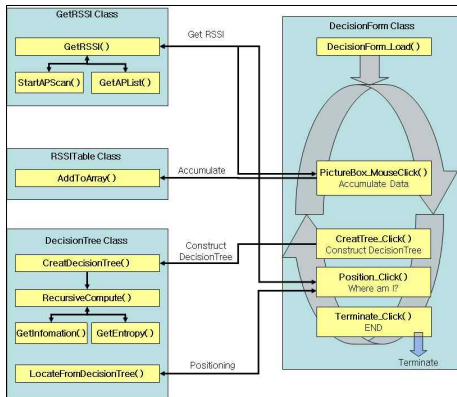


그림 2. 측위 프로그램의 이벤트 흐름도

[그림 3]은 의사결정트리를 이용한 측위 프로그램의 실행 화면이다. 이 측위 프로그램은 해당 장소의 지도정보가 저장된 DXF 캐드 파일을 읽어서 화면에 출력하고 있으며, 출력 화면을 확대, 축소 및 이동 작업을 수행할 수 있다. 여기서 지도정보를 반영하는 시스템은 이미 연구한 [12]에 설명되어 있다.

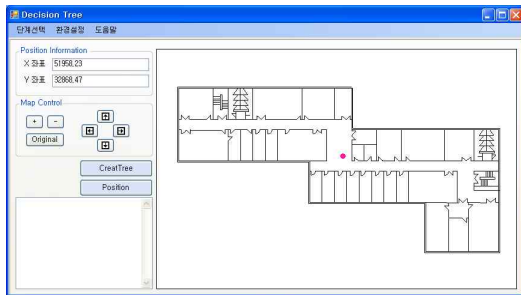


그림 3. 측위 프로그램의 실행 화면

IV. 실험

본 장에서는 실험 결과를 바탕으로 제안하는 방법의 정확도를 보인다. 제안하는 옥내 측위 방법은 준비 단계 시 후보 지점별로 Lab-Top 컴퓨터가 주변에 설치된 AP에 대하여 특정 데이터(신호 세기)를 측정하여 Training Data를 수집해 둔다. 실시간 단계에서는 수집된 Training Data를 가지고 [표 2]의 알고리즘에 따라 의사결정트리를 생성한다. 여기서 의사결정트리를 생성하는 작업은 실시간 단계에서 최소 한번만 수행한다. 이후 후보 지점들 사이의 임의의 지점에서 AP의 신호 세기를 측정하고, [표 3]의 현재 위치를 판정하는 알고리즘을 이용하여 의사결정트리로부터 임의의 지점을 판정하게 된다.

1. 실험 환경

제안하는 측위 프로그램의 정확도를 보이기 위해 [그림 4]에 보이는 동국대학교 경주캠퍼스 자연과학관 4층에서 실험을 실시하였다. 여기에는 총 8개의 AP가 '★'로 표시된 지점에 설치되어 있으며, 실제 실험에 사용된 AP는 3층과 5층을 포함하여 실험 지역에서 신호 세기가 측정되는 총 14대를 대상으로 하였다.

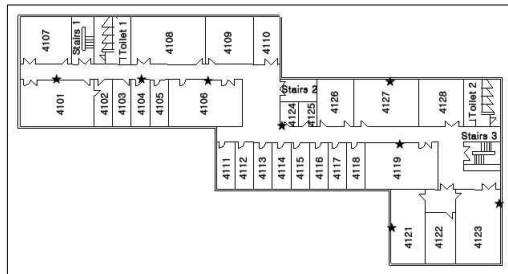


그림 4. 동국대학교 경주캠퍼스 자연과학관 4층

[그림 5]에 보이는 후보 지점들은 중앙 복도에서 1m 간격으로 선정하였고, 준비 단계에서 각각의 지점에 대하여 특정 데이터(신호 세기)를 130회씩 측정하여 Training Data를 작성하였다.

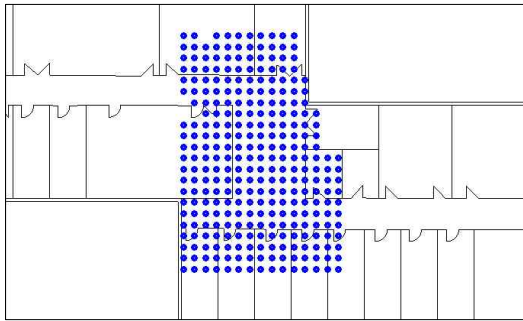


그림 5. 실험에 적용된 후보 지점들

2. 실험 결과

자연과학관 4층의 중앙 복도 영역에서 준비 단계 때 후보 지점별로 미리 측정된 Training Data를 가지고 실시간 단계에서 10개의 interval로 이산화하여 의사결정트리를 생성하면 [그림 6]과 같은 트리를 얻을 수 있다.

이 의사결정트리를 가지고 실시간 단계 때 [그림 7]처럼 후보 지점들 사이에서 원을 형성하도록 걸어가면서 1초 간격으로 신호 세기를 측정하여 현재 위치를 판정하는 실험을 한다. 실험은 [그림 7]의 왼쪽에 보이는 큰 원과 오른쪽에 보이는 작은 원에 대하여 각각 30회씩 수행하였다.

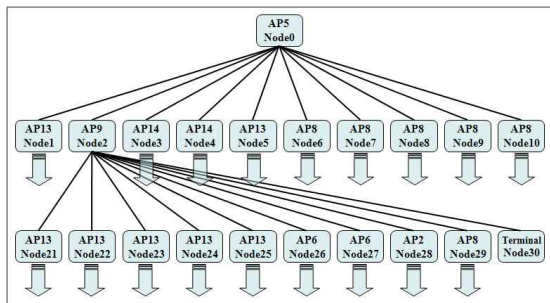


그림 6. Training Data를 이산화한 의사결정트리

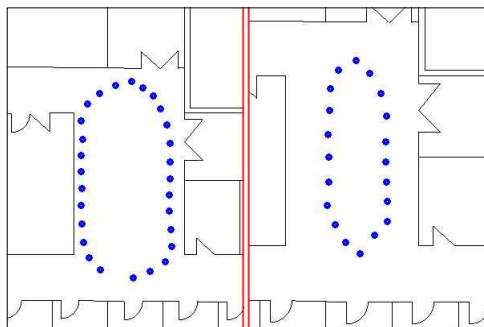


그림 7. 원을 형성하며 걸어 다닌 큰 원과 작은 원 지점

[그림 6]은 생성된 의사결정트리를 도식화한 것으로 트리의 루트에 해당하는 노드를 보면, 이 루트 노드의 label은 AP5에 해당하는 MacAddress가 저장되어 있고 루트 노드가 Linked List 하고 있는 자식 노드들은 10개임을 알 수 있다. 여기서 자식 노드가 10개인 이유는 의사결정트리를 생성할 때 AP5의 신호 세기를 Interval 수(10개) 만큼 그룹으로 구분하였기 때문이다. 즉, AP5의 신호 세기를 Interval 수만큼 그룹으로 구분하여, 각 그룹별로 자식노드를 생성하게 된다. 그리고 AP9 노드가 가리키고 있는 노드들 중 10번째 그룹에 속하는 노드는 Terminal 노드이다. 즉, 이 Terminal 노드는 더 이상 자식노드를 생성하지 않으며, Terminal 노드의 label에는 단일 후보 지점의 정보 또는 부모노드가 가질 수 있는 후보 지점들을 비율로 변환한 정보가 저장된다.

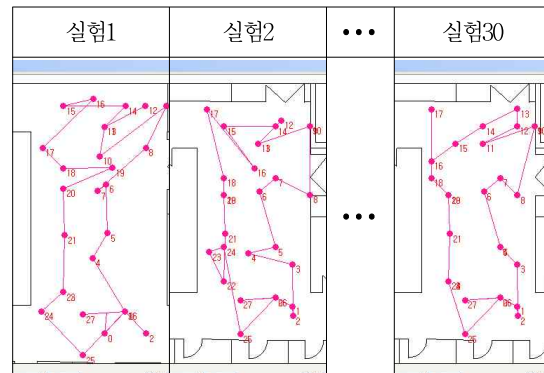


그림 8. 큰 원을 형성하도록 걸어가며 실험한 결과

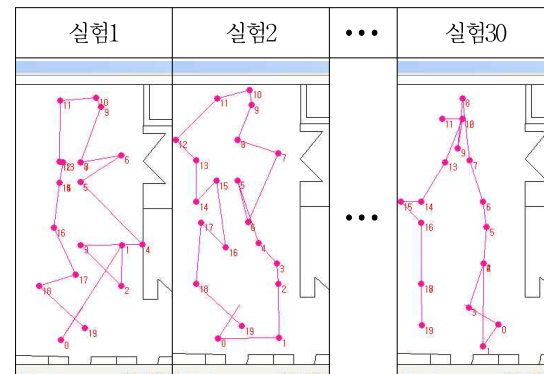


그림 9. 작은 원을 형성하도록 걸어가며 실험한 결과

[그림 8]은 큰 원을 형성하도록 걸어가면서 위치를 판정한 실험 결과이고, [그림 9]는 작은 원을 형성하도록 걸어가면서 위치를 판정한 실험 결과이다. 각 실험별로 판정 위치에 대한

오차들의 평균을 구하여, 큰 원 실험에 대한 평균 오차와 작은 원 실험에 대한 평균 오차를 구한 결과 큰 원일 경우 평균 오차는 1848.41mm이고, 작은 원의 경우 평균 오차는 1399.52mm이었다. 또한 각 실험들에 대하여 이동 패턴을 살펴 볼 수 있었다.

V. 결론

본 논문에서는 옥내 무선근거리통신망 환경에서 Fingerprint 방식으로 의사결정트리를 이용한 옥내 측위 방법을 소개하였고, 이동 패턴 정보를 추출하기 위한 실험 데이터를 수집하는 방법에 대하여 살펴보았다.

Fingerprint 방식에 의사결정트리를 적용한 옥내 측위 방법을 이용한 실험 데이터 수집을 위해 의사결정트리를 생성하는 알고리즘과 현재 위치를 판단하는 알고리즘이 적용된 측위 프로그램을 개발하였다. 이 프로그램은 지도정보가 저장된 DXF 파일을 읽어서 화면에 출력하게 되는데, 확대, 축소 및 이동 작업도 가능하다. 이 옥내 측위 프로그램은 준비 단계 때 후보 지점별로 AP의 신호 세기라는 특징 데이터를 수집하여 Training Data를 작성하며, 실시간 단계 때 앞서 작성된 Training Data를 가지고 최초 한번 의사결정트리를 생성한다. 이후 현재 위치에서 새로운 특징 데이터를 측정하여 생성된 의사결정트리로부터 현재 위치를 판단하게 된다. 실험결과로 후보 지점들 사이에서 큰 원을 형성하도록 걸어 다니며 측위 프로그램을 수행할 경우 평균 오차는 1848.41mm가 나왔고, 작은 원을 형성하도록 걸어 다닐 경우엔 평균 오차는 1399.52mm가 나왔다. 또한, 본 연구에서 수행한 실험들에서 미리 수집된 실험 데이터를 바탕으로 원을 형성하며 이동할 때의 이동 패턴들을 살펴볼 수 있었다.

향후에는 이러한 이동 패턴을 정의할 수 있는 연구가 진행되어야 하며, 실험에 사용된 후보지점의 영역을 자연과학관 전체로 확대하여 실험을 수행하고자 한다. 또한 임의의 위치에서 판정된 위치를 실제 위치와 근사하도록 보정방법을 연구하고자 한다.

참고문헌

- [1] Marwa Mabrouk, OpenGIS Location Services (OpenLS): Core Services OGC 03-006r <http://www.opengis.org/>
- [2] Virrantaus, K., J. Veijalainen, and J. Markkula, "Developing GIS-Supported Location-Based Services", Proceedings of the Second International Conference on Web Information Systems Engineering, 2001, Vol. 2, Dec. 3-6 2001, pp.66-75.
- [3] Adam Smith, HariBalakrishnan, Michel Goraczko, and Nissanka Priyantha, "Tracking Moving Devices with the Cricket Location System", Proc. of MobisSYS'04, June 2004.
- [4] R. Want, A. Hopper, V. Falco and J. Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems 10, pp. 91-102, Jan. 1992.
- [5] Wann, C. & Lin, M. (2004), "Data fusion methods for accuracy improvement in wireless location systems", Proceeding of the IEEE Wireless Communications and Networking Conference, 2004. (WCNC 2004) Vol. 1, 21-25 March 2004, pp. 471-476.
- [6] Youssef, M., Agrawala, A. & Shankar, A.U. (2003), "WLAN Location Determination via Clustering and Probability Distributions", Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003, 23-26 March 2003, pp. 143-150.
- [7] Youssef, M. & Agrawala, A. (2004), "Continuous Space Estimation for WLAN Location Determination Systems", Proceedings of 13th International Conference on Computer Communications and Networks, 2004 (ICCCN 2004), pp. 161-166.
- [8] Yim, J., "Introducing a decision tree-based indoor positioning technique", Expert Systems with Applications, Vol. 34, Issue 2, 2008, pp. 1296-1302.
- [9] Ciurana, M., Barcelo-Arroyo, F., Izquierdo, F., "A ranging system with IEEE 802.11 data frames", 2007 IEEE Radio and Wireless Symposium, 9-11 Jan. 2007, pp. 133 - 136.

- [10] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based User Location and Tracking system", in proceeding of IEEE INFOCOM 2000, 2(3), pp. 775-748, March 2000.
- [11] 임재걸, 심규박, 정승환, "RSSI 판독 라이브러리 함수 및 옥내 측위 모듈 구현", 한국 멀티미디어학회 논문지 10(11), pp. 1483-1495, 2007년 11월.
- [12] 임재걸, 주재훈, 정승환, "위치기반서비스를 위한 지도 정보가 반영된 옥내측위통합 시스템," 한국정보시스템 학회 논문지 17(1), pp. 131-153, 2008년 3월.