An Indoor Crowd Simulation Using a 2D-3D Hybrid Data Model

Chulmin Jun and Hyeyoung Kim

Dept. of Geoinformatics, University of Seoul,
Seoul, Korea
{cmjun,mhw3n}@uos.ac.kr

Abstract. Recent LBS-related technologies tend to extend to indoor spaces using localization sensors such as RFID. In order to implement real time evacuation applications, at least two problems must be resolved in advance; first, proper indoor data models and implementation methods that can accommodate evacuees positioning and routing computations should be available, second, evacuation simulations also need to be performed using the same indoor databases for consistent integration. However, none of these have been suggested explicitly as of now. Although some 3D modeling studies have dealt with topological structures, they are mainly focused on outer building volumes and it is difficult to incorporate such theoretical topology into indoor spaces due to complexity and computational limitations. In this study, we suggest an alternative method to build a 3D indoor model with less cost. It is a 2D-3D hybrid data model that combines the 2D topology constructed from CAD floor plans and the 3D visualization functionality. We show the process to build the proposed model in a spatial DBMS and visualize in 2D and 3D. Also, we illustrate a test CA(cellular automata)-based 3D crowd simulation using our model.

Keywords: Crowd simulation, spatial DBMS, 2D-3D hybrid data model, CA.

1 Introduction

Recently, large scale complex buildings and underground spaces are getting increased. Daily lives in such places always tend to accompany dangers of accidents, and actually, we have observed many emergency events such as fire. With this, there is growing recognition that current location-based real-time applications need to extend to indoor spaces.

However, although many researchers have proposed and developed fire simulation models over the last decades, they have mainly focused on the prediction of crowd behaviors under emergency conditions, and have not related them to real-time evacuation. One of the reasons that have delayed such applications may be that developing real-time indoor applications require most recent technologies from different disciplines such as sensor network and databases. Among them, we see two problems as the most important factors that need to be resolved first; one is, we need proper indoor data model. Most data shown in scientific research are simple polygon types drawn either artificially or by CAD floor plans. To be able to use semantic information as

O. Gervasi et al. (Eds.): ICCSA 2009, Part I, LNCS 5592, pp. 397–412, 2009.

shown in the outdoor LBS applications, we need similar topological data model for indoor spaces that can store semantic attributes. The other is, the data for indoor applications should be geo-referenced and stored in databases. Current commercial packages mostly use proprietary file-based data types. Such file-based data may suffice for the purpose of simulation or visualization in a single building. However, in order to store many buildings and communicate with real-time indoor localization sensors, indoor building data should be in databases with real coordinates.

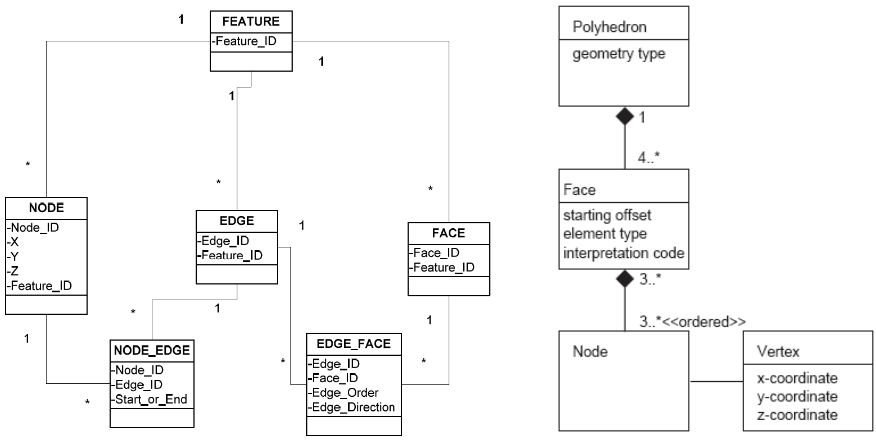
In this paper, we suggest a solution to above two problems. We propose a 3D indoor data model, which is less complex than those in theoretical research while retaining topological properties for semantic-based queries and computations. We show the process to build our data in a spatial database. As a test application that uses our model, we chose an indoor evacuation micro-simulation. For the simulation, we first show a method to partition indoor spaces and plan optimal path routes according to the proximity to exit doors and, then, micro-simulate for the crowd behaviors using CA(cellular automata).

2 Related Works

3D models currently used in the 3D GIS are actually 2.5 dimensional CAD-based data types focusing on visualization purpose in realistic way. They have limitations for analytical purposes in indoor space applications due to its lack of topological and semantic structure. As a solution to this, topological models along with using DBMSs for 3D objects have been recently investigated by some researchers [1, 29, 30, 31]. 3D models suggested by those are generally categorized as follows:

A. SOLID – FACE – EDGE – NODE
B. SOLID – FACE – NODE
C. SOLID – FACE

In solid modeling, boundary representation (B-rep) method is used to represent 3D surface using three primitive types; faces, edges and nodes (or vertices). Faces are bounded portions of a solid surface, edges are used to define the face boundary, and nodes constitute the edges. Type A describes such full topological relations. In type A, actual geometries (i.e. coordinates) are stored in NODE relation while other three relations store non-spatial information. Authors including [5, 6] have carried out research into type A using a relational DBMS (Oracle Spatial) to store 3D objects. However, as shown in Fig. 1-(a), many-to-many relationships such as node-edge and edge-face require additional join tables, which leads to inefficiency for computation due to excessive join queries. Studies of Zlatanova and her colleagues show type B for storing a polyhedron [29, 31] and some other authors use similar approach [1, 4]. In type B, also only the final class, which is NODE, stores the geometries. Without “EDGE” class, we cannot expect such information as “Which edge is this node belonged to?” or “Which edge do these two faces share?”. In a study [30], even more reduced form (Type C) is introduced for storing 3D objects in a DBMS and visualization. While sufficing for the visualization purpose, such “de-normalized” relations as in type B or C suffer duplicated data storage and less capability of topological information retrieval. It can be said that there are always trade-offs between performance and information in 3D modeling.



(a) Type A (b) Type B

Fig. 1. UML class diagrams describing storage of a polyhedron with (a) type A and (b) type B

Table 1. The body and face table in Type C

BODY table	
BID	*FID*
1	1
1	2
1	3
1	4
1	5
1	6

FACE table	
FID	*sdo_ordinate_array*
1 (lower face)	(x4,y4,z4, ,x3,y3,z3, x2,y2,z2, x1,y1,z1, x4,y4,z4)
2 (side 1)	(x3,y3,z3, ,x4,y4,z4, x8,y8,z8, x7,y7,z7, x3,y3,z3)
3 (side 2)	(x4,y4,z4, ,x1,y1,z1, x5,y5,z5, x8,y8,z8, x4,y4,z4)
4 (side 3)	(x1,y1,z1, ,x2,y2,z2, x6,y6,z6, z5,y5,z5, x1,y1,z1)
5 (side 4)	(x3,y3,z3, ,x2,y2,z2, x6,y6,z6, z7,y7,z7, x3,y3,z3)
6 (upper face)	(x5,y5,z5, ,x6,y6,z6, x7,y7,z7, z8,y8,z8, x5,y5,z5)

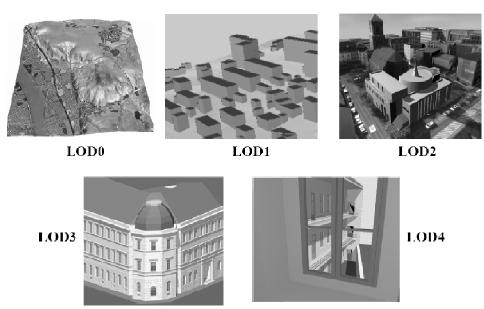


Fig. 2. LoDs in CityGML

The three types investigated in the above are data models for defining 3D volumes not for interior spaces. CAD-based models have been used widely and there is a growing interest in using IFC(Industry Foundation Classes) format especially for modeling and developing building information systems. Although these formats offer flexibility in modeling indoor spaces with various data primitives, they are file-based formats and, thus, have limitations in being used in indoor information systems as mentioned earlier. On the other hand, CityGML which was adopted as a standard by OGC [20], is a 3D model that provides different levels of details ranging from region to interior spaces [19, 27]. CityGML is based on XML format for the storage of data and has capability of storing complex semantic information. However, as of writing this paper, it has not provided fully functional data base implementation. One of the reasons is attributed to the fact that current commercial DBMSs do not fully support topological structure of 3D objects yet.

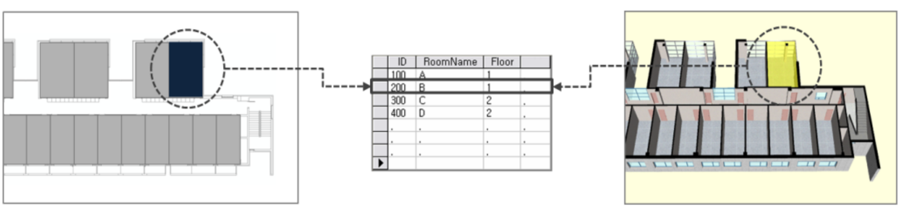
In this study, we propose a simplified approach for building 3D models. While retaining semantic and topological information, our model uses a spatial DBMS, which is necessary in indoor applications such as location-based services. Details are described in the following section.

3 A 2D-3D Hybrid Data Model

3.1 File-Based Approach

In our previous study [21], we had proposed a 2D-3D hybrid data model that can be used both in 2D-based semantic queries and 3D visualization. We used two separate models, 2D GIS layers and 3D models, and combined them using a database table as the linkage method.

First, 2D GIS layers are created from building floor plans. Then, they are converted to vector GIS layers (shapefiles). Then, the IDs of layer's polygons along with other attribute values such as owner's name and status of use are stored in a database table.



(a) 2D layer model (b) DB table (c) 3D model

Fig. 3. Integrating a 2D-GIS floor layer and a 3D model

Polygons in GIS layers are inherently divided separately and contain topological relationships among them, but most 3D models are not constructed such way. Thus, we first had to model a 3D building by creating separated spaces. Not only floors but all rooms in a floor are explicitly divided into individual spaces. Then, the same ID values of spaces as those corresponding spaces in the 2D GIS layer are assigned.

Once both models are constructed following the process described in the above, each space from the two models now shares the identical IDs. Through the shared data table, spatial objects from both sides are synchronized together (Fig. 3). Using this method, we were able to perform analyses and queries using the semantic information stored in 2D layers along with 3D visualization. Fig. 4 shows a test routing simulation under a fire situation. The routing results computed from 2D layer attributes avoiding the fire spot are displayed in 2D and 3D.

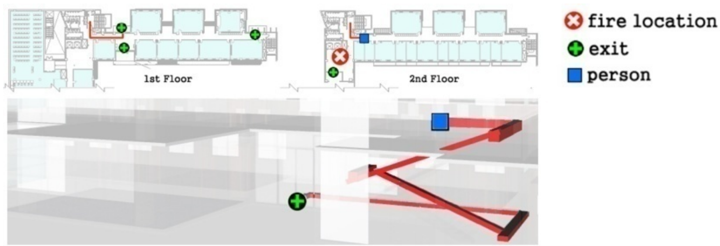


Fig. 4. Evacuation routing simulation under emergency

3.2 DBMS-Based Approach

Although the previous file-based approach was satisfactory in incorporating semantic and topological functionality into a 3D model, it has some drawbacks. First, two models are created separately and need additional table for linkage, which makes consistent maintenance difficult. Second, building a 3D model by separating compartments requires additional time and cost. Finally, such file-based models are not easy to store many buildings and, most importantly, they cannot be integrated with client/server applications such as sensor systems (i.e. RFID, UWB, thermal sensors).

To solve these problems, we proposed in this research a new approach that uses a DBMS instead of files. There are two options to use a DBMS to store spatial data. First, some GIS vendors (e.g., ESRI) store spatial data into DBMS through GIS applications. Since these systems work "under" their GIS applications, we have to rely on these products in order to access and manipulate the spatial data. Other type of DBMSs (e.g, PostgreSQL/PostGIS, Oracle Spatial) offer support of both spatial and non-spatial data types and operations of them in one environment without the help of "helper" GIS software. In order to build a pedestrian application that communicates with a DBMS, we chose the second option as our base approach.

Because semantic information now can be extracted from database tables and used for analyses and 2D/3D visualization, our new model does not require an additional table for linkage. This data model has a multi-layered structure based on 2D building floor plans as the previous file-based model. It retains 2D topology because building floor plans are converted into 2D GIS layers (shapefiles) and then are stored in a spatial database. Thus, it is possible to perform topology-based analyses and operations provided by the DBMS. Also, those records containing geometries can be visualized in 2D and 3D.

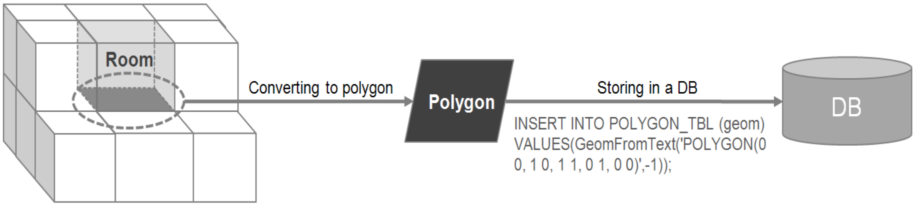


Fig. 5. An example of storing rooms floors in a spatial DB

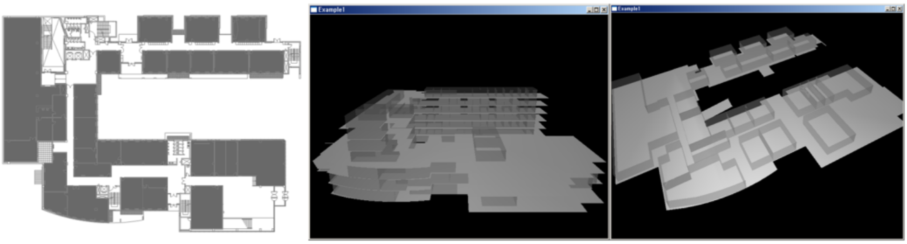
Indoor location-based applications use locations and tracing information of pedestrians who move on the surface floors in the building. This means that it is possible to retrieve semantic data and perform analytical operations only using floor surfaces in such applications (i.e. indoor crowd simulation, indoor wayfinding). This is the reason that we choose to use building floor plans as the base data type. For the connection of floors, we also converted the stairs to a simple set of connected polygons and then stored in the DBMS. Fig. 5 illustrates the process for storing indoor objects in a database. This shows that we used only the bottom part of a room polyhedron. Although current mainstream DBMSs with spatial extensions do not support 3D spatial objects due to the absence of 3D primitives, we made the reconstruction of each object for 3D visualization possible as described in the next section.

This approach can well fit in DBMS-based applications due to less complex and simplified data construction process. Using a DBMS against file format gives many merits including data sharing, management, security, back-up and speed. It is also possible to integrate with sensor systems by storing the sensor information in the database. In this study, we used PostgreSQL/PostGIS for the DBMS. PostgreSQL is an open source object-relational database system, freely downloadable from [23].

3.3 2D/3D Visualization

We developed a test system to display indoor objects in 2D and 3D stored in the database. We used C# and SharpMap [24] for programming the system. SharpMap is a .NET-compatable mapping library that can be integrated with web desktop applications. It is also an open source and supports data sources such as PostGIS, ESRI Shapefile, MSSQL Spatial, ECW and Oracle Spatial [3, 24]. 2D visualization module in our system connects to the PostGIS and make queries to retrieve the necessary floor layers for the 2D display (Fig. 6-a). For 3D visualization part, we used OpenGL library and it also interacts with the PostGIS database for the data retrieval and visualization (Fig. 6-b). The process goes as follows:

a. Queries to 2D floors.
b. Extraction of polygon geometries constituting indoor spaces.
c. 3D visualization by extruding the walls using the stored heights.
d. Repeat above process for the whole building.



a. 2D visualization b. 3D Visualization

Fig. 6. 2D-3D visualization using data from a spatial DBMS

4 Space Partition Using Space Syntax

When an emergency situation occurs in the building that has more than two exits, it is necessary to divide the evacuation routes to avoid biased crowding for exits. This can be done by partitioning the space based on the exit proximity. In this study, we used an accessibility measure introduced in space syntax theory.

Space syntax is a technique that has been used to derive the connectivity of urban or architectural spaces [13, 22]. The theory has primarily been applied in the research areas that seek to find the movement of human beings among architectural spaces or pedestrian paths and it has helped to compute the connectivity of the network of built environment [14]. Readers are advised to refer to the related studies [13, 14, 15, 16, 22] for details. In this study, we will not discuss the space syntax theory and focus on showing the data construction process and the implementation examples.

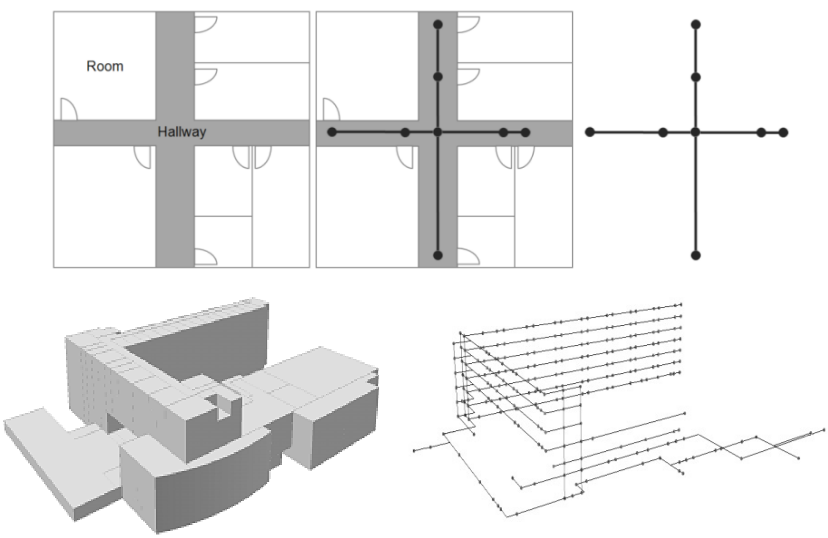


Fig. 7. Constructing indoor networks

To compute spatial accessibility (called *'I'* index in space syntax) and partition the spaces, we first need node-link-based network data. Each room is assigned a node near the door because pedestrian movements begin and end at the room doors or exits. Then Links are built connecting the nodes along the center lines of hallways. Also, floors are connected via stair links. This way, the whole building is mapped to a network structure. Fig. 7 shows how the network is constructed. It shows nodes at the doors and links between them.

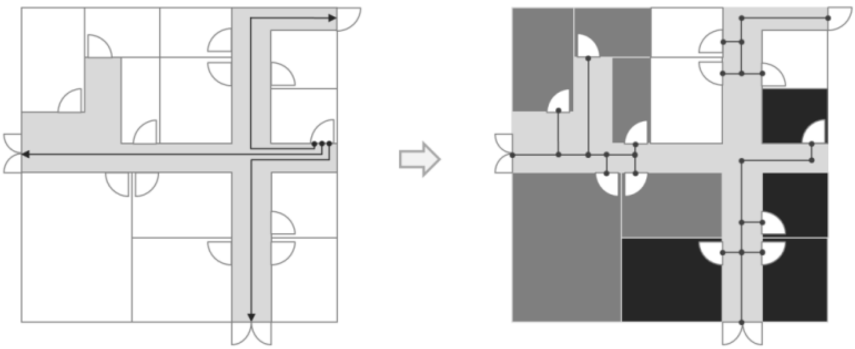


Fig. 8. An example of space partitioning based on space syntax

Fig. 8 shows a simple example of space partitioning based on space syntax. First, we calculate the accessibility for each room to all exits. Space syntax uses visual paths (called '*depth*') as the key cost value of the links. Two network segments with the same length can have different total costs according to how many breaks a path has. Fig. 8 shows that rooms sharing the same exit are grouped together according to similar cost values they have.

We can also consider spatial configuration in collective manner. In large scale buildings such as hospitals, campus, or office buildings, spaces are planned in such way their similar functions are located closely. Thus we measured the 'clusteredness' of spaces that share the similar characteristics. By computing the accessibilities, as for the target rooms repeatedly, we can quantify 'topological' closeness of the related spaces. Fig. 9 shows the department spaces in the test building such as lecture rooms, administrative rooms and professor rooms. We chose two distinct cases for the test; department A and B. The spaces of department A are closely located on the 3rd floor, whereas the rooms of department B are scattered on different floors (1st, 4th and 6th floors). Fig. 10 shows the computed values of accessibility of the spaces of these two departments expressed in 5 gradual color schemes. As a consequence, department A rooms have higher accessibilities than those of department B [16].

This method can be applied to planning optimal evacuation routes. By partitioning the entire floor space into groups of rooms based on the nearest exits, we can plan the normal-time routes. Then, we can perform micro-simulation as described in the following section for the partitioned spaces. The resulting simulation values are stored in a database to be used under real-time emergency conditions for judging if the current population is abnormal against the stored exit capacity and if the alternative routing is necessary.

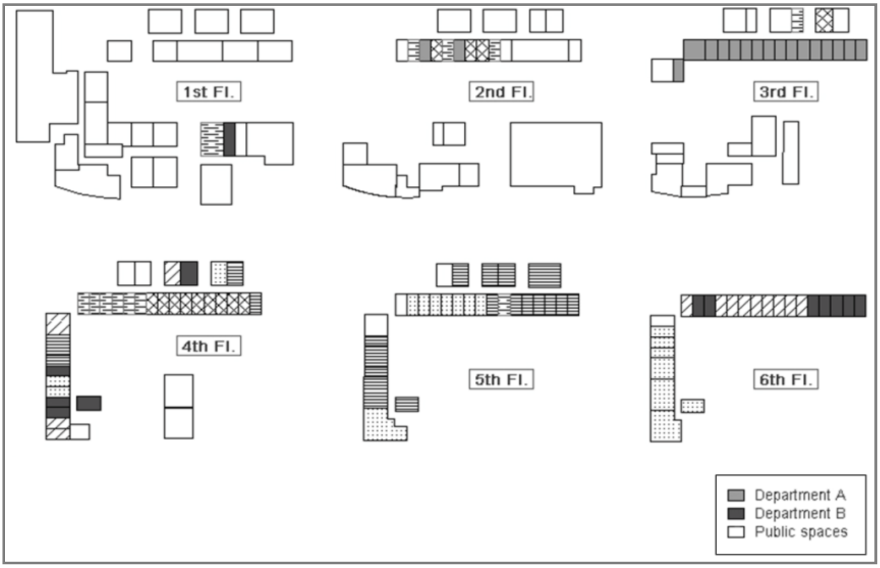


Fig. 9. Classification by spatial property (department)

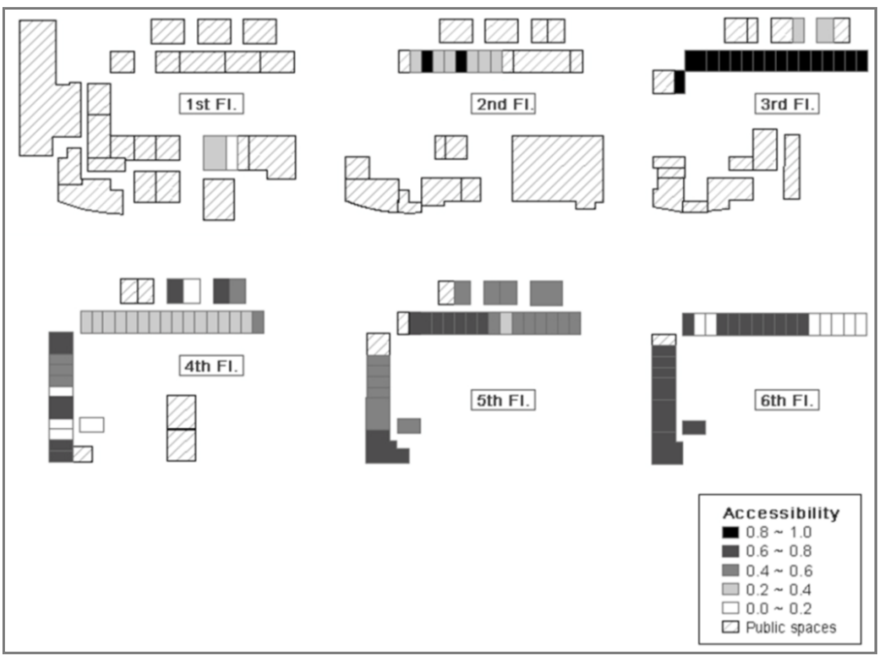


Fig. 10. Accessibility of the classified spaces

5 CA-Based Crowd Simulation

Research approaches used to model evacuation problems come from different disciplines (e.g. transportation network flows, indoor pedestrian behavior simulation) and are generally categorized into two; macroscopic and microscopic models [7]. Macroscopic models appear in network flow or traffic assignment problems and take optimization approach using node-link-based graphs as the data format. They consider pedestrians as a homogeneous group to be assigned to nodes or links for movements and do not take into account the individual interactions during the movement. On the other hand, microscopic models emphasize individual evacuees' movement and their responses to other evacuees and physical environment such as walls and obstacles. Microscopic models are mainly based on simulation and use fine-grained grid cells as the base format for simulation. They have been used by experts in different domains including architectural design for the analytical purposes of the structural implications on the human movement especially in emergency situations.

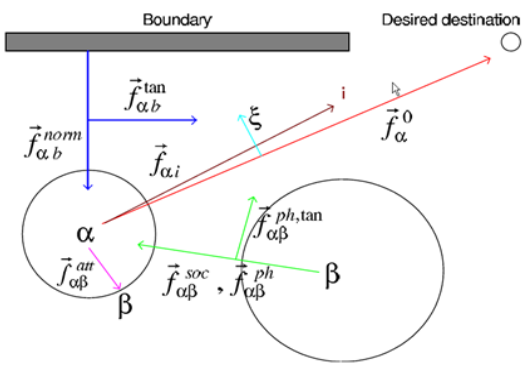


Fig. 11. Helbing's social force model

Different micro-simulation models have been proposed over the last decades [26], but two approaches are getting attention; social force model and floor field model [17]. A frequently cited model of former type is advanced by Helbing and collegues [8, 9, 10] and is based on strong mathematical calculation acted on agents to determine its movement to destination (e.g. exits). Helbing's model considers the effects of each agent upon all other agents and physical environment (Fig. 11) leading to the computation of $O(n^2)$ complexity, which is unfavorable for computer-based simulation with many agents [11, 12].

In recent years there is a growing interest to use cellular automata as the base of micro-simulation [2, 18]. Kirchner and colleagues [17] have proposed CA-based floor field model, where two kinds of fields—static and dynamic—are introduced to translate Helbing's long-ranged interaction of agents into a local interaction. Although this model considers only local interactions, they showed that the resulting global phenomena share properties from the social force model such as lane formation, oscillations at bottlenecks, and fast-is-slower effects. The floor field model uses grid cells as the data structure and computes movement of an agent at each time step choosing the next destination among adjacent cells. This makes computer simulation more effective. The static field *S* represents the shortest distance to an exit door and the dynamic field *D* is a virtual influence area left by a moving agent, which diffuses and gradually vanishes as an agent moves. Each agent can move to adjacent nine cells including itself at each time step $t \rightarrow t+1$ according to probabilities p_{ij}, which is the normalization of the following score.

$$Score(i) = exp(k_d D_i) \times exp(k_s S_i) \times \xi_i \times \eta_i, \quad (1)$$

where

$Score(i)$: the score at cell i
D_i : the value of the dynamic field in cell i
S_i : the value of the static field in cell i
k_d and k_s : scaling parameters governing the degree to which an agent is sensitive to dynamic or static field respectively
ξ_i : 0 for forbidden cells (e.g. walls, obstacles) and 1 otherwise
η_i : 0 if an agent is on the cell, and 1 otherwise.

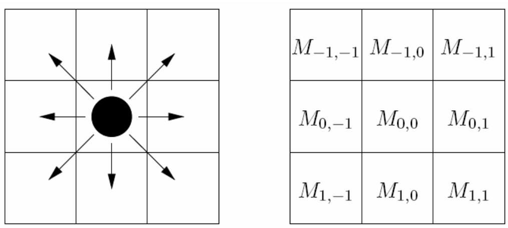


Fig. 12. An agent and its possible transition [25]

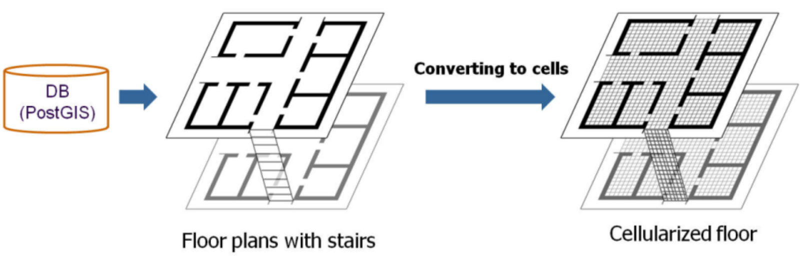


Fig. 13. Reading floor plans from DB and converting to grid cells

We have chosen to base our model on Kirchner's floor field model because CA models are computationally efficient and we have concluded that his model has demonstrated many effects discussed in literature as of now. Also, it leaves the room for us to modify the sensitivity parameters for different situations. The readers are advised to read his literature for more information. Here, we show the process for data construction and illustrates the running examples of our simulation. Instead of using

CAD-based indoor data, we used correctly geo-referenced building data and stored them in a spatial DBMS (PostGIS) considering later integration with indoor location sensors such as RFID or UWB.

As shown in Fig. 13, floor plan data are first read from PostGIS DB, then go through space grouping process based on proximity to exit doors as described in the previous section. Then, they are discretized into cells of size 40cm×40cm considering the human shoulder widths. In our model, we randomly located varying number of pedestrians into the partitioned sections of each floor in order to simulate the crowd egress behaviors in each section to its belonging exits. In real situations, the random data may be replaced with the real pedestrians acquired by location sensors. Fig. 14-(a) shows the process from data retrieval, space partitioning to simulation, and (b) shows the update rules in the simulation. For the static field, we first calculated the distance of each cell from the door it belongs using Dijkstra's shortest path algorithm, and then stored it into the cell. Having decided which cells to move based on the score described in (1), all agents move simultaneously and increment dynamic value. Among any agents competing for a cell, only one is selected randomly for no two agents can occupy one cell. Dynamic field value, D_i is, then, diffused and decayed.

We developed a simulator to test evacuation capacities of floor sections. Fig. 15 shows a 2D and a 3D view of the simulator. In order to implement 3D visualization, we used OpenGL and developed the simulator using C# language. The retrieved geometry information along with stairs was used for rendering in OpenGL and vertical walls were also displayed using the height values stored in the database.

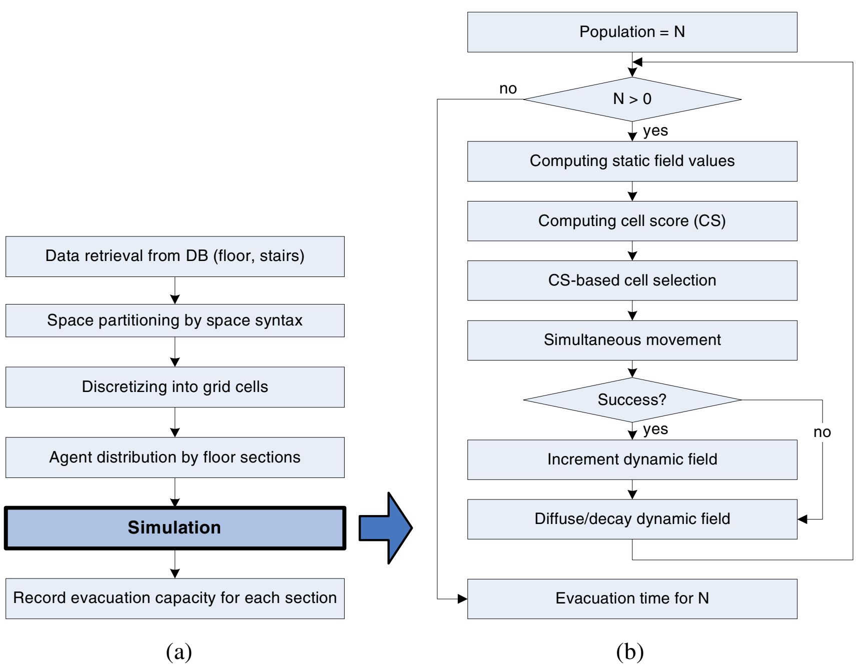


Fig. 14. Processes from data retrieval to simulation

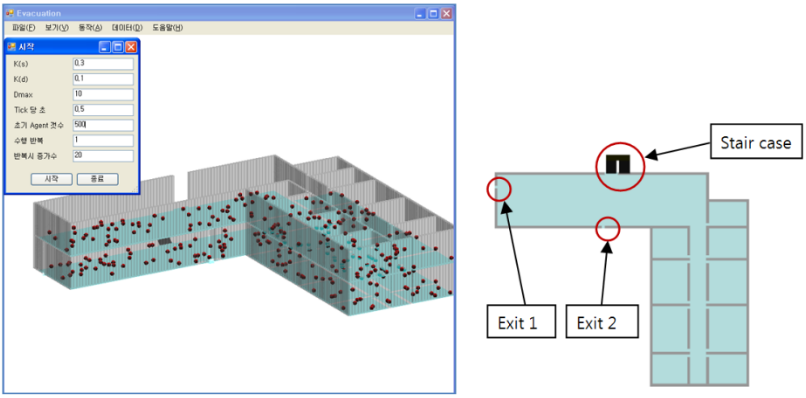
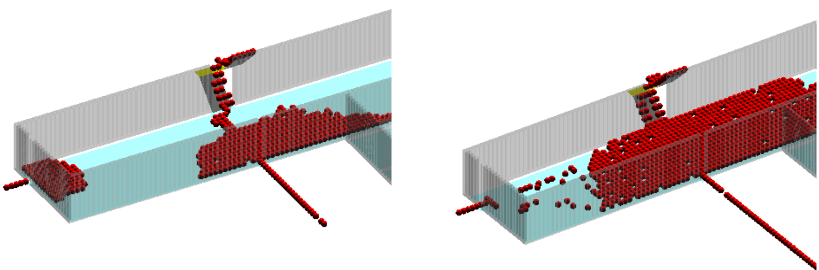


Fig. 15. A snapshot of evacuation simulation with a 3D view

We carried out simulations by varying the parameters k_d and k_s. k_d = 0 causes the agents flow directly towards exits without any herding behaviors while k_s =0 makes them wander around without any clue of direction to exits. Fig. 16 shows that $k_d > 0$ begins to show the herding behaviors following other agents to the second exit. Table 1 shows the effect of k_d on the evacuation time and the use rate of the second exit (Exit 1). 2000 agents were used for the test. We observed that the use rate of the side exit gets increased in proportion to k_d.



(a) $k_d = 0$ (b) $k_d/k_s = 0.5$

Fig. 16. The effect of varying k_d

Table 2. The effect of varying k_d on evacuation time and use of the side exit

	k_d=0	k_d=0.05	k_d=0.1	k_d=0.25	k_d=0.5	k_d=1.0
Exit1	120	351	422	484	566	689
Exit2	1880	1649	1578	1516	1434	1311
evactime	945	723	702	688	670	632

6 Concluding Remarks

In this study, we suggested a process to develop a 3D evacuation simulator instead of trying to improve the scientific investigation of crowd behaviors. In order to be able to integrate our system with real-time evacuation or rescuers' guidance, we suggested the followings:

- A less complex 3D indoor model (actually, multi-layered 2D model) focusing on the semantic information and navigation taking place on the floor surface.
- Implementing the proposed model using a SDBMS and 2D/3D visualization.
- A space partitioning method based on the space syntax theory for the optimized distribution of evacuation routes.
- Developing a 3D crowd simulator using the proposed data model.

Performing evacuation simulation based on the space partitions and SDBMS gives us two benefits:

- It is possible to measure the 'normal' capacities of building exits and analyze the feasibility of architectural design against the pedestrian behavior.
- The simulation results using a varying number of agents can then be stored back into the database to be used to guide the evacuees in real-time situations according to the crowd data captured by indoor location sensors.

We and our colleagues are currently working to relate our model with indoor sensors (localization and temperature sensors) and also to improve the CA model with refinements of parameters.

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No.R01-2008-000-20494-0).

References

1. Arens, C.A.: Maintaining reality: modelling 3D spatial objects in a GeoDBMS using a 3D primitive. M.Sc. Thesis, Delft University of Technology, The Netherlands (2003)
2. Blue, V.J., Adler, J.L.: Using cellular automata microsimulation to model pedestrian movements. In: Ceder, A. (ed.) Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, pp. 235–254 (1999)
3. Boston Geographic Information Systems, `http://www.bostongis.com`
4. Chen, T.K., Abdul-Rahmana, A., Zlatanova, S.: 3D Spatial Operations for geo-DBMS: geometry vs. topology. In: International Archives of XXIth Congress of the ISPRS 2008, Part B2, Beijing (2008)
5. Ellul, C., Haklay, M.: Using a B-rep structure to query 9-intersection topological relationships in 3D GIS – reviewing the approach and improving performance. In: Lee, J., Zlatanova, S. (eds.) 3D Geo-information Sciences, pp. 15–31. Springer, Berlin (2008)
6. Gröger, G., Reuter, M., Plümer, L.: Representation of a 3-D city model in spatial object-relational databases. In: Proc. of the 20th Congress of International Society for Photogrammetry and Remote Sensing, Istanbul, Turkey (2004)

7. Hamacher, H.W., Tjandra, S.A.: Mathematical modelling of evacuation problems- a state of art. In: Schreckenberg, M., Sharma, S. (eds.) Pedestrian and Evacuation Dynamics, pp. 227–266. Springer, Berlin (2001)
8. Helbing, D., Molnár, P.: Self-organization phenomena in pedestrian crowds. In: Schweitzer, F. (ed.) Self-Organisation of Complex Structures: From Individual to Collective Dynamics, Gordon & Beach, London (1997)
9. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407, 487–490 (2000)
10. Helbing, D., Farkas, I., Molnár, P., Vicsek, T.: Simulation of pedestrian crowds in normal and evacuation situations. In: Schreckenberg, M., Sharma, S. (eds.) Pedestrian and Evacuation Dynamics, pp. 21–58. Springer, Berlin (2001)
11. Henein, C.M., White, T.: Agent-based modelling of forces in crowds. In: Davidsson, P., Logan, B., Takadama, K. (eds.) MABS 2004. LNCS, vol. 3415, pp. 173–184. Springer, Heidelberg (2005)
12. Henein, C., White, T.: Macroscopic effects of microscopic forces between agents in crowd models. Physica A 373, 694–712 (2007)
13. Hillier, B.: Space is the Machine. Cambridge University Press, Cambridge (1996)
14. Hillier, B., Hanson, J.: The Social Logic of Space. Cambridge University Press, Cambridge (1984)
15. Jiang, B., Claramunt, C., Batty, M.: Geometric accessibility and geographic information: extending desktop GIS to space syntax, Computers. Environment and Urban Systems 23, 127–146 (1999)
16. Kim, H., Jun, C.: Indoor spatial analysis using Space syntax. In: International Archives of XXIth Congress of the ISPRS 2008, Beijing, WG II/1, pp. 1065–1070 (2008)
17. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionicsinspired cellular automaton model for pedestrian dynamics. Physica A 312, 260–276 (2002)
18. Klupfel, H., Konig, T., Wahle, J., Schreckenberg, M.: Microscopic simulation of evacuation processes on passenger ships. In: Proceedings of Fourth International Conference on Cellular Automata for Research and Industry, Karlsruhe, Germany (October 2002)
19. Kolbe, T.H.: Representing and exchanging 3D city models with CityGML. In: Lee, J., Zlatanova, S. (eds.) 3D Geo-information Sciences, pp. 15–31. Springer, Berlin (2008)
20. OGC (Open Geospatial Consortium), http://www.opengeospatial.org
21. Park, I., Kim, H., Jun, C.: 2D-3D Hybrid Data Modeling for Fire Evacuation Simulation. In: ESRI international User Conference 2007, San Diego (2007), http://gis.esri.com/library/userconf/proc07/papers/papers/pap_1731.pdf
22. Penn, A., Hillier, B., Banister, D., Xu, J.: Configurational modelling of urban movement networks. Environment and Planning B-Planning & Design 25(1), 59–84 (1998)
23. PostgreSQL, http://www.postgresql.org
24. SharpMap, http://www.codeplex.com/SharpMap
25. Schadschneider, A.: Cellular automaton approach to pedestrian dynamics - Theory. In: Schreckenberg, M., Sharma, S. (eds.) Pedestrian and Evacuation Dynamics, pp. 75–86. Springer, Berlin (2001)
26. Schreckenberg, M., Sharma, S.D. (eds.): Pedestrian and Evacuation Dynamics. Springer, Berlin (2001)
27. Stadler, A., Kolbe, T.H.: Spatio-semantic coherence in the integration of 3D city models. In: Proceedings of 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede (2007)

28. Stewart, J.Q., Warntz, W.: Physics of population distribution. Journal of Regional Science 1, 99–123 (1985)
29. Stoter, J.E., van Oosterom, P.J.M.: Incorporating 3D geo-objects into a 2D geo-DBMS. In: ACSM-ASPRS 2002 (2002)
30. Stoter, J.E., Zlatanova, S.: Visualising and editing of 3D objects organised in a DBMS. In: Proceedings EUROSDR Workshop: Rendering and Visualisation, pp. 14–29 (2003)
31. Zlatanova, S.: 3D GIS for urban development, PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Austria, ITC, the Netherlands (2000)