

# 가시성을 고려한 3차원 실내 보행자 시뮬레이션 모델

## An Indoor Pedestrian Simulation Model Incorporating the Visibility

곽 수 영\*

남 현 우\*\*

전 철 민\*\*\*

Su Yeong Kwak

Hyun Woo Nam

Chul Min Jun

**요 약** 실내 화재와 같은 재난, 재해시의 보행자의 행태를 모델링하거나 건축물의 구조를 분석하기 위해 지난 수십 년간 다양한 보행모델, 또는 화재대피모델들이 연구되어 왔다. 그러나 최근까지 개발된 모델은 대피 시 구조물들에 의해 보행자의 시야가 제한되는 것을 고려하고 있지 않다. 보행자의 시야는 대피에 영향을 미치는 중요한 요인 중 하나이므로, 이를 고려해야 현실적인 시뮬레이션 결과를 도출할 수 있다. 대피시뮬레이션에서 보행자의 시야에 대한 영향을 고려하는 방법은 시야의 제한 정도에 따라서 보행자의 대피 속도를 다르게 하는 것이다. 본 연구에서는 보행자의 시야에 따라 서로 다른 대피 속도를 갖게 하기 위해서 cellular automata를 이용한 floor field 모델을 기반으로 개선된 알고리즘을 제시하였다. 공간구문론(space syntax)을 활용하여 시야에 따라 공간을 분할하고, 동시다발적인 움직임 대신 분할된 공간별로 다른 이동속도를 갖게 하는 개선된 알고리즘을 구현하여 대피의 행태를 적절하게 모델링할 수 있게 하였다. 또한, 본 연구에서는 추후 실내 센서와의 연동을 통한 실시간 시뮬레이션 시스템으로의 개발을 위하여 공간DBMS를 이용한 3차원 보행자 시뮬레이터의 구현과정을 제시하였다. 캠퍼스 건물을 대상으로 개선된 알고리즘의 시뮬레이션 테스트를 수행하였다.

**키워드** : 보행시뮬레이션, CA, 3차원모델, 공간DBMS, 화재대피

**Abstract** Many pedestrian or fire evacuation models have been studied last decades for modeling evacuation behaviors or analysing building structures under emergency situations. However, currently developed models do not consider the differences of visibility of pedestrians by obstacles such as furniture, wall, etc. The visibility of pedestrians is considered one of the important factors that affect the evacuation behavior, leading to making simulation results more realistic. In order to incorporate pedestrian's visibility into evacuation simulation, we should be able to give different walking speeds according to differences of visibility. We improved the existing floor field model based on cellular automata in order to implement the visibility. Using the space syntax theory, we showed how we split the indoor spaces depending on the different visibilities created by different levels of structural depths. Then, we improved the algorithm such that pedestrians have different speeds instead of simultaneous movement to other cells. Also, in order for developing a real time simulation system integrated with indoor sensors later, we present a process to build a 3D simulator using a spatial DBMS. The proposed algorithm is tested using a campus building.

**Keywords** : Pedestrian Simulation, Cellular Automata, 3D Model, Spatial DBMS, Fire Evacuation

### 1. 서 론

지난 수십 년 동안 많은 마이크로 스케일의 보행

자 모델 또는 화재대피 모델들이 제시되어 왔다[2, 3, 14]. 하지만 기존의 모델들은 다양한 환경에 대한 파라미터들을 제공하면서 알고리즘을 개선해 왔지

<sup>†</sup> 이 논문 또는 저서는 공간정보 전문인력 양성사업 및 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제 (07국토정보C04)의 지원을 받아 수행된 연구임.

\* 서울시립대학교 공간정보공학과 석사과정 ksykk0@uos.ac.kr

\*\* 서울시립대학교 공간정보공학과 석사과정 nhw612@uos.ac.kr

\*\*\* 서울시립대학교 공간정보공학과 교수 cmjun@uos.ac.kr(교신저자)

만[15, 18], 보행자의 시야에 대한 영향은 고려하지 않고 있다. 보행자의 시야는 대피에 영향을 미치는 중요한 요소이기 때문에 이를 고려해야 보다 현실적인 시뮬레이션 결과를 기대할 수 있다. 대피 시뮬레이션에서 보행자의 시야에 대한 영향을 고려하는 방법은 시야의 제한 정도에 따라서 보행자의 대피 속도를 다르게 하는 것이다. 공간구문론(space syntax)을 활용하여 시야에 따라 공간을 분할하고, 동시다발적인 움직임 대신 분할된 공간별로 다른 이동 속도를 갖게 하는 개선된 알고리즘을 구현하여 대피의 행태를 적절하게 모델링 할 수 있게 한다. 본 연구에서는 기존 보행 모델에 보행자의 시야에 따른 영향을 추가한 개선된 보행 모델을 제시하였다. 기존의 floor field 모델에 기반을 두면서, 보행자의 위치로부터 출구까지 가시성의 정도에 따라 보행 속도를 조절할 수 있는, 개선된 floor field 모델을 제시하고자 한다.

최근에 실내 측위 기술에 대한 많은 연구들이 진행되고 있고[12, 16, 23, 25], 3D 모델의 연구와 함께 실내에서의 길 찾기나 화재 대피와 같은 실시간 응용에 대해 관심이 높아지고 있다. 그러나 기존의 모델들은 대부분은 실제계 응용보다는 알고리즘 개선에 초점이 맞추어져 있다. 기존의 보행 모델을 실시간 상황에 적용하기 위해서는 다음의 제약들이 해결되어야 한다. 기존의 학문적 연구나 상업적 응용에서 사용된 접근 방법들은 CAD와 같이 지리적인 좌표체계가 참조되지 않은 파일 기반의 데이터를 이용하였다[5, 17]. 이러한 접근 방법들은 시뮬레이션 목적으로는 적합하지만, 실시간 응용에 적용하기에는 한계가 있다. 실시간 응용에 필요한 client-server간의 데이터 연동을 위해서는 파일이 아닌, DBMS를 이용해야 효과를 높일 수 있다. 이를 해결하기 위해 본 연구에서는 DBMS와 3D 모델을 이용한 대피시뮬레이터의 구축 방안을 제시한다. 실제 건물의 실내 공간모델을 구축하고, 이를 공간 DBMS에 저장하여 대피시뮬레이션에 적용한다. 저장된 실내 공간 요소들을 시뮬레이션에 이용하게 하고 3차원 가시화가 가능한 데이터로 변환하는 방법을 제시한다. 본 연구에서는 가시성을 고려하고 3차원 DBMS를 이용하도록 개선된 시뮬레이터의 구축과정과 시뮬레이션 수행과정을 캠퍼스건물에 적용하여 예시하였다.

## 2. 관련연구

보행모델은 network flow, traffic assignment, simulation 등 다양한 분야에서 연구되어 왔으며[1], 대체로 macroscopic 모델과 microscopic 모델로 나누어진다[6]. Macroscopic 모델은 일반적으로 실외에서의 교통흐름 최적화나 길 찾기 등에 적용되며, 기본 데이터 구조로 node-link 구조를 갖는다. Macroscopic 모델이 개개인들의 개별성을 고려하지 않고 보행자를 노드나 링크에 할당될 수 있는 동질적인 그룹으로 보는 반면에, microscopic 모델은 개별적인 파라미터(예, 보행속도, 반응속도, 개인성향 등)와 다른 사람이나 물리적인 환경(예, 벽, 장애물, 연기 등)과의 상호작용을 고려한다. 최근에는 microscopic 모델에서 social force 모델과 floor field 모델이 주목을 받고 있다. Helbing이 제안한 social force 모델은 목적지까지(예, 출구) 이동할 때, 보행자에게 작용하는 모든 힘을 수학적으로 모델링한다[7, 8, 9]. Helbing의 모델에서는 모든 보행자 각각이 받는 영향과 물리적인 환경요인(예, 어깨폭, 기대속도, 목표점 등)이 고려된다. 이 모델은  $O(n^2)$ 의 연산 복잡도를 갖는데, 이는 컴퓨터 기반의 시뮬레이션 수행에는 매우 불리하게 작용한다[10, 11].

반면에, Kirchner와 그의 동료들이 제안한 floor field 모델은 cellular automata 기반의 접근방법을 사용한다. 대상공간에 있는 한 보행자가 모든 보행자들의 영향을 받는 대신에, 주위에 있는 보행자와의 상호작용만을 고려한다. 매 time step 마다 각각의 보행자의 이동을 계산하고, 인접한 셀 중에서 다음에 이동할 셀이 결정된다. 이렇게 한 순간에 주변 셀들만을 고려하는 연산은 컴퓨터 기반의 시뮬레이션에 유리하게 작용한다. 본 연구에서는 floor field 모델을 기본 모델로 이용하여 알고리즘을 개선하고자 하였다.

Kirchner의 floor field 모델은 static field와 dynamic field, 두 가지의 field를 이용한다. 이들은 보행자의 이동과 관련된 값들을 저장한 field들이다[13]. Static field에 저장된 셀 값은 출구까지의 최단 거리를 나타낸다. 보행자는 주변 셀의 값들을 통하여 가장 가까운 출구의 위치를 알게 되고, 이는 수치화되어 셀 값으로 저장된다. Static field가 물리적인 거리로 계산된 고정 값인 반면에, dynamic field는 동적으로 변화되는 값이다. Dynamic field는

보행자가 움직이는 동안에 주변 셀에 미치는 영향을 계산한 값이며 움직임에 따라 확산(diffuse)하고 소멸(decay)한다.

$$Score(i) = \exp(k_d D_i) \times \exp(k_s S_i) \times \zeta_i \times \eta_i$$

$Score(i)$ : 하나의  $i$ 번째 Cell의 선호도 값

$D_i$ :  $i$ 번째 Cell의 Dynamic Field 값

$S_i$ :  $i$ 번째 Cell의 Static Field 값

$k_d, k_s$ : 보행자 개개의 움직임과 출구의 위치를 나타내는 Scaling parameter

$\zeta_i$ : 이전 값을 가지며, 0은 이동 불가능한 Cell(ex. Wall, obstacles)을 나타내고 1은 그 외의 Cell을 나타냄

$\eta_i$ : 이전 값을 가지며, 0은 보행자에 의해 점유된 Cell을, 1은 점유되지 않은 Cell을 나타냄

위 수식에 포함되어 있는 scaling parameter를 통해 static과 dynamic field의 상대적인 영향의 정도를 조절할 수 있으며, 다양한 시나리오(예를 들어, 공포의 정도)를 실험할 수 있다. 이와 관련된 자세한 내용은 [15, 18]에 제시되어 있다.

본 연구에서는 세 번째 field로써 가시성 요소를 추가하여 floor field 모델을 개선하였다. Visibility field는 가장 가까이 있는 출구에 대한 보행자 시야의 제약에 따라 대피의 유리함 정도를 나타낸 값이다. 이와 관련된 세부적인 사항은 다음 장에서 자세하게 설명된다.

### 3. 가시성을 반영한 floor field 모델

Floor field 모델에서의 static field는 시뮬레이션이 시작되기 전에 출구와의 최단거리로 계산된 값이며, 시뮬레이션이 구동되는 동안 변하지 않는 값이다. 같은 공간에서 다른 위치에 있으나 출구까지 같은 거리에 있으면 static field의 값은 같은 값을 가지게 된다. 그러나 이것은 가구나 벽과 같은 장애물이 존재하여 출구까지의 시야가 방해받는 경우에 대해서는 고려하지 않고 있다. 출구까지의 시야가 제한될수록 보행자는 대피에 어려움을 겪게 될 것이다. 예를 들면, 그림 1에서 A와 B는 출구로 부터 같은 거리에 있다. 이때, 보행자 B는 장애물 뒤에 존재하기 때문에 출구를 바로 볼 수 없다. 즉, A보다 B가 출구까지 가는데 시간이 더 소요될 것이며, 특히 긴급한 상황에 낮은 환경이라면 탈출시간이 더 길어질 것으로 예상할 수 있다. 이에 착안하여

본 연구에서는 기존 floor field 모델에 가시성 필드(visibility field)를 추가한 모델을 소개한다. 이를 구현하기 위해서는 우선, 출구에 대한 가시성에 따라 대상 공간을 몇 개의 세부 공간으로 분할해야 한다. 그리고 시뮬레이션 중에 분할된 공간에 보행자들을 배치시키고 분할된 공간별로 보행자들이 서로 다른 이동속도를 갖게 해야 한다. 이는 다음 절에서 설명된다.

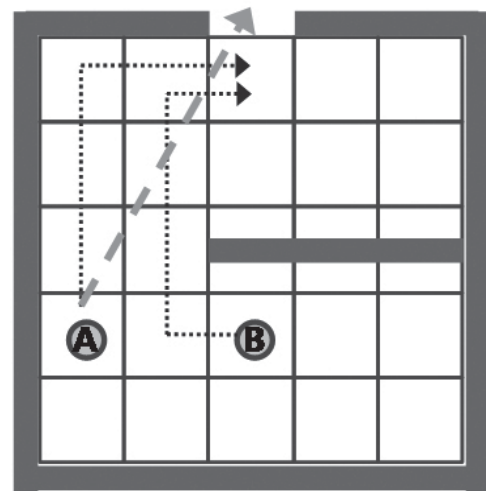


그림 1. 출구에 대해 서로 다른 가시성을 갖는 두 보행자의 예시

#### 3.1 보행자의 가시성을 고려한 공간 분할

보행자의 가시성을 고려한 공간 분할 기법으로 공간구문론(space syntax)을 응용하였다[4]. 공간구문론은 도시나 건축공간의 연결성과 접근성을 측정하는데 이용되는 기법이다[19]. 공간구문론에서는 이를 통합도(integration)라는 인덱스로 표현하며, 주요 cost로써 물리적인 거리는 고려하지 않고 구조적인 '깊이(depth)'만을 고려한다. 이 이론은 거리나 복도를 시축(visual path)으로 구성된 그래프(graph)로 변환한다. 깊이는 시축의 수 또는 시야의 굴곡성에 의해 결정된다. 깊이 값이 클수록 대상 공간의 구조적인 연결성이 좋지 않음을 의미한다.

전통적인 네트워크 모델은 graph  $G(V, E)$ 를 통해 정의된다.  $V$ 는 공간을 정의하는 노드들의 묶음  $\{v_i | i = 1, 2, \dots, n\}$ 이고  $E$ 는 그것들을 연결하는 edge나 link들의 묶음  $\{e_{ij} | e_{ij}, i, j = 1, 2, \dots, n\}$ 이며,  $e_{ij}$ 는 다음과 같이 정의된다.

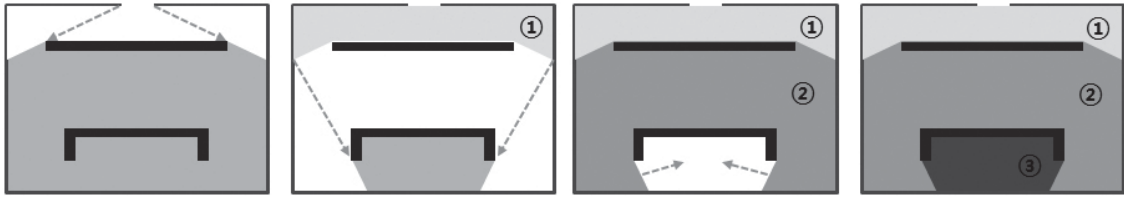


그림 2. 가시성의 정도에 따른 공간 분할

$$e_{ij} = \begin{cases} 1, & \text{if } \nu_i \nu_j \in E \\ 0, & \text{otherwise} \end{cases}$$

우리는 여기에서 노드  $i$ 에서 다른 노드  $j$ 를 가로 지르는 경로의 개수( $S_{ij}$ )를 알 수 있다. 이 값은 노드에서 다른 노드로의 깊이를 측정한 값이며 공간 구문론에서 보면 깊이에 기반을 둔 요소이다. 깊이  $S_{ij}^z$ ( $z$ 는 노드  $i$ 에서  $j$ 로의 깊이)는 다음과 같이 계산 된다.

$$\text{Let } S_{ij}^1 = e_{ij}, \text{ then,}$$

$$S_{ij}^z = \begin{cases} 1, & \text{if } \sum_k S_{ik}^{z-1} e_{jk} > 0 (k \neq i, k \neq j) \text{ and } \sum_t S_{ij}^t = 0 \\ 0, & \text{otherwise} \end{cases}$$

본 연구에서 우리는 visibility field를 구현하는데 이러한 깊이 기반의 접근 방법을 사용하였다. 대상 공간을 시야의 깊이에 따라 몇 개의 세부공간으로 나누었다. 그림 2는 출구까지의 시야를 고려하여 깊이를 부여하고, 이에 따라 공간을 분할하는 과정을 보여준다. 그림 2에서 보는 바와 같이 방 안에 두 개의 장애물이 있다고 가정하면, 출구가 바로 보이는 지역을 공간깊이 ①로 설정한다. 다음으로 공간 깊이 ①이 보이는 지역을 공간깊이 ②로 설정한다.

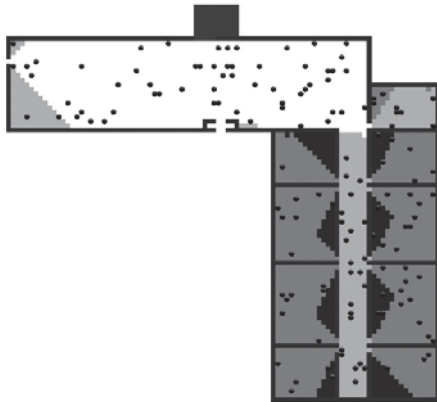


그림 3. 공간 분할의 예

같은 방법으로, 공간깊이 ②인 지역이 보이는 지역을 공간깊이 ③으로 설정한다. 이 과정은 전체공간이 세부공간으로 모두 분할될 때까지 재귀적으로 반복된다. 최종적으로, 분할된 지역들은 1, 2, 3, ...으로 공간깊이가 설정되고, 이는 출구까지의 가시성의 정도를 나타내게 된다. 그림 3은 실제 건물에 공간 분할 기법을 적용한 예이며 4개의 세부 영역으로 분할된 것을 확인할 수 있다.

### 3.2 Visibility field의 구현

전술한 바에 따라 공간깊이에 의해 visibility field가 구현되면, 보행자들은 static과 dynamic field 값을 참고함과 동시에 공간 깊이, 즉, visibility field값에 따라 서로 다른 보행속도를 가지면서 이동하게 된다. 본 연구에서는 공간 깊이에 따라서 보행자의 이동속도를 다르게 하는 기법을 구현하였다. 그림 4는 visibility field 계산 과정을 포함한 수정된 시뮬레이션 처리과정을 보여준다. 우리는 프로그래밍 언어에서 대부분 제공하는 time tick을 이용하여 보행속도를 조절하였다. Time tick은 보행자가 매 셀마다 이동할 때 대기하는 시간을 설정한 값이다. Time tick을 이용하여 visibility field의 값을 계산하기 위해서는 두 개의 파라미터가 필요하다. 첫째는 time tick의 시작 값이다. 이것은 보통 0으로 설정된다. 둘째는 time tick의 증가량이다. 증가량이 클수록 보행자는 다음 셀로 이동할 때의 대기시간이 길어지게 된다. 기존 모델에서는 매 time step마다 모든 보행자가 동시에 이동하는 반면에, 개선된 모델에서는 가시성이 반영되어 할당된 time tick에 따라 이동속도가 달라진다. 따라서 출구에 대해 가시성이 떨어지는 지역에 있는 보행자들은 대기시간이 더 오래 걸리게 된다. time tick의 값을 조정함으로써, 다양한 상황에 대한 시뮬레이션을 수행할 수 있다.

그림 5는 기존 floor field모델과 visibility field 추

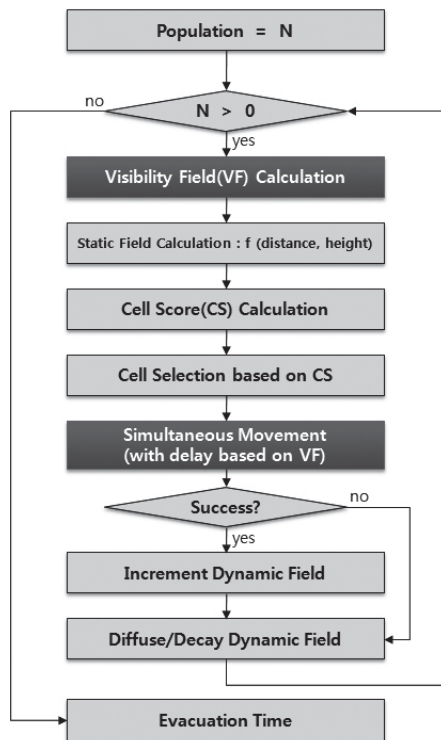


그림 4. Visibility field 연산을 포함한 시뮬레이션 처리과정

가를 통해 개선된 모델에서의 보행자 이동을 서로 비교한 것이다. 이것은 시뮬레이션이 가동된 후, 같은 시간에 캡처한 그림이다. Static field만을 고려한 그림 5의 왼쪽 그림을 보면 장애물의 양쪽 끝에서 병목현상이 나타난다. 이것은 장애물 뒤에 있는 보행자들이 출구로부터 동등하게 떨어져있는 위치일 때, 같은 속도로 이동하기 때문이다. 반면에, 그림 5의 오른쪽 그림은 공간깊이가 깊은 지역에 있는 보행자들이 공간깊이가 얇은 지역에 있는 보행자들보다 더 천천히 이동하는 것을 확인할 수 있다.

그림 6은 인원별로 time tick의 증가량에 따른 대피시간을 보여준다. 증가량이 0인 경우는 기존의 floor field 모델과 같게 된다. 우리는 대피인원의 숫자를 100, 200, 300, 400명으로 달리하여 시뮬레이션을 수행하였다. 그림 6의 하단 표를 보면, 기존 floor field 모델(증가량 = 0)은 인원별로 대피시간의 차이가 크지 않은 반면에 가시성(time tick)을 적용한 개선된 모델은 대피인원이 증가함에 따라 대피시간이 급격하게 증가하는 것을 볼 수 있다. 인원이 400명, 즉, 증가량이 3일 때, 총 대피 시간

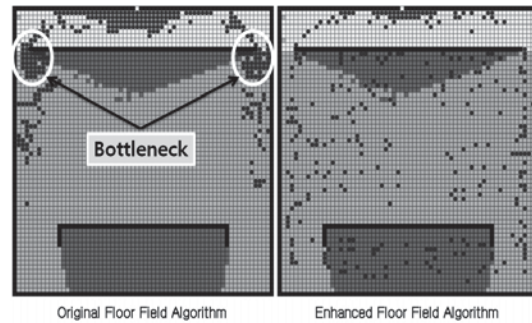


그림 5. 기존 floor field 모델과 visibility field가 추가된 모델의 비교

(810.5초)은 기존 모델의 대피시간(205초)의 4배에 이른다. 이것은 보행자들의 시야가 제한되어있는 상황에서는 대피에 보다 많은 시간이 소요된다는 것을 의미한다. 또한, 대피인원이 증가함에 따라 대피 시간도 급격히 증가하게 된다. 증가량을 조절함에 따라서, 보행자가 실내구조를 잘 알고 있는 경우(증가량이 낮음)와 낯선 공간에 있는 경우(증가량이 높음) 등 다양한 상황에 따른 시뮬레이션이 가능하다. 즉, 실내 구조를 잘 알고 있을 경우에는 구조물에 의한 굴곡(깊이)의 영향이 상대적으로 적을 것으로 예상할 수 있다.

#### 4. 공간 DBMS 기반 3D 시뮬레이터 구현

본 연구는 가시성을 추가하여 알고리즘을 개선하

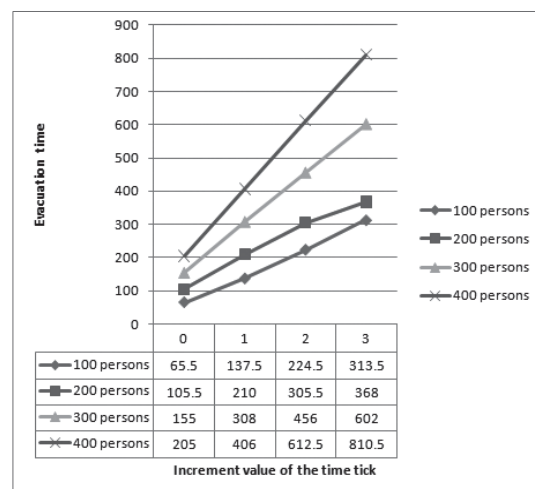


그림 6. 인원별 time tick의 증가량에 따른 대피시간



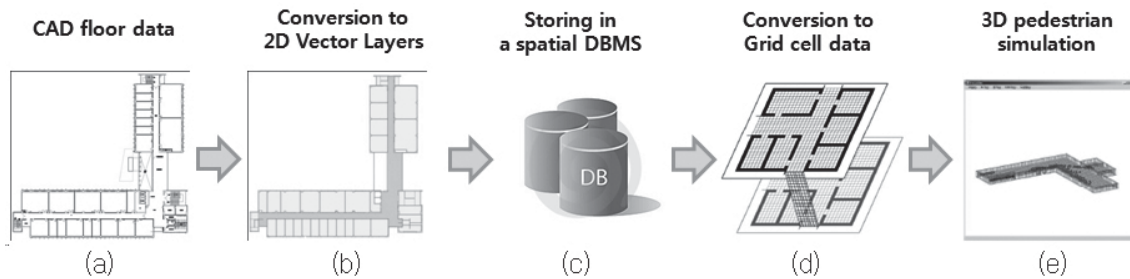


그림 7. 데이터 변환 및 시뮬레이션 구현 과정

는 부분이 중심을 이루지만, 추후 실내 센서와 결합된 실시간 시스템의 개발을 위하여 공간 DBMS 기반의 3차원 실내 대피 시뮬레이터를 구현하였다.

시뮬레이터의 구현과정은 2 단계로 나뉜다. 첫 번째는, 위상적으로 연결된 건물의 층과 계단의 공간 및 속성 데이터를 생성하고 DBMS에 저장하는 것이다. 두 번째는, 시뮬레이션의 입력 데이터인 그리드셀(grid cell) 데이터로 변환하는 과정이다. 그리드셀 데이터는 시뮬레이션의 연산과 3D 가시화에 이용된다. 구현과정은 그림 7에 요약되어 있다. 우선, CAD 형태로 된 건물의 층과 계단 공간데이터를 shapefile인 2D 벡터 레이어로 변환한다(그림 7-(a)와 (b)). 다음으로는 변환된 공간과 속성데이터를 공간 DBMS에 저장한다(그림 7-(c)). 그리고 저장된 데이터를 시뮬레이션에서 요구하는 데이터 형식인 그리드셀 데이터로 변환한다(그림 7-(d)). 마지막으로 시뮬레이터에서는 이 그리드셀 데이터를 이용하여 보행자 대피시뮬레이션을 수행한다. 또한 시뮬레이션 과정을 2D와 3D로 보여준다(그림 7-(e)). 보행자의 총 대피시간과 같은 시뮬레이션 결과는 로그(log) 파일로 기록되며 이는 DBMS에 저장되어 후에 대피분석 및 실시간 응용을 위한 자료로 활용된다.

#### 4.1 공간데이터의 생성과 저장

건물의 층 데이터는 CAD 파일을 통해서 얻을 수 있다. 하지만 CAD 데이터들은 기본적으로 지리참조가 되어있지 않고, 위상정보가 존재하지 않는다. 따라서 실시간으로 건물의 인원을 파악하여 저장하고 대피경로를 안내하는 등 실시간 응용에 적용하기에 한계가 있다. 위상적인 다층 데이터를 이용하기 위해서, 본 연구에서는 새로운 형식을 개발하기 보다는 기존에 존재하는 shapefile 형식을 사용하였

다. Shapefile은 많은 GIS 어플리케이션에서 처리할 수 있기 때문이다. 우리는 QuantumGIS[22]를 이용하여 데이터변환을 수행하였다. QuantumGIS에서 CAD 파일을 불러온 후, 문, 벽, 방, 복도, 출구 등의 정보를 폴리곤과 라인의 형태로 추출한다. 추출된 데이터는 shapefile형식으로 저장된다. 추출된 2D 레이어들은 공간 DBMS 테이블에 저장된다. 본 연구에서 데이터 저장에 이용한 DBMS는 PostgreSQL/PostGIS이다[20]. 데이터 저장에 사용한 DBMS 테이블의 구조는 그림 8에 나타나 있다. 방, 복도, 벽, 문, 출구 등의 공간 요소들은 각각에 대응하는 테이블에 저장되며, 건물의 층, 방 번호와 같은 속성데이터도 테이블에 저장된다. 위의 과정을 거쳐 모든 층과 계단 데이터를 DBMS에 저장한다.

계단 데이터는 층 데이터와 별도로 처리되는데, 이는 계단 데이터는 층 바닥과 같이 평면(plane) 구조를 이루고 있지 않아서 바로 CAD파일에서 변환하여 이용할 수 없기 때문이다. 다음 절에 설명하는

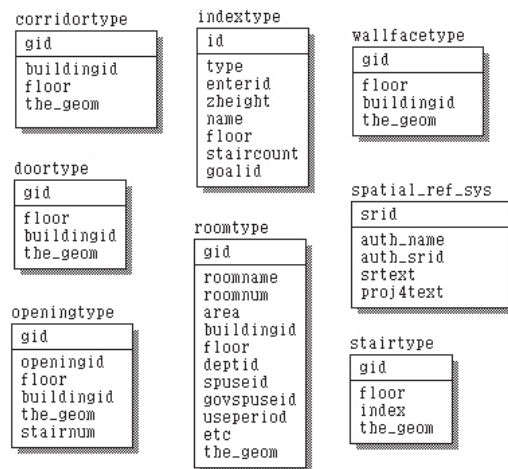


그림 8. 공간 DBMS에 저장된 테이블의 구조

바와 같이, 본 대피 시뮬레이션 시스템은 기본 데이터 구조로 평면 데이터를 필요로 한다. 그림 9는 계단 데이터의 변환과정을 나타낸다. 우선, 계단의 형태를 단순화시켜 몇 개의 폴리곤으로 분할한다. 그림 9에서 계단을 4개의 폴리곤으로 나눈 것을 볼 수 있다. 그리고 분할된 폴리곤들은 DBMS에 저장된다. 저장된 계단 데이터는 이후에 시뮬레이터에서 불러와 계단의 형태로 재조합하여 대피연산 및 가시화에 이용된다.

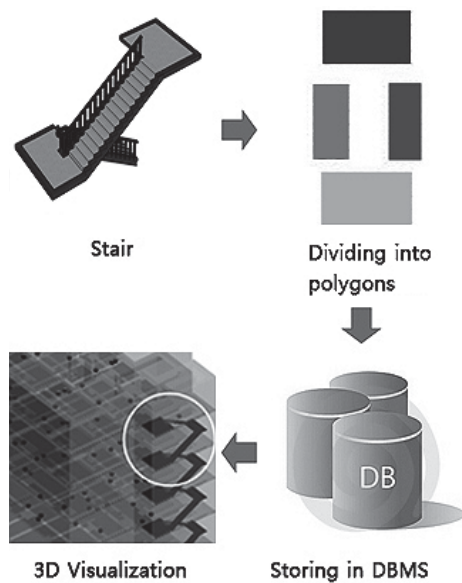


그림 9. 계단 데이터의 처리 과정

#### 4.2 그리드셀 데이터로 변환

본 연구는 전술한 바와 같이 floor field 모델을 기반 모델로 이용하였으며, 이 모델은 그리드셀 데이터를 통해 시뮬레이션을 수행한다. 본 연구에서 C#을 이용하여 시뮬레이션 시스템을 구현하였다. 또한, 공간데이터를 시스템으로 불러오기 위해서

PostgreSQL에서 지원하는 npgsql library[21]를 이용하였다. 그리드셀 데이터로 변환하는 과정은 그림 10에 나타나있다. 우선, DBMS에 저장된 건물데이터와 속성데이터를 읽어온다. 그리고 불러온 건물데이터에서 층과 계단데이터를 SharpMap library[24]를 이용하여 비트맵(bitmap) 형식으로 변환한다(그림 10-(a)). SharpMap library는 공간 DBMS에 저장된 공간데이터를 2차원으로 가시화하는 기능을 제공한다. 비트맵 형식으로 변환하는 이유는 동일한 래스터(raster)구조인 그리드셀로의 변환이 용이하고, 각각의 레이어를 서로 다른 RGB값을 부여하여 구분할 수 있기 때문이다. 비트맵 데이터에서 문, 방, 벽, 외부 공간 등의 레이어들은 각각 다른 색으로 지정되고, 부여된 RGB값을 기준으로 셀 값이 정해진다. 그림 10-(d)는 레이어별로 부여된 RGB 값과 그에 따른 셀 값을 나타낸다. 그 다음, 층과 계단 데이터의 좌표를 설정해주어야 하는데, 그리드셀 데이터가 좌표체계를 포함하고 있지 않기 때문이다. 본 연구에서는 1층의 좌상단 지점을 기준 좌표로 설정하였다. 이것을 기준으로 하여, 다른 층과 계단의 위치를 상대적으로 조절하였다(그림 11). 다음에 셀의 크기와 각 공간 요소별 셀의 개수를 산정한다. 우리는 40cm x 40cm를 셀 크기로 설정하였는데, 이는 사람의 어깨 폭을 고려한 값이다. 이를 이용하여 각 층과 계단별로 셀의 개수가 산정된다. 이러한 과정을 통해 모든 층과 계단 데이터는 그리드셀 데이터로 변환된다. 이러한 변환과정이 시스템 내에서 연속적으로 처리될 수 있도록 대피 시뮬레이션 시스템을 구현하였다.

#### 4.3 시뮬레이션 수행과 3차원 가시화

우선, 보행자들의 수를 설정하여 대상 건물에 임의의 위치에 배치한 후, 대피 시뮬레이션을 수행하고, 그 과정을 3차원으로 가시화하였다. 대상 건물

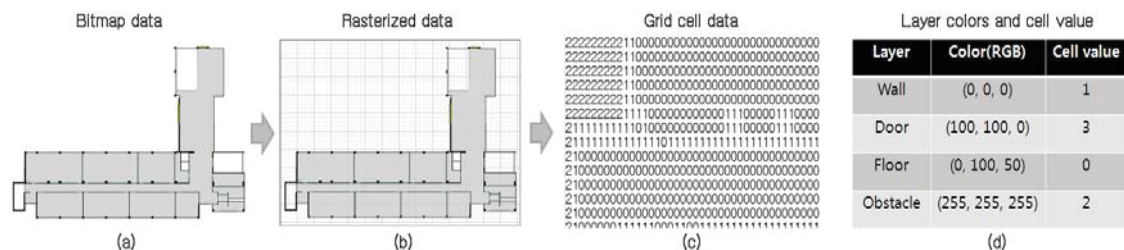


그림 10. 그리드셀 데이터 구축과정

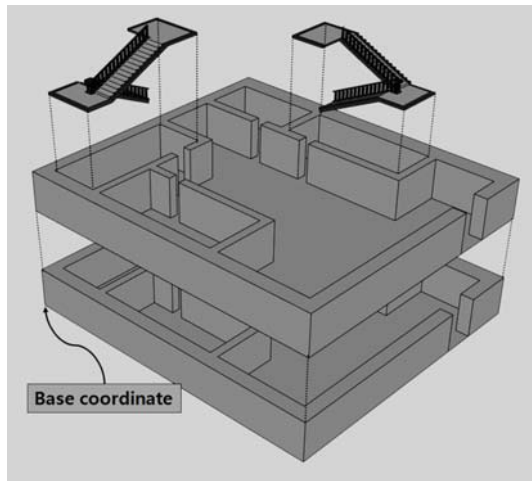


그림 11. 기준 좌표의 설정 및 층과 계단의 좌표 매칭

과 보행자의 가시화는 OpenGL Library를 사용하였다. 벽과 3차원 모양의 계단 모델링은 DBMS의 각 공간 속성으로 저장된 높이 값을 이용하였다. 계단은 기울기를 계산하여 계단의 형태가 되도록 연산하였고, 이를 3차원으로 가시화하였다.

## 5. 시스템 테스트

본 연구에서는 대학 캠퍼스 건물을 이용하여 시뮬레이션 시스템을 테스트 하였다. 전술한 처리과정에 따라서, CAD 형식인 건물 데이터를 shapefile로 변환하였고, 이를 공간 DBMS에 저장하였다. 시뮬레이션 시스템은 DBMS에 저장된 데이터를 불러와 그리드셀 형식으로 변환한다. 본 시스템은 그림 12, 그림 13과 같이 2D, 3D 보기를 지원한다. 본 시뮬레이션 시스템은 visibility field의 적용 여부를 포함한 다양한 파라미터들을 미리 설정할 수 있으며, 설정된 값들을 통해 시뮬레이션이 수행된다. 그림 14는 시뮬레이션의 결과를 저장한 로그 파일이다. 보행자가 빠져나간出口的 좌표와 보행자의 ID, 탈출시간 등의 정보가 기록되며, 이를 통해 전체 인원이 건물에서 탈출한 시간도 알 수 있다. 그림 14에 나와 있는 'TIMETICK'은 보행자가 대피에 소요된 시간을, 'AGENT ID'는 보행자의 ID번호를, 'EXIT POS'는 보행자가 대피한出口的 좌표를 나타낸다. 이 로그 파일은 다시 DBMS에 저장되어 관리된다. 결과파일을 통해 보다 다양한 분석을 할 수 있는데,

예를 들면, 건물 내에서 어떤 공간이 대피에 불리한지 분석할 수 있다.

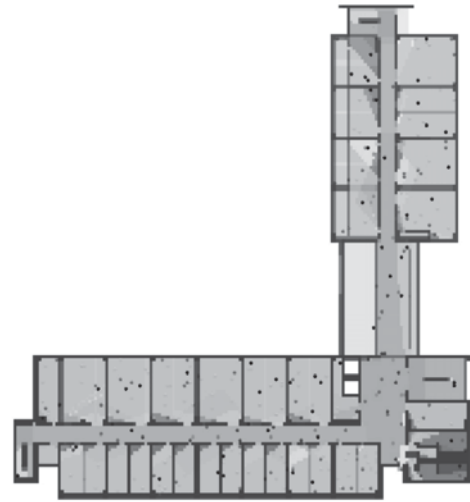


그림 12. visibility field에 의해 분할된 공간을 보여주는 2D 시뮬레이션

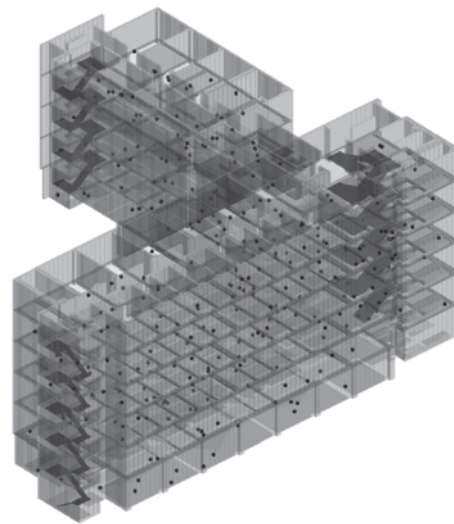


그림 13. 3D 보기를 지원하는 시뮬레이션

#	TIMETICK(SEC)	AGENT ID	ACTION	INIT POS	EXIT POS
3	(1.5)	32	OUT	(3,6,0)	(0,5,0)
3	(1.5)	248	OUT	(1,2,0)	(0,5,0)
4	(2)	54	OUT	(44,16,0)	(47,20,0)
4	(2)	223	OUT	(46,16,0)	(46,20,0)
6	(3)	36	OUT	(43,14,0)	(47,20,0)
7	(3.5)	202	OUT	(42,17,0)	(47,20,0)
7	(3.5)	256	OUT	(42,18,0)	(46,20,0)

그림 14. 시뮬레이션 결과를 기록한 로그 파일



## 6. 결 론

본 연구에서는 공간 DBMS를 이용한 3차원 실내 대피 시뮬레이터를 구현하였으며, 2가지가 중점적으로 다루어졌다. 첫째, 기존의 floor field 모델에 visibility factor를 추가하여 대피 알고리즘을 개선하였다. 출구에 대한 시야의 제한에 따라서 공간을 분할하였고, 분할된 공간에 따라 대피자의 이동속도를 조절하였다. 둘째, 건물의 데이터를 3차원으로 변환하여 공간 DBMS에 저장하고, 이를 시뮬레이션 시스템에서 불러와 그리드셀 형식으로 변환하여 시뮬레이션을 수행하는 과정이 연속적으로 이루어지도록 하였다. 본 연구에서 수행한 시뮬레이션의 결과를 추후에 실시간 시스템에 적용하기 위해서, 본 시스템을 실내 센서와 연계할 수 있도록 개선하고 있다. 현재 개발 중이지만, 추후에 본 시스템이 실내 센서와 통합이 될 경우, 건물 내 각각의 공간에 실시간으로 사람들이 몇 명이 있는지 알아내고, 이를 이용한 대피 시뮬레이션을 수행하여 대피 경로를 안내하는 실시간 인명 구조 등에 활용될 수 있을 것이다. 또한, 본 연구에서 제안한 visibility field 알고리즘의 검증(validation)과정을 수행하고 있다.

## 참 고 문 헌

- [1] Ahuja, R. K., Magnate, T. L., and Orlin, J. B., 1993, Network Flows: Theory, Algorithms and Applications.
- [2] Blue, V. J. and Adler, J. L., 1999, Using cellular automata microsimulation to model pedestrian movements, In A. Ceber (Ed.), Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, pp. 235-254.
- [3] Burstedde, C., Klauck, K., Schadschneider, A., and Zittartz, J., 2001, Simulation of pedestrian dynamics using a two-dimensional cellular automaton, Physica A, vol. 295, no. 3-4, pp. 507-525.
- [4] Cho, D., 1999, A Study on the Tools of Analysis in the Space Syntax Theory, Architectural Institute of Korea, vol. 156, pp. 71-76.
- [5] Gwynne, S., Galea, E.R., Owen, M., Lawrence, P.J., and Filippidis, L.L., 2005, A systematic comparison of building EXODUS predictions with experimental data from the Stapelfeldt trials and the Milburn House evacuation, Applied Mathematical Modeling, vol. 29, No. 9, pp. 818-851.
- [6] Hamacher, H. W. and Tjandra, S. A., 2001, Mathematical modeling of evacuation problems—a state of art. In M. Schreckenberg and S. Sharma, (Eds.), Pedestrian and Evacuation Dynamics, Springer-Verlag, Berlin, pp. 227-266.
- [7] Helbing, D., Farkas, I., Molnár, P., and Vicsek, T., 2001, Simulation of pedestrian crowds in normal and evacuation situations, In M. Schreckenberg and S. Sharma, (Eds.), Pedestrian and Evacuation Dynamics, Springer-Verlag, Berlin, pp. 21-58.
- [8] Helbing, D., Farkas, I., and Vicsek, T., 2000, Simulating dynamical features of escape panic, Nature, vol. 407, pp. 487-490.
- [9] Helbing, D. and Molnár, P., 1997, Self-organization phenomena in pedestrian crowds, In F. Schweitzer (ed.), Self-Organization of Complex Structures: From Individual to Collective Dynamics, Gordon & Beach, London, UK.
- [10] Henein, C. and White, T., 2005, Agent-based modeling of forces in crowds, Springer, vol. 3415, pp. 173 - 184.
- [11] Henein, C. and White, T., 2007, Macroscopic effects of microscopic forces between agents in crowd models, Physica A, vol. 373, pp. 694-712.
- [12] Hightower, J., Boriello, G., and Want, R., 2000, SpotON: An indoor 3D location sensing technology based on RF signal strength. University of Washington CSE Report.
- [13] Kirchner, A., and Schadschneider, A., 2002, Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics, Physica A, vol. 312, pp. 260-276.
- [14] Klupfel, H., König, T., Wahle, J., and Schreckenberg, M., 2002, Microscopic simulation of evacuation processes on passenger ships, In

- Proceedings of Fourth International Conference on Cellular Automata for Research and Industry, Karlsruhe, Germany.
- [15] Kretz, T. and Schreckenberg, M., 2006, Floor field- and Agent-based Simulation Tool, International Symposium of Transport Simulation, Lausanne, Switzerland.
- [16] Li, B. et al., 2006, Indoor positioning techniques based on wireless LAN, 1st IEEE International Conference on Wireless Broadband and Ultra Wideband Communication, Sydney, Australia, pp. 13-16.
- [17] Lo, S.M., Fang, Z., Lin, P., and Zhi, G.S. 2004, An evacuation model: the SGEM package, Fire Safety Journal, vol. 39, no. 3, pp. 169-190.
- [18] Nishinari, K., Kirchner, A., Namazi, A., and Schadschneider, A., 2005, Simulations of Evacuation by an Extended Floor Field CA Model, Traffic and Granular Flow '03, pp. 405-410.
- [19] Penn, A., B. Hillier, D. Banister, and Xu, J., 1998, Configurational modeling of urban movement networks, Environment and Planning B-Planning & Design, vol. 25, no. 1, pp. 59-84.
- [20] PostgreSQL, <http://www.postgresql.org/>.
- [21] PostgreSQL/npgsql, <http://pgfoundry.org/projects/npgsql/>.
- [22] QuantumGIS, <http://www.qgis.org/>.
- [23] Randell, C. and Muller H., 2001, Low cost indoor positioning system, Ubicomp 2001 Conference of Ubiquitous Computing, pp. 42 - 48.
- [24] SharpMap, <http://www.codeplex.com/SharpMap/>.
- [25] Zhou, R., 2006, Wireless indoor tracking system (WITS), In doIT Conference on Software Research, Verlag Heidelberg, Germany, pp. 163-177.
- [26] 전철민, 2010, “개선된 Floor Field 기반 보행 시뮬레이션 모델,” 한국공간정보시스템학회논문지, 제 12권, 제1호, pp. 76-84.
- [27] 전철민, 2007, “GIS 기반 Space Syntax를 이용한 대중교통 접근성,” 한국공간정보시스템학회논문지, 제9권, 제3호, pp. 25-33.

논문접수 : 2010.11.10  
수정일 : 2010.12.14  
심사완료 : 2010.12.17



#### 곽수영

2010년 서울시립대학교 공간정보공학  
공학사

2010년~현재 서울시립대학교 공간정  
보공학 석사과정

관심분야는 GIS, 공간분석, 공간데이  
터베이스, 3차원 모델링 등



#### 남현우

2010년 서울시립대학교 공간정보공  
학 공학사

2010년~현재 서울시립대학교 공간정  
보공학 석사과정

관심분야는 공간데이터베이스, 웹  
GIS, 모바일 GIS 등



#### 전철민

1988년 서울대학교 도시공학 공학사

1990년 서울대학교 도시공학 공학석사

1997년 Texas A&M University 도시  
및 지역계획학 공학박사

1997년~1999년 North Carolina RTI

GIS 전문요원

1999년~현재 서울시립대학교 공간정보공학과 정교수

2008년~2010년 서울시립대학교 공간정보연구센터 센  
터장

관심분야는 GIS, 공간 데이터베이스, 3차원 모델,  
network algorithm 등