

# 실내 공간 이동객체의 궤적 데이터의 생성

## Generation of the Moving Object Trajectories on Indoor Space

강혜영\*, 이기준  
Hye-young Kang\*, Ki-joune Li  
부산대학교 정보컴퓨터공학부  
{hyezero\*, lik}@pnu.edu

### 요약

이동컴퓨팅과 센서, GPS 기술의 발달에 따라, 실외뿐만 아니라 실내 공간에서 이동객체의 위치 정보를 이용하는 다양한 응용분야들이 개발되고 있다. 그 중에서 이동 객체의 궤적을 이용하여 분석하는 응용분야들의 경우, 실제의 데이터를 획득하는 것이 불가능하거나 이용이 제한되어 있어서, 실제의 속성을 반영한 합성 데이터를 이용하고 있다. 그러나 대부분의 이동 객체 궤적 생성기는 실내 공간에서 이동하는 객체의 특성을 잘 반영하지 못한다. 이에, 본 논문에서는 실내 공간에서 이동하는 객체의 특성을 살펴보고, 이를 반영한 실내 공간 이동 객체의 궤적 생성기를 구현한다.

### 1. 서론

이동 컴퓨팅과 센서, GPS기술이 발달함에 따라, 실외뿐만 아니라 실내 공간에서도 이동 객체의 위치정보를 이용하는 다양한 응용 시스템들이 개발되고 있다. 이러한 응용 시스템들은 대량의 이동 객체를 관리하거나 이용하므로, 성능측정과 개선점 발견을 위해 실 데이터에 대한 충분한 성능분석이 이루어져야 한다. 그러나 이러한 실 궤적 데이터를 위한 분석, 가공에 대한 비용이 비쌀 뿐만 아니라, 실 궤적 데이터를 수집하는 것도 어렵고, 그에 대한 신뢰성을 보장하기 또한 어렵다. 따라서 실 데이터를 대신할 수 있는 현실과 매우 유사한 합성 데이터를 생성하는 하는 데이터 생성기가 필요하다. 기존의 몇몇 연구들 중에서 이러한 필요성을 인식하여 위치 데이터 생성에 관한 방법들이 제시되었다. 기존의 데이터 생성기들은 이동객체의 활동영역을 유클리디안 공간으로 설정한 연구가 대부분이며, 실제의 이동객체의

특성을 고려한 데이터 생성기들은 주로 차량 데이터를 생성하기 위한 도로 네트워크 공간을 가정하고 있다. 그러나 실내 공간은 유클리디안 공간 또는 도로 네트워크 공간과는 다른 속성을 가지는 공간이다. 따라서 기존의 데이터 생성기들은 실내 공간에서 이동하는 이동 객체의 궤적을 생성하기에 부적합하다.

본 논문에서는, 실내공간이 가지는 속성과 실내공간에서 이동하는 이동 객체의 궤적의 특성들에 대하여 살펴본다. 또한, 셀로 구성된 실내 공간에서 움직이는 이동 객체의 궤적을 표현하기 위한 방법과 이동 규칙을 소개하고, 이러한 규칙을 기반으로 이동객체의 궤적을 생성하는 궤적 생성기를 구현한다.

### 2. 관련 연구

본 장에서는 이동객체의 위치 표현방법과 이동객체의 위치 데이터 생성에 관한 기존의 연구에 대하여 살펴보도록 한다.

## 2.1 실내공간에 존재하는 이동객체의 위치 표현방법

기존의 네트워크 기반의 연구들은 이동객체의 위치를 2차원 또는 3차원 좌표, 즉 유클리디안 기반으로 표현하였다[1][2]. 그러나 실내 공간에서 실외에서와 같은 정확한 좌표 값으로 객체의 위치를 표현하는 것은 어렵다. 이에, 실내 공간에서 이동 객체의 위치를 표현하기 위한 연구가 진행되었다. APIT[3]은 이동 객체의 좌표를 계산하지 않고, 방 내부에 이동 객체의 존재 여부를 결정할 수 있다. [4]에서는 실내 공간을 좌표가 아닌 심볼로 표현하는 심볼릭 위치 모델들을 제시하였다. [4]에서 제시한 심볼릭 위치 모델들은 모두 실내 공간을 표현하기 위한 것으로, 실내 공간에 존재하는 이동객체의 위치를 표현하는 방법을 제시하지는 않았다.

## 2.2 이동객체 위치 데이터 생성기

이동 객체 위치 데이터 생성에 관한 기존 연구로는 정규분포, 가우스 분포 등을 이용하여 임의의 데이터를 생성하는 GSTD (Generation SpatioTemporalDataset)[5], 실제 시나리오를 고려한 Oporto[6], 그리고 네트워크를 따라 이동하는 이동객체를 생성하는 Brinkhoff 데이터 생성기[7]가 있다.

GSTD[5]는 이동객체들의 위치 데이터 생성에서 고려되어야 할 여러 가지 매개변수-공간분포, 데이터 개수, 이동 정도-를 사용자로부터 입력받아 계산한다. 또한, 학교와 빌딩 등과 같은 기반 구조물을 고려한 데이터 생성이 가능하다. 그러나 GSTD로부터 생성되는 이동객체들은 특정 분포를 따르는 이동형태를 가지게 된다. 따라서 특정 분포를 따르는 이동 형태 이외의 이동 객체 속성을 고려한 데이터를 생성할 수 없다.

Oporto[6]는 고깃배가 폭풍우 지역은 피하려고 하는 반면 고기떼를 찾아 움직이고, 고기떼들은 플랑크톤이 풍부한 영역을 향해 움직인다는 움직임의 규칙을 보고 착안한 이동 객체 궤적 생성 방법이다. Oporto는 이동 객체를 위한 실제의 속도,

방향, 분산, 크기 등의 변수를 사용하여 다양하고 실제적인 시나리오를 생성할 수 있으며, 이를 시간의 흐름에 따라 보여준다. 그러나 Oporto는 실내공간에 존재하는 구조물과 같은 지형지물들에 의해 이동에 제한이 주어지는 공간을 고려하지는 않는다.

Brinkhoff[7] 데이터 생성기는 주어진 네트워크를 따라 움직이는 이동객체를 생성한다. Brinkhoff 데이터 생성기는 특히, 외부객체(external object)라는 방해 요소를 적용하여, 이 영역 안에 있는 이동 객체의 속력은 외부객체의 영향을 막아 감소하도록 구현되었다. 따라서 Brinkhoff 생성기를 통해 생성된 이동 객체의 움직임 패턴은 모두 일률적이고, 데이터의 이동 속도는 실제 이동 객체의 평균속도나 네트워크 크기 등을 전혀 고려하지 않는다. 또한, 네트워크 기반으로 이동 객체의 위치를 생성하지만, 이동 객체의 위치는 유클리디안 기반의 표현을 이용한다.

기존에 제시된 여러 이동객체 위치 데이터 생성기들은 모두 유클리디안의 좌표로 이동 객체의 위치를 표현하고 있다. 그러나, 실제로 실내 공간에 존재하는 이동 객체의 위치는 정확한 좌표로 나타내는 것이 불가능하다.

따라서, 본 논문에서는 실내 공간에 존재하는 이동 객체의 위치 표현방법을 제시하고, 이동에 제한이 주어지는 실내공간에서 움직이는 이동객체의 궤적 생성 방법을 제시하고, 위치 데이터 생성기를 구현하였다.

## 3. 실내 공간의 이동 객체 생성기

본 장에서는 실내 공간의 이동 객체를 생성하기 위하여, 우선 본 논문에서 사용하는 실내공간에 대하여 정의하고, 이동객체의 궤적을 표현하는 방법을 제시한다.

### 3.1 실내 공간의 이동객체의 궤적 표현

실내 공간의 이동 객체의 궤적을 표현하기 위하여, 본 논문에서는 실내 공간에 존재하는 이동객체는 다음과 같은 속성을

가신다고 가정하였다.

- 주어진 실내공간은 셀의 집합으로 나누어져있다.
- 셀들 간의 겹침은 존재하지 않는다.
- 모든 객체들은 하나의 셀 안에 존재한다.
- 객체는 움직이지만, 모든 셀은 움직이지 않는다.

셀의 예로는 방, 복도, 계단 등이 있다. 본 논문에서는 엘리베이터와 같이 움직이는 셀의 경우는 제외하였다. 처음 2개의 조건을 만족하는 실내공간을 우리는 셀룰러 실내 공간(cellular indoor space)라고 부른다. 그림1은 이러한 가정을 만족하는 셀룰러 실내공간의 예를 나타낸다. 이동객체에 대한 4가지의 가정에 따라, 이동객체의 위치는 이동객체가 위치한 셀의 식별자로 표현할 수 있다. 이것과 유사한 셀과 공간의 개념이 Active Badge location system[8]에서 소개되기도 하였다.

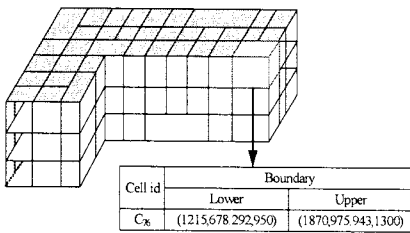


그림 2. 셀룰러 실내공간의 예

앞에서 제시한 가정을 기반으로 셀룰러 실내공간의 이동객체  $m$ 의 궤적은 다음과 같이 표현된다.

$$TR(m) = \{(c, t) | c \in C\}$$

여기서  $c$ 는 이동객체  $m$ 이 시간  $t$ 에 위치하고 있는 셀을 나타낸다.

### 3.2 이동 객체 궤적 생성기 설계

셀룰러 실내공간에 존재하는 이동객체를 생성하기 위하여, 다음과 같은 점을 고려하였다.

- 이동객체는 반드시 계단을 통해서 다른 층으로 이동한다.
- 이동객체는 문을 통해서 다른 셀로 이동한다.
- 계단에서 다른 층으로 이동하면 처음 만나는 곳은 복도이다.

그림 2는 이러한 이동 객체의 움직임의 특성을 간단하게 도식화 한 것이다.

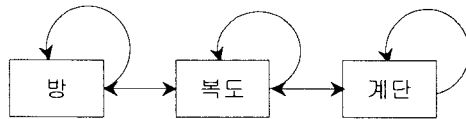


그림 3 . 이동 객체의 움직임

셀룰러 실내 공간에 존재하는 이동 객체의 위치는 시간의 흐름에 따라 결정된다. 본 연구에서는 궤적 생성을 하나의 시점에서 다음 시점으로 흐를 때, 각각을 상태로 보아 이전의 상태의 영향으로 다음 상태를 결정하는 state machine과 같이 생각하였다. 이동객체의 각 상태를 결정하기 위하여 적용하는 규칙은 다음과 같다.

#### 규칙 1.

방에 존재하는 이동객체는 일정시간 이상 동일한 공간에 머문다.

#### 규칙 2.

복도에 존재하는 이동객체는 목적지에 도착할 때까지, 다른 방에 들어가지 않는다.

#### 규칙 3.

목적지와 현재 위치가 서로 다른 층일 경우, 항상 가장 가까운 계단으로 이동한다.

또한, 이동 객체 궤적 생성기를 통해 생성되는 이동객체의 이동 속도는 시속 4km 이내로 한정 하였다.

## 4. 구현

### 4.1 셀룰러 실내 공간 지도 생성

본 연구에서는 셀룰러 실내 공간 기반의 궤적 데이터 생성이 목적이므로, 실제 건

물의 건축 도면으로부터 이동 객체의 이동에 제약을 주는 구조물인 벽, 계단, 문을 이용하여 2차원 실내 공간 지도를 생성하였다. 그림 3은 이러한 방법을 생성한 2차원 평면 실내 공간 지도의 예이다.

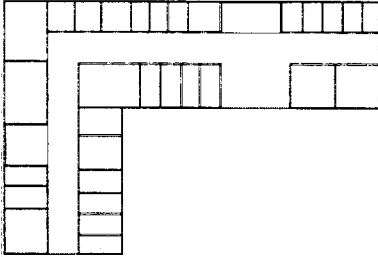


그림 3. 실내 공간 지도

이렇게 생성된 2차원 평면 실내 공간을 셀룰러 실내공간으로 표현하기 위하여, 건물의 구조적 정보를 이용하여 셀 정보를 생성한다. 셀의 종류는 공간의 용도에 따라 방, 복도, 계단으로 나누었다. 특히, 복도와 같이 긴 공간은 임의로 여러 개의 분할된 셀로 나타내었다. 긴 공간을 여러 개의 셀로 분할하는 것은 또 다른 하나의 큰 이슈가 될 수 있지만, 본 연구에서는 다루지 않는다.

좀 더 의미 있는 이동객체의 궤적을 생성하기 위하여, 셀룰러 실내공간에 존재하는 셀 간의 연결성을 나타내는 네트워크 데이터를 생성하였다. 셀 간의 연결성을 나타내는 네트워크 데이터를 이용하여 이동객체의 이동 경로를 결정할 수 있다.

#### 4.2 생성 알고리즘

본 연구의 데이터 생성기는 다음과 같은 흐름을 가진다. 먼저 사용자로부터, 전체 객체수와 객체가 존재하는 시간 간격인  $t_{max}$ 를 입력받는다. 생성을 실행하면 전체 객체 수만큼 이동객체를 생성한 후, 객체들의 출발지와 도착지를 정한다. 출발지가 방일 경우, 현재 시간을 *enterTime*에 저장한다. 생성된 이동객체들은 객체 리스트에 추가한다. 이동객체를 모두 생성한 후에는 시작 상태에 대하여 파일에 기록한다. 그리고, *time stamp*가  $t_{max}$ 가 될 때까지, 각

객체에 대하여 각 상황에 맞는 규칙을 적용하여 새로운 위치를 계산하여 할당하는 것을 반복한다. 이 과정을 알고리즘 1에 기술하였다.

---

#### 알고리즘 1 : DataGenerator

---

```

 $t_{max}$  : LifeSpan,
input  currentObjList : current Objects List
       samplingtime : period of file writing

Begin
1   $i \leftarrow 0$ 
2  while  $i < newObjNum$  do
3    newMO  $\leftarrow$  new MObject( )
4    newMO.startPosition  $\leftarrow$  MakeRandomPosition( )
5    newMO.endPosition  $\leftarrow$  MakeRandomPosition( )
6    currentCell  $\leftarrow$  FindPosition(newMO)
7    if currentCell.type == ROOM
8      newMO.enterTime =  $t$ 
9    end
10   currentObjList.Add(newMo)
11    $i \leftarrow i + 1$ 
12 end
13 WriteFile(currentObjList);
14  $t \leftarrow 0$ 
15 for  $t < t_{max}$  do
16   for  $obj \in currentObjList$  do
17     FindNextCell(obj, $t$ )
18   end
19   if  $t \% samplingtime == 0$ 
20     WriteFile(currentObjList);
21   end
22    $t \leftarrow t + 1$ 
23 end
End

```

---

네트워크 기반의 환경에서 이동 객체의 출발지와 도착지 사이의 경로를 구하는 것은 또 다른 하나의 큰 이슈가 될 수 있다. 특히 실내 공간의 경우, 층간의 이동 방법에 따라 실제의 경로가 다양하게 나타날 수 있기 때문이다. 하지만, 본 연구에서는 이에 대해서는 깊이 다루지 않고, 층간의 이동이 필요할 경우, 현재 위치에서 가장 가까운 계단을 이용하는 것으로 가정하였다.

알고리즘 1에서 객체의 위치를 결정할 때, 상황에 맞는 생성 규칙을 실행하는 *FindNextCell*을 호출하는 것을 보였다. 이에, *FindNextCell*의 상세 알고리즘은 알고리즘 2에서 기술하였다.

알고리즘 2 : FindNextCell

```

input  obj : current object,
       t : current time

Begin
1  nowCell ← FindCurrentCell(obj)
2  finalCell ← FindCell(obj.endPosition)
3  if nowCell.type == ROOM
4    if t - obj.enterTime > threshold
5      obj.enterTime ← -1
6      MoveToAdjacentCorridor(obj)
7    end
8  end
9  end
10 else if nowCell.type == CORRIDOR
11  if IsAdjacentCell(nowCell, finalCell)
12    && finalCell.type == ROOM
13    obj.enterTime ← 1
14    MoveToAdjacentRoom(obj)
15  end
16 end
17 else if nowCell.floor != finalCell.floor
18  MoveToNearestStair(obj)
19 end
20 end
21 else
22  MoveToAdjacentCorridor(obj)
23 end
24 end
25 else if nowCell.type == STAIR
26  if nowCell.floor != finalCell.floor
27    MoveToNearestStair(obj)
28  end
29 end
30 else
31  MoveToAdjacentCorridor(obj)
32 end
33 end
End

```

다음 셀을 선택 할 때는, 현재 위치한 셀의 종류에 따라 서로 다른 선택을 한다. 우선, 현재 존재하는 셀이 방일 경우, 방에서 머문 시간이 일정시간 이상이 될 경우에만 방과 인접한 복도로 이동하도록 한다. 현재 복도에 있을 경우, 목적지가 방이면서 현재 있는 복도와 인접해 있을 경우만, 방으로 이동한다. 그렇지 않으면, 현재 존재하는 층과 목적지 층이 다를 경우에는 가까이 존재하는 계단을 향해 이동한다. 그 이외의 경우에는 목적지와 가까워지는 방향의 인접한 복도로 이동한다. 마지막으로 계단에 있을 경우, 목적지와 동일한 층이 될 때까지 계단으로 이동하고, 목적지와 동일한 층일 경우에는, 인접

한 복도로 이동하도록 한다.

4.3 사용자 인터페이스

그림 3은 데이터 생성기의 사용자 인터페이스이다. 왼쪽은 화면보기의 옵션을 선택할 수 있는 체크박스와 사용자 변수 입력 입력창이다. 오른쪽 디스플레이 창은 실제 크기를 화면 크기에 맞게 정규화 시켰다. 본 연구에서 사용한 건물은 총 3층으로 이루어져 있으므로, 각층에 대한 도면을 각각 볼 수 있도록 왼쪽의 화면 보기 옵션에서 층을 선택할 수 있도록 하였고, 선택한 층의 도면이 오른쪽 디스플레이 창에 보인다. 3개의 사용자 입력 변수를 넣고 아래쪽의 Run버튼을 클릭하면 실제 데이터가 생성된다.

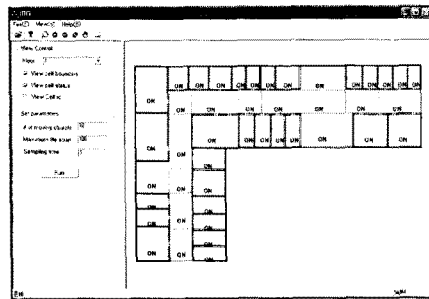


그림 4. 사용자 인터페이스

4.4 입력과 출력

사용자 입력은 표1과 같다. #of moving objects는 생성할 이동객체의 총 개수이다. maximum life span은 객체가 존재하는 총 시간이다. 즉, [0, t<sub>max</sub>)가 이에 해당한다. sampling time은 전체 데이터 생성기에서 파일을 기록하는 주기이다.

표 1. 사용자 입력 변수

변수 이름	설명
#of moving objects	이동객체의 개수
maximum life span	객체가 존재하는 총시간
sampling time	파일에 기록하는 주기

생성기의 출력은 텍스트 파일로, 시간별 이동 객체의 위치를 셀 아이디의 형태로

출력한다.

표 2에 그 내용을 나타낸다.

표 2. 파일 출력 변수

속성이름	설명
MOid	이동 객체 아이디
cell id	이동객체가 존재하는 셀아이디
Time	시간 ( <i>timestamp</i> )

### 5. 향후 연구 및 결론

실내 공간에 존재하는 이동객체는 건축 구조물에 의해서 이동의 제약을 받고, 이동 객체의 위치 또한 정확한 좌표로서 표현할 수 없다. 이에, 본 연구에서는 셀의 집합으로 실내공간을 표현하고, 셀의 시퀀스로 이동객체의 궤적을 표현하는 방법을 제안하였다. 또한, 이러한 셀룰러 실내 공간에 존재하는 이동객체의 위치 데이터 생성기를 설계하고 구현하였다.

그러나 현실 세계에서는 RFID와 같은 센서를 통하여 이동 객체의 위치를 결정하고, 수집할 수 있다. 이것은 실제로 이동 객체가 센서의 감지 반경 내에 존재할 경우에는 위치 정보의 수집이 가능하지만, 감지 반경 외에 있을 경우에는 수집이 불가능함을 의미한다.

향후 연구로는 RFID와 같은 센서의 감지 영역을 셀로 정의하여 데이터를 생성하도록 할 것이다. 더불어, 센서의 위치, 개수 및 속성을 조절할 수 있도록 하여 다양한 환경에서의 실내 공간 이동 데이터를 생성할 수 있는 데이터 생성기를 구현할 것이다. 또한, 본 연구에서는 이동객체에 대한 이동모형을 적용하지 않았으나, 향후 제약공간에서 이동하는 이동객체에 대한 이동모형을 적용하여 실 데이터와 유사한 데이터를 생성할 수 있도록 할 것이다.

### 감사의 글

본 연구는 국토해양부 첨단도시기술개발 사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C04)에 의해 수행되었습니다.

### 6. 참고문헌

[1] Ludger Becker, Henrik Blunck, Klaus Hinrichs, and Jan Vahrenhold. A framework for representing moving objects. In Proceedings of the 15th International Conference on Database and Expert Systems Applications, pp 854-863, 2004.

[2] Kathleen Hornsby and Max J. Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):177-194, 2002

[3] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek F. Abdelzaher. Range-free localization schemes for large scale sensor networks. In Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, pp 81-95, 2003.

[4] Christian Becker and Frank DÄurr. On location models for ubiquitous computing. *Personal and Ubiquitous Computing*, 9(1):20-31, 2005.

[5] Dieter Pfoser and Yannis Theodoridis. Generating Semantics-Based Trajectories of Moving Objects. In Proceedings of International Workshop on Emerging Technologies for Geo-Based Applications, pp 59-76, 2000

[6] Jean Marc Saglio and Jose Moreira. Oporto: A Realistic Scenario Generator for Moving Objects. *Geoinformatica*, Vol.5, No.1, pp.71-93, 2001

[7] T. Brinkhoff. A Framework for Generation Network-Based Moving Objects. *Geoinformatica*, Vol.6, No.2, pp.153-180, 2002

[8] Roy Wan, Andy Hopper, Vernica Falcão, and Honathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91-102, 1992