

CA 모델을 이용한 GIS 기반 화재 대피 시뮬레이션

박인혜* · 전철민** · 이지영***

GIS-based Fire Evacuation Simulation using CA Model

Inhye Park* · Chulmin Jun** · Jiyeong Lee***

요 약

최근에는 대형건축물, 지하철과 같은 대규모 실내공간에서의 화재와 같은 사고에 대한 인식이 확대되고 있다. 그러나 현재까지는 그에 대한 적절한 서비스, 설계, 평가 방법이 미약한 실정이며 관련 연구들도 학술적 연구차원이거나 가상의 시뮬레이터 중심인 경우가 대부분이다. 최근 유비쿼터스 관련 분야에 대한 관심과 함께 실내 위치측위 기술 역시 발전하고 있는데, 이러한 기술을 이용할 경우, 방재분야 역시 단순 시뮬레이터에서 실시간 대피 안내시스템으로 확대시킬 수 있는 가능성이 있다. 그러나 현재 개발되어있는 대피 시뮬레이션의 경우에는 지리참조 되지 않은 건축도면을 기반으로 구축되어 있어서 이러한 실내 센서와 연동하기에는 한계가 있다. 본 연구에서는 이러한 실내 위치기반 실시간 방재시스템으로의 확대를 위한 기초연구로서, 데이터 구조 및 대피경로 산출 방법을 제시하고 이를 적용하여 대피시뮬레이션 테스트를 실시하였다. 좌표계가 적용된 GIS 데이터를 활용하는 방안을 제시하였고, 실내 이동자의 움직임을 나타내는 데에는 Cellular Automata(CA) 모델을 적용하였다.

주요어 : GIS, Cellular Automata, 화재대피시뮬레이션

ABSTRACT : With emerging technologies on wireless networks and mobile computing environment, a number of researches have been carried out for ubiquitous computing. An important functional requirement of ubiquitous computing is to handle location data with ease. With the increase of accidents in large complex buildings, move attention is being paid to indoor spaces and evacuation. However, most currently used evacuation-related applications are simulation based on hypothetical data. Also, since they use non-georeferenced CAD data, it is

*서울시립대학교 공간정보공학과 박사과정(ihpsm@uos.ac.kr)

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

***서울시립대학교 공간정보공학과 교수(jlee@uos.ac.kr)

not easy to integrate them with indoor positioning devices.

With the recent progress of indoor positioning systems, the simulators can be enhanced to real-time evacuation systems. As a preliminary stage to make such systems possible, this study proposes using a georeferenced data and evacuation simulation. This study used GIS data and Cellular Automata theory as the algorithm for the movement of the evacuee.

Keywords : GIS, Cellular Automata, fire evacuation simulation

1. 서론

최근 몇 년 사이 지하나 대형건축물, 지하철과 같은 대규모 실내공간에서의 사고를 경험하고 내재되어있는 사고에 대한 인식이 확대되면서 실내공간에 대한 관심과 중요성이 확대되고 있다. 우리나라와 같이 밀집된 도시환경에서는 대규모 복합공간이 점차 증가하고 그 활용도도 증가하고 있기 때문에 그 중요도가 더 높다고 할 수 있다. 그 예로 코엑스(COEX)나 킨텍스(KINTEX)와 같은 건축물을 들 수 있는데, 이러한 대규모 복합공간이 늘어나면서 실내 유동인구도 증가하게 되었고 그에 따라 실내공간과 관련된 서비스의 요구도 증가하게 되었다. 그러나 현재까지는 그에 대한 적절한 서비스, 설계, 평가 방법을 찾아보기 힘들고 현재까지 연구되어 온 방법은 학술적 연구이거나 가상의 시뮬레이터 차원에서 머무는 경우가 대부분이다.

최근 유비쿼터스 관련 분야에 대한 관심이 증대되면서 실내센서를 이용한 위치추위 및 이를 기반으로 한 응용분야에 대한 연구가 다양하게 시도되고 있다. 이와 같은 사회적인 요구와 기술적인 발달과

관련하여 실내 위치추위 기술을 이용할 경우, 방재분야 역시 단순 시뮬레이터에서 실시간 방재시스템으로 확대시킬 수 있는 가능성이 있다. 그러나 현재 개발되어있는 대피 시뮬레이션의 경우에는 대부분이 지리참조 되지 않은 건축도면을 기반으로 구축되어 있고 가상의 데이터로 시뮬레이션을 수행하는 형태이어서 실내 실시간방재시스템에 응용하는 데에는 한계가 있다.

따라서 본 연구에서는 실내 실시간 방재시스템을 위한 기초연구로서 실시간 방재시스템을 위한 기반 구조 및 알고리즘을 제시하는 것을 목적으로 한다.

대피 시뮬레이션 알고리즘에는 Cellular Automata(CA) 모델을 적용하여 화재와 같은 응급상황 발생시, 실내 이동자의 움직임을 나타내는 데 사용하였다. 또한 추후에 실내 센서와 동기화시켜 실시간 방재안내시스템에 적용할 수 있도록 실내 데이터가 지리적으로 참조된 데이터를 이용할 수 있게 하였다. 이를 위하여 본 연구에서는 GIS layer를 입력데이터로 사용할 수 있도록 구축하였다.

본 연구에서는 대피 시뮬레이션에 실좌표계가 정의된 건물의 구조를 표현하고 있는 GIS layer를 적용함으로써 실시간 실

내방재시스템에의 응용가능성을 테스트하여 보고자 하였다. GIS데이터를 기반 데이터로 하여 실 좌표계가 적용되도록 하였으며, CA 모델을 활용하여 대피자의 움직임이 현실에 가깝도록 시뮬레이션 하였다.

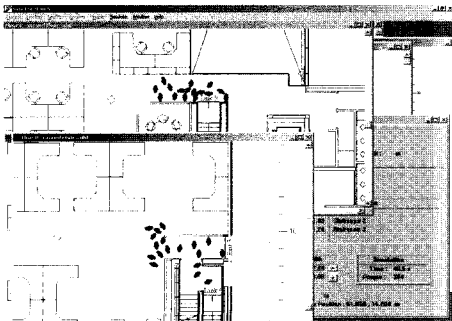
2. 선행연구 및 관련 이론

2.1 선행연구

1) Fire Evacuation 관련 상용시스템

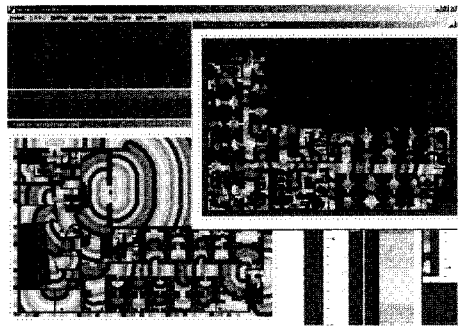
Simulex(2006)는 가장 보편적으로 사용되는 건물 내 대피시뮬레이션 도구로 복층 건물에서의 대피 시뮬레이션을 보여준다. 그 움직임을 대피자 단위로 보여준다.

이는 2차원 CAD DXF 파일을 바탕으로 건물 도면을 생성하고 계단을 이용하여 층과 층 사이를 연결하는 방법으로 건물 모델을 구성한다. 그리고 이를 일정한 크기(0.2m×0.2m)의 cell로 나누고 각 cell 별로 출구까지의 거리를 계산하여 Distance Map ([그림 2])을 구성하고 이를 이동경로로 이용한다.

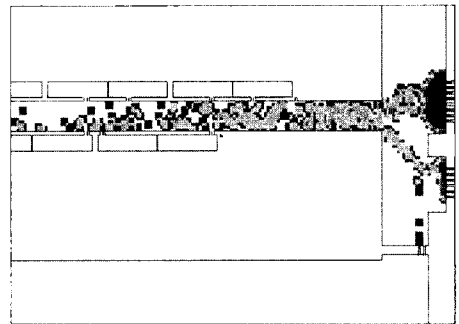


[그림 1] Simulex (<http://www.ies4d.com/>)

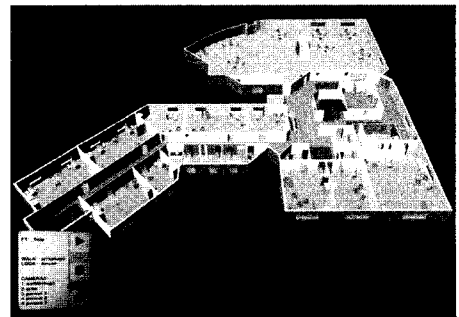
EXODUS([그림 3]) 또한 Simulex와 함께 널리 사용되는 대피시뮬레이션 도구이다. 이는 비상상황과 평상시의 전환에 사용할 수 있도록 되어있고 이 또한 대피자별 움직임을 보여준다. 사람-사람, 사람-환경, 사람-장애물 사이의 상호작용을 고려하였으며 [그림 4]와 같이 VR 애니메이션 기능이



[그림 2] Distance Map(<http://www.ies4d.com/>)



[그림 3] building EXODUS(<http://fseg.gre.ac.uk/exodus/>)



[그림 4] VR EXODUS(<http://fseg.gre.ac.uk/exodus/>)

있다는 특징이 있다. 이 또한 CAD DXF 파일을 바탕으로 건물 모델을 구성한다. 이 소프트웨어는 위의 Simulex와 함께 구조물의 효율적 소개 및 피난 성능을 진단하는 용도로 활용되고 있다.

이외에도 십여 개의 상용소프트웨어가 상용되고 있는데 Gwynne et al.(1999)은 이를 활용 목적, 공간 재현 방법, 모델 적용 대상, 행태 예측 방법에 따라 분류하는 방법을 제시한 바 있다. <표 1>은 그 분류 기준으로 두 시뮬레이터를 비교한 것이다.

<표 1> Simulex와 EXODUS의 비교

		EXODUS	Simulex
목적		시뮬레이션 (Simulation)	시뮬레이션 (Simulation)
재현	형태	cell 기반 (Fine Network)	cell 기반 (Fine Network)
	사람	독립적 (Individual)	독립적 (Individual)
	행태	규칙기반 (Rule Based)	잠재적 행태모델 (Implicit)
기타		2차원적 표현	VR 애니메이션

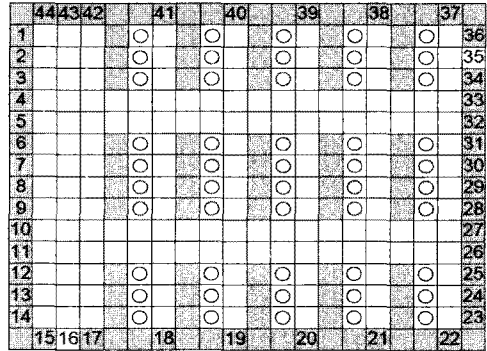
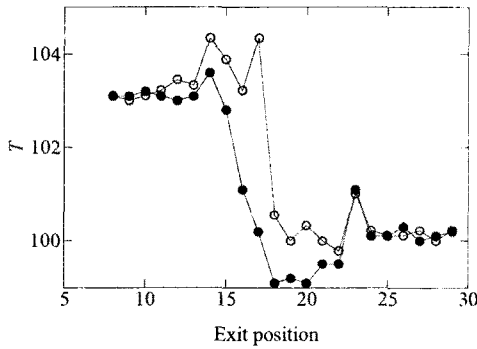
위에서 언급한 바와 같이 현재 상용되고 있는 소프트웨어들은 건물 모델의 구성을 CAD DXF 파일과 같이 토폴로지가 없고 지리참조 되어 있지 않은 도면을 활용한다. 그리고 가상의 데이터와 조건을 적용하기 때문에 이는 실시간방재시스템에의 응용에는 한계가 있다. 본 연구에서는 GIS데이터를 기반데이터로 하는 시뮬레이션 시스템을 구현하여 실시간 실내방재시스템으로의 응용이 가능하도록 하였다.

2) Cellular Automata 모델 기반 시뮬레이션

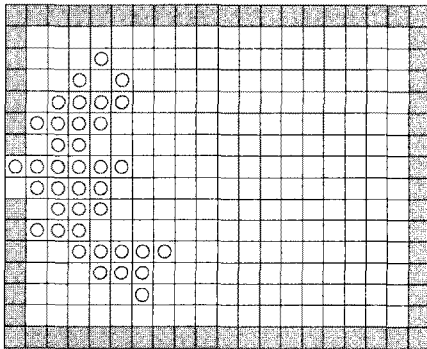
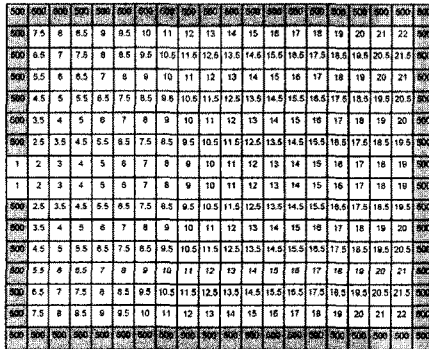
최근 교통의 흐름이나 보행자의 움직임과 같은 복잡한 현상을 물리적인 관점에서 다룬 다양한 연구가 진행되고 있다. 그 중에서도 대피 시뮬레이션을 다룬 연구에서는 molecular dynamics이나, lattice gas 모델, CA 모델을 적용한 사례들이 있다. 특히 CA 모델은 연산과정이 간단하고 이동객체의 개별 움직임을 나타낼 수 있기 때문에 시뮬레이션에 많이 사용된다. CA를 대피시뮬레이션에 활용한 예를 살펴보면 Ioakeim G. Georgoudas 등(2006)은 대피상황시의 군중의 움직임을 추적하는 시스템에 활용하였고, Yuan Weifeng 등(2007)은 다양한 속도를 갖는 대피자의 움직임을 나타내는 데에 CA 모델을 적용하였다. B. Toledo (2007)는 가상의 공간을 설정하고 공간 내의 장애물 유무와 출구의 위치에 따른 대피 시간을 비교하였다([그림 5]). 여기서는 CA 모델을 적용하여 연속적인 시간을 timestep이라는 이산적 단위시간으로 분할하고 공간을 일정한 크기의 격자로 분할한 뒤 간단한 공식으로 각 격자 별 출구로부터 거리 값을 계산하였으며([그림 6]) 장애물의 배치와 출구의 위치를 변경하며 대피시뮬레이션을 하였다. 시뮬레이션의 결과는 대피가 완료되기까지 걸린 timestep의 횟수이며 출구의 위치별로 이를 비교하는 실험을 하였다.

본 연구에서는 B. Toledo의 연구에 쓰인 Cellular Automata 모델 적용방법을 응용하여 이동경로를 산출하는 알고리즘을 개발하였다. 그의 연구에서 사용된 간단한 cell value의 산출방법과 이동규칙에 cell의 크

CA 모델을 이용한 GIS 기반 화재 대피 시뮬레이션



[그림 5] 출구의 위치에 따른 대피시간(B. Toledo, 2007)



[그림 6] cell value와 시뮬레이션(B. Toledo, 2007)

기와 보행속도 등의 상황에 대해 가변적인 파라미터를 적용하여 보다 다양한 상황에 유연한 시뮬레이션을 수행하였다.

2.2 관련 이론

1) Cellular Automata(CA) 모델

Cellular Automata(CA)는 1951년 von Neumann에 의해 소개되었다. 1990년대 초부터 복잡한 시스템에 활용되면서 도시계획 분야에서 도시성장모델(Batty 등(1999), White와 Engelen(2000), Clarke와 Gaydos(1998))이나 보행, 교통 모델(Takashi Nagatani(1994), Dietrich

E. Wolf(1999), 손영태 등(2004), Shi-qiang Dai(2007)) 등의 시뮬레이션에 주로 사용되어 왔다. 최근 10여 년간 Fire Evacuation에서도 이를 이용한 시뮬레이션이 활발하게 연구되어오고 있다.

Cellular Automata는 cell형태의 automaton의 집합이라 할 수 있는데 automaton은 상태, 규칙(automaton 간의 상호작용), neighborhood(인접한 automaton)로 설명될 수 있다(2.1). 우선 이 세 요소들에 대해 간단히 살펴보자면, 상태는 이진수(0 또는 1)로 간단하게 표현된다. 이는 grid의 cell점유 상태를 나타낸다([그림 7]의 (a)). 규칙은 각 automaton의 상태가 변화하거나 다른 cell로의 이동을

결정하는 것을 말한다. 그리고 neighborhood는 인접한 셀의 개수를 나타내는데 이는 인접방향의 종류에 따라 von Neumann과 Moore로 분류할 수 있다([그림 7]의 (b)).

$$(2.1) \quad A \sim (S, T, R)$$

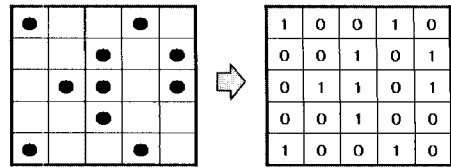
- A*: Automaton
- S*: Automaton의 상태
- T*: transition 규칙
- R*: 인접한 automaton

Automaton의 상태는 시간이 흐르면서 규칙과 인접 cell에 의해 변화된다. CA 모델에서 이러한 automaton의 형태는 1차원 또는 2차원 grid상의 단위 cell이다. 따라서 CA 모델은 일정한 격자모양의 공간에서 automaton의 배열이라고 할 수 있다.

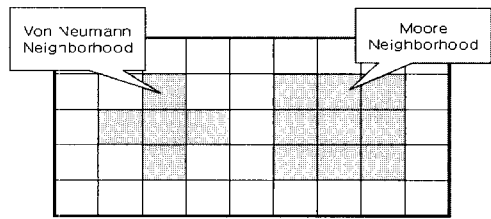
CA에서 각각의 cell은 automaton을 한정 짓는 이산적인 공간이고 각 automaton은 몇 개의 다른 automaton에 인접하게 된다. 이들 간의 거리로 서로의 특성에 영향을 끼치고 영향을 받는다. 각 cell은 입력된 정보를 처리하고 주변 환경으로부터 제공된 데이터와 정해진 규칙을 따라 그 움직임을 진행한다. 이렇게 한번 움직임을 연산하는 시간을 timestep으로, 연속된 시간을 이산적인 시간단위로 분리한다. 이를 수식으로 나타내면 다음과 같다.

$$(2.2) \quad T: (S_t, I_t) \rightarrow S_{t+1}$$

- S_t : time step이 t 일 때 상황
- S_{t+1} : time step이 $t+1$ 일 때 상황
- T : transition 규칙
- I_t : input 정보

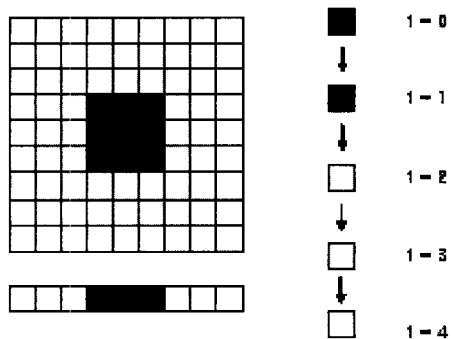


(a) cell의 점유상태



(b) neighborhood

[그림 7] cell value와 시뮬레이션 (Itzhak b., 2004)



[그림 8] CA에서 시간의 흐름에 따른 cell의 변화

[그림 8]는 CA에서 시간(timestep)에 따라 cell의 상태가 달라지는 것을 보여준다.

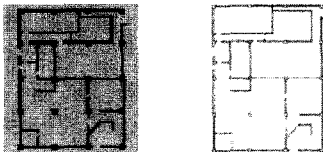
본 연구에서는 이와 같은 Cellular Automata의 특징을 활용하여 연속된 시간과 공간을 timestep과 단위 cell로 나누어 2차원 grid 구조를 기반으로 실내공간에 적용이 가능한 시뮬레이션 모델을 개발하였다.

3. 데이터 구축 및 시스템 구현

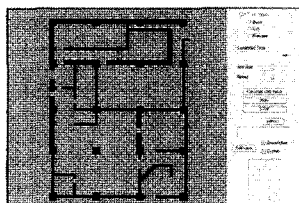
3.1 개요

본 연구에서는 실시간방재시스템에 관한 기초연구로 CA와 GIS 데이터를 적용한 대피시뮬레이션을 수행하였다. 수행한 시뮬레이션에는 공간분석 및 실내 측위기술과의 연계가 가능하도록 하기 위해 GIS layer를 기반데이터로 활용하였고, 대피자 개개의 대피 과정을 나타내기 위해 cell 기반에서 실시하였다. 그리고 간단한 연산으로 빠른 실행이 이루어지도록 하기 위해 Cellular Automata 모델을 적용하였다. [그림 9]는 시스템 구현 과정과 방법을 나타낸다.

① shape file을 불러들이는 컨트롤과 CA 모델을 표현할 컨트롤을 구축한다.



② 같은 인터페이스에 두 컨트롤을 포함시키고 시스템 상의 위치를 일치시킨다.



[그림 9] 시스템 구현 과정

3.2 CA 모델 적용 방안

본 연구에서 실내 공간은 2차원의 grid로 표현된다. CA 모델의 특징과 간단한 연산의 장점을 고려하여 각 cell의 특징과 cell 간의 상호작용에 관한 규칙을 세워 적용하였다.

그 적용방법은 크게 세 가지로 구분할 수 있다. 먼저 cell의 크기, 속도와 같은 파라미터, 시뮬레이션에서 대피자의 움직임의 결정에 사용될 cell 별 distance value, 그리고 마지막으로 이동규칙이다.

Cell의 크기는 한국표준과학연구원(1999)에서 조사한 신체적 공간영역(남자 : 0.15m², 여자 : 0.14m²)을 참고하고 여유 공간을 고려하여 0.16m²(0.4m×0.4m)로 설정하였다. 속도는 도로교통안전공단에서 조사한 보행 속도인 1.56m/s로 설정하였다. 이를 가지고 CA 모형에서의 이산적인 공간과 시간을 표현한다.

Distance value는 GIS layer(shape file)를 입력받아 cell로 변환한 후, 벽체로 인식되어 점유되어진 cell과 출구로 결정되어진 cell을 고려하여 일정한 규칙을 가지고 산출한다. 출구에 해당하는 cell의 값을 기준으로 설정하고 출구에서 멀어질수록 큰 값을 갖도록 한다. distance value의 계산은 이미 계산되어져 있거나 초기에 설정된 cell

<표 2> 사용된 주요 파라미터

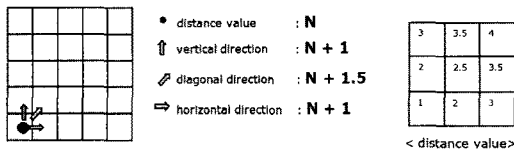
변수	정의
cell 크기	0.4 * 0.4 m ²
timestep	$\Delta t \approx 0.3 \text{ s}$
distance value	$N + \lambda (\lambda > 1)$
보행속도	1.56 m/s

의 값을 기준으로 인접한 cell의 값부터 계산하여 또 그에 인접된 cell의 값을 계산하는 방법으로 하게 된다.

수평 또는 수직방향으로 인접한 cell의 distance value는 기준이 되는 값에 1을 더하고 대각방향으로 인접한 cell의 값은 기준 cell의 값에 대각방향의 거리를 고려한 1.5를 더하는 방식을 택하였다. 이 때, 한 cell에 각각 3개 이상의 인접한 cell을 갖기 때문에 2개 이상의 계산된 값이 발생할 수 있다. 이때는 계산된 값 중 가장 작은 값을 적용하도록 하고 장애물의 경우 매우 큰 값을 적용하여 장애물로 인식되도록 한다.

이렇게 계산이 끝나고 나면 이것을 기준으로 객체가 이동하게 되는데 distance value가 작은 방향으로 이동하게 된다. 이 과정에서 같은 cell로 이동하려고 하는 충돌상황이 발생 할 수 있는데 이 경우에는 random 변수를 사용하여 결정하게 된다. [그림 12]는 객체의 이동 과정과 그 움직임을 계산하는 과정을 간단하게 보여준다.

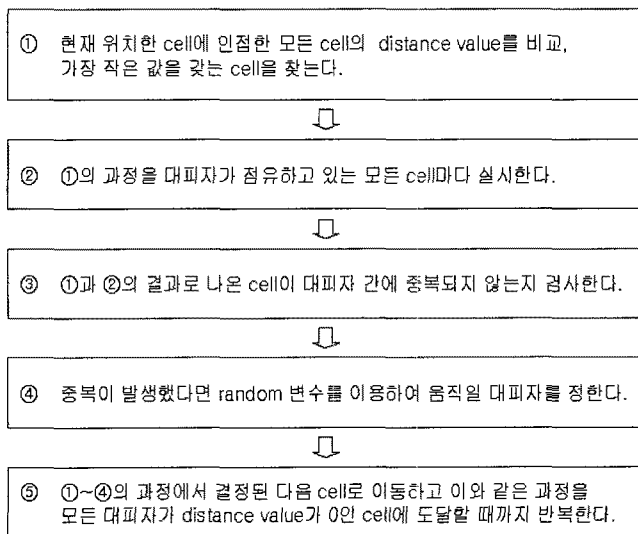
먼저 대피자가 점유하고 있는 모든 cell에서 인접한 cell 중 가장 작은 크기의 distance value를 갖는 cell을 찾는다. 그 cell이 한 번의 timestep이 지난 뒤 이동하려고 하는 cell이 된다. 같은 cell로 동시에 이동하게 되는 오류를 피하기 위해 같은 cell이 있는



[그림 10] distance value 연산과정



[그림 11] 이동규칙



[그림 12] 객체의 이동 과정

지 중복검사를 한다. 만약 같은 cell로 이동하려는 대피자가 있으면 random 변수를 이용하여 한 대피자만 뽑아 이동시키고 나머지 대피자는 움직이지 않고 다음 timestep 이 지난뒤에 이동한다. 이 과정을 모든 대피자가 distance value가 0인 출구에 도달할 때까지 반복한다.

3.3 GIS Layer의 활용

본 연구에서는 가장 보편적으로 활용되고 있는 shape file을 GIS Layer로 활용하였다. 폴리곤(polygon), 폴리라인(polyline) 등 여러 가지 형태의 피쳐(feature)로 이루어진 shape file 중 건물 내부 공간의 벽체를 인식시키기 위해 폴리라인 형식의 파일을 활용하였고 공간의 구조를 인식하기 위해 폴리곤 형식의 파일을 활용하였다.

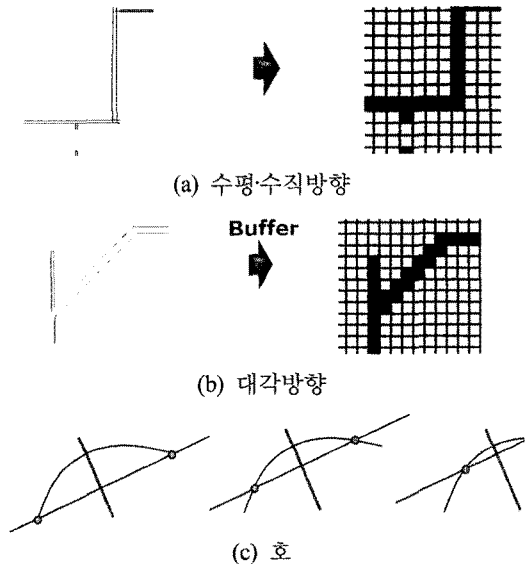
CA 모델을 적용하기 위해 shape file을 grid로 변환하는데 폴리라인 데이터를 사용하여 위에서 언급한 바와 같이 벽체를 grid 위에 나타내어 시뮬레이션에서 장애물로 인식하게 된다. 그 과정은 다음과 같다.

- [1] 폴리라인 데이터를 읽어온다.
- [2] 피쳐를 라인(line)과 호(arc)로 분류한다.
 - a. 라인으로 분류된 피쳐의 형태를 수평·수직방향과 대각방향 으로 분류한다.
- [3] 분류된 피쳐 별로 적절한 방법을 적용하여 grid에 나타낸다.
 - a. 수평·수직방향 : grid상에서 라인의 시·종점에 해당하는 cell을 결정, 그 사이의 cell을 벽체로 인식한다.
 - b. 대각방향 : 위의 [3]a.의 방법과 같다. 단, 연속된 벽체로 표현하기 위

해 해당 라인과 평행한 두 개의 가상라인을 함께 인식한다.

- c. 호 : 시·종점을 잇는 직선의 중점에서 직교하는 선을 생성한다. 추출한 선과 해당 호가 만나는 점으로 향하는 방향으로 시·종점을 잇는 직선을 cell의 크기만큼씩 이동하며 만나는 점을 찾아서 저장한다. 시·종점과 저장한 점을 잇고 이로 인해 만들어진 라인을 [3]a. 또는 [3]b.의 방법을 적용하여 grid에 표현한다.

[그림 13]은 위의 과정을 보여준다. 벽체로 인식된 cell은 CA 모델을 적용하는 과정에서 장애물로 간주된다.



[그림 13] shape file의 grid로 변환

3.4 시스템 구현

본 연구에서는 가장 보편적으로 활용되고 있는 shape file을 GIS Layer로 활용하였

다. 시뮬레이션 테스트를 위한 시스템은 .Net Framework 2.0기반의 환경에서 구현하였고 프로그램 언어는 C#을 사용하였다. 먼저 shape file을 표현하는 뷰어와 cell로 변환된 맵을 표현하는 뷰어를 두고 이 둘이 일치되도록 하였다. GIS layer의 변환은 각 line의 시·종점의 좌표를 이용하였으며, 이 때 line이 위치한 지점의 cell은 장애물로 인식되도록 하였다.

시뮬레이션에서 GIS layer는 정적인(static) 조건이 되고 cell을 표현하는 컨트롤은 동적인(dynamic) 조건이라고 할 수 있다. Shape file로부터 변환된 피쳐들은 시뮬레이션을 수행하는 동안 달라지지 않는 벽이나 장애물로 인식되지만, grid위에서는 이동객체들이 그들 간 서로에게 장애물이 될 수 있고, grid를 구성하고 있는 cell의 속성이 계속 달라지면서 cell간에 서로 영향을 주고받기 때문이다.

CAD 데이터를 기반으로 만들어진 shape file에서 건축도면의 보조선과 같은 건물의 실제 구조와 상관없는 선을 제거하고 편집한다. 이렇게 편집한 shape file을 불러들여 나타낼 컨트롤과 CA 모델이 적용된 컨트롤을 같은 인터페이스에 나타낸다. 이들의 시스템 상의 위치를 연결하여 불러들인 shape file의 라인 피쳐(line feature)가 1절에서 소개한 방법으로 grid에 적용될 수 있도록 한다. 이는 시뮬레이션에서 장애물로 인식된다.

대피자와 출구의 위치를 사용자가 임의로 설정할 수 있도록 하였는데 이는 다양한 시뮬레이션을 통해 결과의 비교가 가능하도록 하기 위함이다. 대피자의 숫자나 위치에 따른 대피 시간을 비교하거나 출

구의 위치를 다양하게 배치하여 시뮬레이션 하여 그 결과를 비교 분석하거나 하는 경우를 포함한다.

시뮬레이션의 결과는 위에서 언급한 바와 같이 대피가 완료되기까지 시뮬레이션에 걸린 시간, 또는 일정 시간까지의 대피 상황, 대피자의 현재 위치 등으로 산출할 수 있다.

4. CA 및 GIS 기반 대피시뮬레이션

4.1 시뮬레이션 개요

3장에서 소개한 방법론을 적용하여 시뮬레이션을 수행하였다. 같은 인터페이스에 GIS layer를 표현하는 컨트롤과 grid를 표현하는 컨트롤을 추가하여 두 컨트롤의 시스템상의 위치를 연결하는 방법을 사용하였다.

먼저 시뮬레이션을 통해서 대피시간과 대피자의 위치를 산출하는 것을 기본으로 하여 여러 가지 시뮬레이션을 수행하였다.

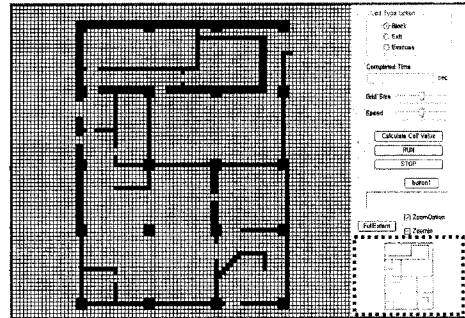
Shape file을 추가하고 출구를 지정한 뒤 각 cell별 distance value를 계산 한다. 그리고 계산된 distance value와 3장 2절에서 소개한 이동규칙을 적용하여 시뮬레이션을 수행한다. 결과로는 대피자의 위치와 대피시간이 산출되는데 필요에 따라 일정 시간까지 시뮬레이션을 실시하거나 현재 대피자의 위치좌표 정보를 산출할 수 있다. 그리고 출구를 수행자 임의로 지정할 수 있기 때문에 출구의 위치나 너비를 다르게 하여 수행할 수도 있다. 다음 절에서는 몇 가지 다른 시뮬레이션을 수행하고 그 결과에 대해 분석하였다.

4.2 시뮬레이션 결과 및 분석

1) 시뮬레이션 결과 분석

◎ GIS Layer(shape file) 적용결과

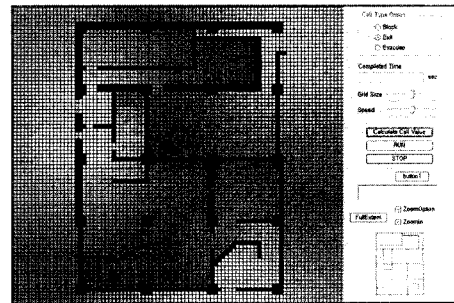
[그림 14]는 시뮬레이션의 초기화면이자 shape file을 적용한 모습을 보여준다. 인터페이스의 우측 하단에 점선 부분은 shape file을 보여주고 왼쪽 grid로 된 컨트롤은 우측 하단의 shape file의 line 피쳐를 표현한 것으로 짙은 색으로 보이는 점유된 cell은 시뮬레이션에서 장애물로 인식된다.



[그림 14] 테스트 시스템 초기화면

◎ Distance value의 계산 및 렌더링 결과

[그림 15]의 출구는 [그림 14]와 같이 장애물이 인식된 뒤 사용자 임의로 설정할 수 있다. [그림 15]는 실제 출구의 위치를 고려하여 산출된 distance map이다. 출구를 설정하여 준 뒤 이를 기반으로 distance value를 계산하고 그 값을 가지고 grid 전체를 렌더링을 실시한 것이다. 출구로부터의 거리가 반영된 것으로 짙은 색을 띄는 곳일수록 출구로부터 거리가 먼 곳임을 의미하고 대피시간이 길다고 할 수 있다. 이렇게 계산된 distance value와 이동규칙을 반영하여 시뮬레이션을 수행한다.

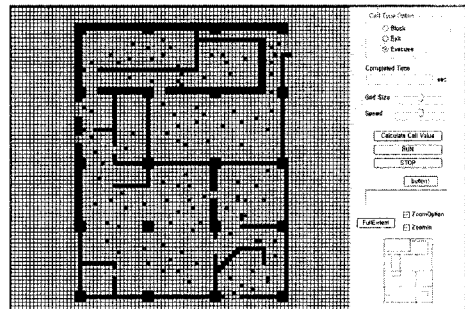


[그림 15] Distance value 렌더링

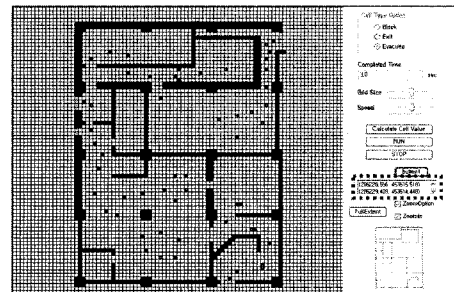
◎ 대피 시뮬레이션

출구의 위치가 정해지고 distance value도 계산되고 나면 [그림 16]와 같이 대피자의 위치를 지정해 준다. [그림 17]는 시뮬레이션 시작 후 3초 경과 후 이동객체의 모습이고 [그림 18]은 10초 경과 후의 모습이다.

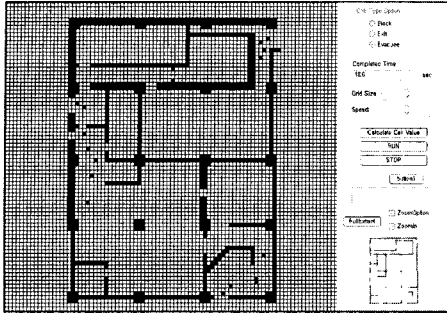
[그림 17]에서 우측 중간에 점선박스로 묶인 부분은 대피자들이 현재 위치한 지



[그림 16] 이동객체의 위치 지정



[그림 17] 3초 경과 후 이동객체의 위치



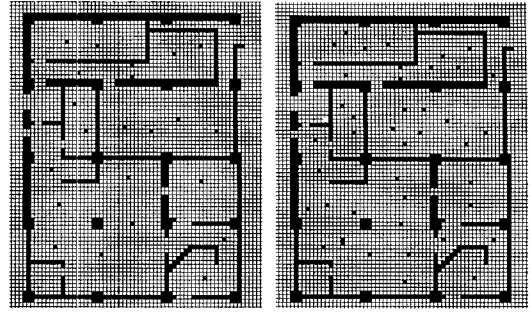
[그림 18] 10초 경과 후 이동객체의 위치

점의 좌표를 보여준다. 이는 shape file이 제대로 적용되었다는 것을 의미하고 공간별 속성과 피쳐(feature)를 제공하는 shape file을 추가하여 연동하면 대피자나 구조자에게 서로의 위치에 대한 다양한 정보를 제공할 수 있게 된다.

위의 시뮬레이션을 확장하여 몇 가지 시뮬레이션을 수행하여 보았다. 먼저 출구의 위치와 너비에 따른 대피시간을 비교해 보았는데 출구의 너비에 따라서는 많은 차이를 보이진 않았으나 같은 개수의 출구를 다르게 배치한 경우에는 비교적 큰 차이를 보였다.

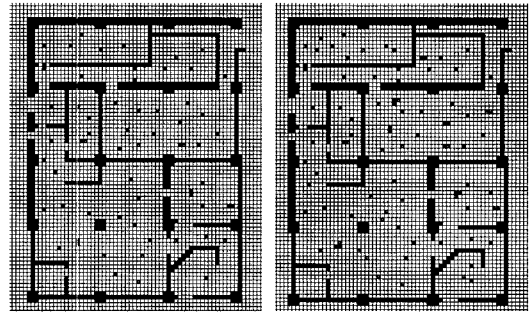
다음은 대피자의 수와 대피자의 위치에 따라 시뮬레이션한 결과를 비교한 것이다.

[그림 19]는 대피자의 수에 따른 시뮬레이션 결과를 보여주는데 대피시간에 큰 차이를 보이지 않고 있다. [그림 20]은 같은 수의 대피자를 고르게 또는 몇몇 구역에 밀집되도록 배치하여 시뮬레이션을 수행하였다. 이 경우에는 대피시간에 큰 차이가 나타났다. 이러한 결과는 <표 3>에서 볼 수 있듯이 대피자가 밀집되어 있을 경우 이동객체 간의 충돌이 고르게 배치하였을 경우보다 빈번히 일어나기 때문이라고 판단된다.



(a) 20명 (10.2s)

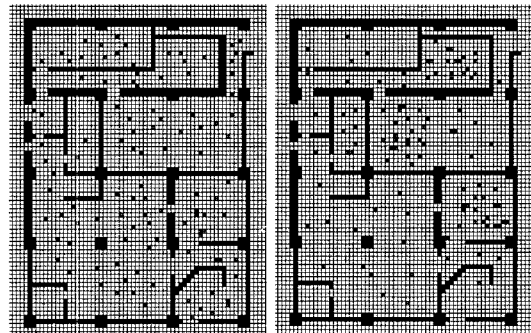
(b) 40명 (10.5s)



(c) 60명 (11.1s)

(d) 80명 (11.5s)

[그림 19] 대피자 수에 따른 비교



[그림 20] 대피자의 분포에 따른 비교

<표 3> 대피시간과 random 함수 발생 빈도수

	(a)	(b)
대피시간	11.9초	15.8초
random 함수 발생 빈도수	17회	62회

2) 응용 시뮬레이션

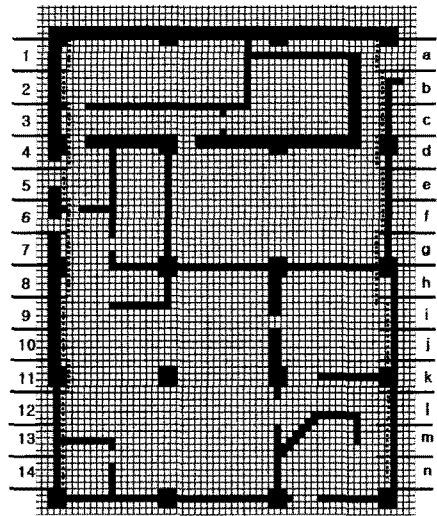
위 시뮬레이션의 응용으로 최적의 출구 위치를 산출하여 보았다. [그림 21]과 같이 직사각형 형태의 공간에서 긴 방향의 벽면에 출구를 양쪽으로 두 군데를 만들려고 할 때 대피자의 수와 위치가 같은 조건에서 대피에 가장 적은 시간이 걸리는 출구를 산출해 보았다. [그림 22]이 그 결과를 보여주는데 그 결과로 산출된 대피시간의 평균은 17.6s로 나타났고 10.1s로 최소값을 나타낸 좌측의 12번과 우측의 e에 해당하는 위치에 출구를 설치하는 것이 비상시 대피의 경우를 생각할 때 가장 적절한 것으로 나타났다.

위와 같은 분석을 실내 공간을 설계하거나 구조를 관리하는 데에 활용하면 방재와 관련하여 효율적인 건물의 설계와 관리에 도움이 될 것으로 판단된다.

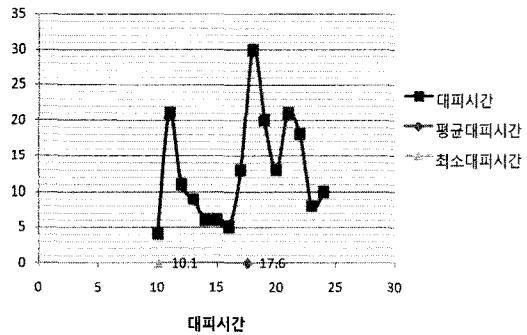
5. 결 론

최근 그 활용도와 중요성이 확대되어가고 있는 실내공간에 대한 서비스 중 실시간 방재시스템에 대한 기초연구로서 실시간 방재시스템을 위한 기반 구조 및 대피 경로 산출방법을 제시하여 보았다.

현재 개발되어있는 대피 시뮬레이션의 경우에는 대부분 지리참조 되지 않은 건축도면을 기반으로 구축되어있고 가상의 데이터로 시뮬레이션을 수행하는 형태로 실내 실시간방재시스템에 응용하는 데에는 한계가 있다. 본 연구에서는 이를 개선하기 위한 방안으로서 평면적인 공간에



[그림 21] 출구의 위치



[그림 22] 출구의 위치에 따른 대피시간

실 좌표계를 적용한 GIS 데이터를 활용한 방재시뮬레이터를 구축하였으며, 다양한 상황을 고려하여 테스트하여 보았다.

시스템의 구현에는 C# 언어를 이용하였고, 기본 알고리즘으로서 Cellular Automata (CA) 모델을 적용하여 화재와 같은 응급 상황 발생시, 실내 이동자의 움직임을 나타낼 수 있도록 하였으며, shape file을 활용하여 지리참조된 데이터를 사용할 수 있도록 하였다. 테스트 결과 약 12m의 대피경로를 빠져나오는데 10초 안팎의 결과

가 나왔고 대피자의 위치를 산출할 수 있었다. 그리고 시뮬레이션의 응용을 통해 대피경로 뿐만 아니라 최적의 출구의 위치를 결정할 수 있다는 결과를 도출할 수 있었다.

이는 대피경로를 계산하여 대피자에게 경로를 제공하거나 대피자의 현재 위치를 제공하여 구난구조에 도움을 주는 등 실시간 방재시스템에 적용할 수 있는 가능성을 보여준다. 향후 본 연구를 보완하여 대피자의 다양한 특성과 대피상황에서 심리적인 요인을 적용하면 현실과 유사한 상황에서의 시뮬레이션이 가능해 질 것이고 센서와의 연계 및 실 데이터를 활용할 경우, 실시간 방재시스템이나 예비 대피경로 분석 등 그 활용도를 높일 수 있을 것이다. 그리고 현재는 2차원 데이터를 활용하였지만 3차원 GIS 데이터를 적용하게 되면 열, 연기의 움직임을 고려한 시뮬레이션이나 실시간위치기반 시스템으로의 응용이 가능할 것이다.

감사의 글

본 연구는 건설교통부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C04)에 의해 수행되었습니다.

참고문헌

박기호·이양원·안재성, 2005, “시간지리학 응용을 위한 시공간데이터베이스 기반의 GIS

컴퓨팅 연구”, 한국GIS학회지, 제13권, 제3호, 2005년 11월, pp.221-237.

박인혜, 2008, “GIS 및 CA 기반 방재 알고리즘 연구”, 서울시립대학교 대학원 학위논문

손영태·박우신·김상구·김태완·김영호, 2004, “CA 기반 보행교통류 Simulation 모형 개발”, 대한토목학회논문집, 제24권, 제4D호, 2004년 7월, pp.563-568.

신승철·조형규·최재필, 2004, “피난 시뮬레이션의 검토 및 공간분석기법의 응용가능성”, 대한건축학회 학술발표대회논문집, pp.611-614.

신운호·박수홍, 2003, “PC 클러스터를 이용한 레스터 GIS 연산의 병렬화”, 한국GIS학회지, 제11권, 제3호, 2003년 11월, pp.213-226.

B.Toledo, 2007, Cellular automata model for evacuation process with obstacles, Physica. A (2007) 10602.

Dietrich E. Wolf, 1999, Cellular automata for traffic simulations, Physica A, 1999, 263: 438-451.

L.G. Chalmet, R.L. Francis and P.B. Saunders, 1982, Network model for building evacuation, Management Science, 28(1): 86-105.

H.J. Miller, Y. Wu and M. Hung, 1999, GIS-based dynamic traffic congestion modeling to support time-critical logistics, In Proceedings of the Hawai'i International Conference on System Science, January 5-8, Maui, Hawaii.

Ioakeim G. Georgoudas, Georgios Ch. Sirakoulis, and Ioannis Th. Andreadis, 2006, A Cellular Automaton Crowd Tracking System for Modelling Evacuation Processes.

Jian-ping Meng, Shi-qiang Dai, Li-yun dong, Jie-fang Zhang, 2007, Cellular automaton model for mixed traffic flow with motorcycles, Physica A.

M. Meijers, S. Zlatanova, N. Pfeifer, 2005, 3D Geo-information indoors: Structure for evacuation, In Proceedings of Next generation 3D city

- models 21-22 June, Bonn, Germany, 6.
- Sven Maerivoet, Bart De Moor, 2005, Cellular automata models of road traffic, Science Direct, Physics Reports, 2005, 419, pp.1-64.
- S.L. Cutter, 2003, GI Science, Disasters, and Emergency Management, Transaction in GIS, 2003, 7(4): 439-445.
- Takashi N., 1994, Dynamical jamming transition induced by a car accident in traffic-flow model of a two-lane roadway, Physica A, 1994, 202: 449-458.
- Yuan W., Tan Kang Hai, 2007, A novel algorithm of simulating multi-velocity evacuation based on cellular automata modeling and tenability condition, Physica A, 2007, 379: 250-262.
- Allan B., 2003, GIS, Environmental modeling and engineering, Taylor & Francis.
- Itzhak Benenson, Paul M. Torrens, 2004, Geosimulation - Automata-based Modeling of Urban Phenomena, JHON WILEY & SONS, LTD.
- Nathalie Waldau, Peter Gattermann, 2005 Pedestrian and Evacuation Dynamic 2005, Springer.
- EXODUS (<http://fseg.gre.ac.uk/exodus/>)
- Simulex (<http://www.ies4d.com/>)
- http://www.sidedirectory.com/Computers/Artificial_Life/Cellular_Automata/Conway's_Game_of_Life.aspx
-
- 접수일 (2008년 6월 18일)
수정일 (2008년 7월 15일)
게재확정일 (2008년 7월 22일)