Chapter 18
3D Geo-Network for Agent-based Building Evacuation Simulation

Jinmu Choi and Jiyeong Lee

Abstract. This paper discusses 3D geometric network extraction for building evacuation simulation with an agent-based model. 3D geometric network represents the internal structure of a building, which provides agent-based models with the shortest path for evacuation. 3D geometric network of a building can be built from computer-aided design (CAD) file (vector) and scanned blueprint (raster) through wall extraction and 3D topology construction. We test two wall extraction methods: vector-based medial-axis transformation (MAT) and raster-based thinning. For vector-based MAT, straight MAT is used to extract wall structure from wall polygon generated from CAD data. For raster-based thinning, boundary peeling thinning is used to extract wall structure from scanned blueprint. The extracted 3D geometric network is then used in an agent-based model for building evacuation simulation. In the evacuation simulation, human beings are considered to be the only moving agents. To model human behavior, we adopt a social force model to consider human-to-human and human-to-wall interactions during evacuation. We test simple evacuation scenario in a situation of jam by enforcing different numbers of people in three rooms. The results show that the average velocity increases continuously before jams, decreases during jams at doorways and outer exits, and eventually increases again as individuals escape the jams.

Keywords: 3D Data Models, 3D Geometric Network, 3D Topological Data, Internal Structure Configuration, Straight Medial-Axis Transformation, Boundary Peeling Thinning, Skeleton Extraction, Agent-Based Model, Social Force Model, Human Behavior, Building Evacuation.

18.1 Introduction

As geographic information system (GIS) continues to mature, three-dimensional (3D) modeling has become a tool for GIS analysis with varying levels of success [34]. There is increasing demand for 3D geometric network model in various disciplines including emergency services and cadastre management [2, 6, 16, 33, 34]. If there is
an emergency situation in a building, 3D geometric network can provide fast way to egress the building.

Emergency services rely on both macro- and micro-level GIS [16]. The ultimate emergency management would function on macro scales such as urban areas and on a micro scale such as individual buildings. On the macro-level, an emergency response GIS in a large city might route responders to a building, and then, on the micro-level, route them to the emergency room through the shortest path once they arrive at the building’s entrance [20]. Therefore, emergency response on micro-level features such as buildings depends upon 3D geometric network regarding the internal structure of a building.

Most current commercial GIS, however, do not provide tools to model 3D geometric network representing the internal structure of a building. They only include surface-based 3D representation methods. ESRI 3D Analyst provides tools for surface generation, volume calculation, and viewshed analysis. ESRI ArcScene emphasizes visualization through texture mapping and fly-through. ERDAS Imagine VirtualGIS and Intergraph GeoMedia also provide 3D fly-through tools. While these commercial GIS systems are primarily concerned with 3D visualization, they do not provide any tool for 3D geometric network representing and analyzing the internal structure of buildings [35].

On the other hand, simulation on the emergency situation requires a model that is a simplified representation of reality [21]. A model can be dynamic if the output represents a later point in time and represents time steps in the operation of a dynamic process [10, 22]. Dynamic models are used to assess different scenarios by attempting to project quantifiable impacts into the future [4]. The possible results of simulation can be visualized and help decision-makers avoid disastrous situations. As a dynamic modeling tool, agent-based model has become one of the key computational approaches to simulate collective outcomes of complex geographic phenomena based on individual agents’ states and behaviors. In particular, human behavior with respect to the evacuation situation can be simulated on the 3D geometric network in a building.

The purpose of this paper is to provide a method to build 3D geometric network data and to use the 3D network data to simulate a building evacuation. 3D geometric network data can be produced from computer-aided design (CAD) files and scanned blueprints. Human behavior in the evacuation process is then simulated using agent-based model on the 3D network data.

In Section 18.2, we describe the framework of building evacuation simulation system that utilizes both agent-based model and 3D geometric network in GIS environment. Section 18.3 and 18.4 detail the procedure to extract 3D geometric network from blueprints that are either CAD file and scanned paper designs. In Section 18.3, 3D geometric network data are derived from CAD file using straight medial-axis transformation that extracts line wall structure from polygon wall. In Section 18.4, scanned blue print data are converted to 3D geometric network. In Section 18.5, building evacuation is simulated using agent-based model based on the internal structure of building and 3D geometric network information. Key ideas are summarized in the conclusion.
18.2 Framework of Building Evacuation Simulation System

An integration of agent-based modeling and GIS offers improvements to understand complex spatial phenomena. GIS provides agent-based model with spatiotemporal GIS data to simulate the distributed nature of actions and reactions at the agent (individual) level. Agent-based model provides GIS with the representation of feature dynamics such as temporal changes in spatial patterns. Therefore, in this study, a building evacuation system is designed in GIS environment (Fig. 18.1). For evacuation simulation, we utilize Agent Analyst\(^1\) as agent-based modeling tool and 3D GeoNet for input data generation.

For agent movement, we adopt and implement Helbing et al.’s social force model [13] (see details in section 18.5.2) in Agent Analyst. Agent Analyst takes building structure data (Fig. 18.1d) and runs a social force model that incorporates a mixture of socio-psychological and physical forces of human-to-human and human-to-wall interactions for pedestrian movement. The simulation is iterative and the results in each step are visualized dynamically in ArcMap (Fig. 18.1e). So, we can explore agents’ evacuation process in a building.

Current commercial software for pedestrian evacuation simulation such ASERI\(^2\) and Legion\(^3\) utilizes CAD file for the physical environment of a moving agent in a building. To build shortest paths from CAD file, users should manually model agents’ routes from each room to destination, which is labor intensive process. Instead of

\(^1\) Agent Analyst is an extension of ArcGIS and can download software and materials at http://www.institute.redlands.edu/agentanalyst/AgentAnalyst.html.

\(^2\) Information on ASERI (Advance Simulation of Evacuation of Real Individuals) are available at http://www.ist-net.de

\(^3\) Information on Legion is available at http://www.legion.com
manual modeling, we have developed 3D GeoNet that automatically extracts the internal structure of a building such as rooms, walls, and doors (and exits) and 3D geometric network from building blueprint data. The internal structures of a building from 3D GeoNet provides agent-base model with geometric configuration (Fig. 18.1b) for agents’ egress movement and the 3D geometric network provides with connectivity and adjacency of rooms for agents’ egress routes (Fig. 18.1c). These data are used as input of Agent Analyst through ArcMap.

Two common data used as blueprint of a building (Fig. 18.1a) are CAD files and paper designs that we can utilize in order to extract the internal structure of a building. Following two sections describes how to build 3D geometric network data from both CAD and scanned blueprint data.

18.3 3D Geometric Network Extraction from CAD Data

CAD data are commonly used as digital blueprint of a building [24]. The extraction of 3D geometric network of a building from CAD data requires the conversion of a room to a node and a shared wall to a link based on dual graph theory [18]. The process starts by extracting wall structure of a building since walls in CAD data are stored closed double lines that need to be converted to single lines. From the wall structure, 3D geometric network can be generated through 3D topological structure. Details on individual steps are explained in the following sub sections.

18.3.1 Straight Medial-Axis Transformation

Walls in CAD blueprint of a building are represented as closed double lines. To build 3D network data, the double line walls need to be converted to single lines after the conversion of the closed double lines to polygons using ArcInfo’s Build command. Then, lines are extracted from the wall polygons. There are several approaches to medial-axis transformation (MAT) for extracting a line from a polygon. A skeleton of a polygon can be extracted by the Delaunay triangulation based transformation [25]. One or two edges of the Delaunay triangles are inside the input polygon. Connecting the midpoints of the inside edges of the triangles produces the skeleton of the polygon (Fig. 18.2a). The skeleton approximates a medial-axis of the polygon. This method has been implemented to extract morphological structures of natural features such as the human body [25].

A true medial-axis can be derived from the Voronoi-Diagram based MAT [5, 17]. Lee [17] has developed an O(n log n) algorithm for polygons with concave corners. His algorithm is based on the divide-and-conquer method, which requires four steps. First, the Voronoi edges for each vertex of a polygon are generated. Second, polygon edges are grouped into chains at the concave vertexes. Third, Voronoi edges are generated for each chain and merged with each successive chain. Finally, the medial-axis is created by merging the final two chains and by removing the Voronoi edges incident at the vertexes of the polygon (Fig. 18.2b). In Chin et al.’s algorithm [5], the medial-axis is the connection of all centers of inner circles of a polygon. The inner circles touch the polygon in two or more points. Both MAT methods create second
order lines at the ‘T’ intersections of skeleton lines. It is more difficult to calculate the interaction between walls and humans using a second order line rather than a straight line.

Fig. 18.2. Medial-Axis Transformation Algorithms

The straight MAT is another method to extract a medial-axis [7]. The straight medial-axis is constructed by a shrinking process in which the edges of a polygon move inward with the same speed (Fig. 18.2c). All edges are reduced to three event points eventually: edge, split, and vertex events. An edge event occurs when an edge collapses down to a point (Fig. 18.3a). If neighboring edges of the edge event still have nonzero length, they become adjacent. A split event occurs when a reflex vertex collides with and splits an edge (Fig. 18.3b). A split event divides a component of the shrinking polygon into two smaller components. A vertex event occurs when two or more reflex vertices are collapsed to the same point (Fig. 18.3c). A vertex event can introduce a new reflex vertex into the shrinking polygon, which eventually create another split event. Once the component of the shrinking polygon becomes a triangle, the straight medial-axis is completed by connecting three vertices of the triangle to its center. Connecting these reduced points forms a straightened skeleton. The algorithm is thus applied to construct a roof on a given ground plan [12]. To extract wall structure from wall polygon data in this paper, Eppstein and Erickson’s straight MAT algorithm [7] is used because the algorithm can avoid the second order lines in order to calculate the interaction between walls and humans in a building evacuation model. The straight medial-axis transformation can also avoid the extraction of “Y” shape medial axis from “T” polygon.

Fig. 18.3. Event Points of Straight Medial-Axis Transformation [7]
18.3.2 Topological Data Structure and 3D Geometric Network

Geometric network for 3D internal structure of buildings can be created through 3D topological data extraction. Boundary-based representations (B-Rep) constitute the most popular model to store 3D topological data. The Urban Data Model (UDM) [6] is based on the B-Rep model and uses boundaries: faces, edges, and nodes. UDM allows us to query topological relationships between rooms. However, topological relationships in UDM are implicit, which requires extra processing power for interpretation. Billen and Zlatanova [3] have proposed the Dimensional Model (DM) in which objects are composed of “dimensional elements.” DM allows for topographical relationships between any combination of one, two, and three-dimensional objects. However, DM would also be resource intensive when applied to an entire building.

Since B-Rep models require a large volume of data and considerable power to process, Lee and Kwan [20] have proposed a simple 3D topological data structure called Combinatorial Data Model (CDM). CDM consists of node-relation structure (NRS) and hierarchical network structure (HNS) to represent topological relationships between objects. In NRS, a node represents a room and an edge represents a wall shared by two rooms. HNS, a subset of NRS, stores edges that connect two nodes through a door, an elevator, or a stair. Therefore, NRS represents adjacency information and HNS connectivity. This paper adopts CDM to build 3D topological structure because it allows for fast processing of 3D analysis with the concise 3D topological data structure. Based on the wall structure extracted from CAD data using the straight MAT, the NRS algorithm builds a 3D topological structure, in which a node represents a room and an edge represents a topological relationship (adjacency and connectivity) (Fig. 18.4).

Fig. 18.4. 3D Topological Data Construction from Wall Structure
The algorithm to generate NRS data needs four steps. First, on the wall structure, doorways of individual rooms are easily extracted by extending direction vector of each end node to the nearest end node (Fig. 18. 4a). The doorway data are stored separately for later use to build connectivity. Second, walls and exits are then used to build room polygons using ArcGIS Build operator that split a line segment shared by three rooms (Fig. 18. 4b). Third, nodes for NRS are identified for individual rooms (Fig. 18. 4c). Finally, shared walls are converted to links that represent adjacency and doorways are converted to links that represent connectivity in 3D topological structure (Fig. 18. 4d). All edges represent connectivity is a subset of the edges that represent adjacency (Fig. 18. 5). Once horizontal NRS data is generated for each floor, HNS data links NRS of each floor by vertical connectivity through an elevator or a stairway [20, 30]. Each node (a room) is vertically adjacent to above and below nodes (rooms).

Fig. 18.5. 3D Topological Data based on Combinatorial Data Model [30]

Three dimensional topological data should be converted to 3D geometric network structure in order to model human behavior in building evacuation simulation. Hallways are represented by nodes in 3D topological data but they should be recognized as edges that connect rooms for navigation. The process to build 3D geometric network from 3D topological data requires two steps [19] (Fig. 18. 6). First, nodes that represent hallways are converted to edges. Hallway nodes are removed and hallway polygons are converted to edges using the MAT algorithm. Second, all edges from the other nodes need to be adjusted perpendicularly to the hallway edges in order to maintain shortest connectivity in 3D topological data.
18.4 Conversion from Scanned Blueprint to 3D Geometric Network

The wall structure of a building can be extracted from scanned blueprint using three fundamental algorithms: Voronoi diagram-based thinning, mathematical operator-based thinning, and the boundary peeling method [1] (Fig. 18.7). The Voronoi diagram-based thinning algorithm is the vector-based method, which starts by collecting sample points of a region’s boundary from raster data [29]. The incremental algorithm [11] computes Delaunay triangulations of the sample points. A Voronoi edge is a part of the skeleton if its corresponding dual Delaunay edge lies completely within the region’s boundary (Fig. 18.7a). If a skeleton has parasitic branches, the skeleton can be simplified by retracting the leaf nodes of the skeleton to their parent nodes [9]. Finally, resulting vector skeleton is converted to raster skeleton. While the Voronoi diagram-based thinning algorithm may preserve the morphological shape of complex region, it does not extract straight line even from simple rectangular area without an extreme number of sample points (Fig. 18.7a).
18. 3D Geo-Network for Agent-based Building Evacuation Simulation

The mathematical operator extracts skeleton of a region using two steps: thinning and pruning [28] (Fig. 18.7b). The thinning derives a general skeleton of an area feature and pruning removes parasitic branches of the skeleton [31]. Thinning process requires predefined sequential window filters (3x3 windows) that preserve eight directions from the center of the window. These sequential filters are applied one by one to thin an area feature iteratively until no changes occur. The thinning process produces the parasitic branches mostly at the end part of the skeleton (left image in Fig. 18.7b), which are removed by pruning. The pruning process utilizes some of the eight directional filters of which direction is corresponding to the directions of the parasitic branches [31]. Pruning is also iterative process until no further changes occur and produces the final skeleton of a region (right image in Fig. 18.7b). While the mathematical operator better approximates medial-axis of a region than the Voronoi diagram-based thinning method, it requires interactive selection of directional filters for pruning. Since the result skeleton depends on the selection of pruning filters, the mathematical operator method is not robust.

The boundary peeling algorithm [27] does not need the sequential filters and the pruning process of the mathematical operator method. The boundary peeling algorithm starts by initializing the region to ON and the background to OFF. Every ON pixel is inspected by a 3x3 window, in which the center pixels are erased (set to OFF) if the pixels are not required for preserving connectivity, maintaining end lines, and preventing inward erosion. In the thinning process, ON-valued center pixels are erased (ERASED) if three conditions are satisfied. First, if the connectivity, defined as the number of chains of connected ON pixels in the neighborhood, is equal to 1, the center ON pixel can be erased (Fig. 18.8a). The erasure of the center pixel will not destroy connectivity within any ON chains in the neighborhood. If the
connectivity is 2 or more, the connectivity will be broken. Second, if the maximum length of a chain of the connected ON pixels in the neighborhood is greater than 1, the center ON pixel can be erased (Fig. 18.8b). If the maximum length is 1, the center ON pixel is the end location of the end line that should be maintained. Third, if the maximum length of a chain of the connected OFF pixels in the neighborhood is greater than 1 and less than 7, the center ON pixel can be erased (Fig. 18.8c). If the maximum length is 1, the erasure of the center pixel can intrude further into ON regions. Finally, if all these three conditions are satisfied, the center ON pixel is set to ERASED. The ERASED-value pixels are treated as if they were ON values to prevent uneven erosion on a single iteration. At the end of individual iterations the ERASED pixels are set to OFF pixels. Iterations stop when pixels are no longer erased. The results are the skeleton of input area (see Fig. 18.7c).

Fig. 18.8. Conditions to Erase Center ON-Value Pixel in Boundary Peeling Process

Skeletons from the boundary peeling method in raster domain can approximate the medial axis of a region better than those produced by the Voronoi diagram-based method. Further, the boundary peeling algorithm is robust than the mathematical operator-based method since it does not require pruning that uses interactive directional filters. Therefore, boundary peeling algorithm is implemented in this study and used to extract the skeleton of wall structure in order to build 3D geometric network data. For experiment, vector polygon walls (Fig. 18.9a) are converted to raster data (Fig. 18.9b) and used as input for the boundary peeling process. The output of the boundary peeling process is the skeleton of wall structure (Fig. 18.9c).

Fig. 18.9. Wall Skeleton Extraction using Boundary Peeling Algorithm

Although skeletons may not be medial-axis, they are enough to represent walls to identify rooms in a building. The result wall skeleton needs to be converted to vector line segments in order to build 3D geometric network through 3D topological
structure. The process to build 3D geometric network from wall skeleton is already explained in the previous section (see Section 18.3.2). In the next section, 3D geometric network information is used to generate input data for building evacuation simulation. The moving agents (human) in the evacuation simulation utilize the shortest path information from 3D geometric network to find the nearest outer exit from their respective locations.

18.5 Building Evacuation: Agent-Based Simulation

The key element of an agent-based model is agent. Agents are autonomous individuals who can make independent decision and influence on a simulation [8, 23, 32]. Agents have their own data and behaviors. Agent data are current states that include the agent’s internal state and relationship with other agents. Agent behaviors sense their surrounding to solve complex problems, to communicate, to move, and to adapt their altered state [4]. The other elements of an agent-based model are relationships and environments. A relationship between agents is specified by linking agents within a system. Environments define the space in which agents interact with the environment and other agents.

18.5.1 Internal Structure of a Building and Agent’s Data

Building evacuation simulation requires four input data: humans, rooms, walls, and doors (or exits). 3D geometric network is used to store connectivity in doors data and adjacency in rooms. Among these input data, humans and rooms are agents. Humans are moving agents that navigate to exit the building. Humans are located randomly in several rooms. Human agents have five data: LOCATION, SIZE, ROOMID, SITUATION, and SPEED. LOCATION stores current location of an agent for his movement. SIZE stores each agent’s physical size for calculating the interaction with other agents. ROOMID stores agent’s current room. SITUATION stores the awareness of an emergency situation in a building. SPEED is agent’s desired speed. The direction of agent’s movement is decided by the agent’s location and the interaction with obstacles such as other agents and walls.

Rooms are modeled as static agents that update their own states in case a room has an emergency situation such as a fire. Rooms data are generated by ArcGIS Clean operator during 3D NRS construction process (see Fig. 18.4b and Section 18.3.2). Room agents have their own data: ROOMID, STATE, and NEIGHBOR. ROOMID is a room number. STATE stores the situation of the room. NEIGHBOR stores the adjacency information of each room, which is built from 3D geometric network.

Walls and doors are environments that provide humans with locational information to assess exit strategy in a building. Walls are extracted by straight MAT (see Section 18.3.1) and doors are generated by directional vector during 3D NRS construction process (see Fig. 18.4a and Section 18.3.2). Especially, geometric network data are used to build evacuation routes from each room to outer exit based on the connectivity of rooms. Doors data stores the connectivity of two rooms for human agent’s moving direction at each door.
18.5.2 Human Behavior

Behaviors rule an agent’s actions that change agents’ location and update agents’ status. In a building evacuation, the only moving agents are humans. The behaviors of human can be derived from the paradigms of pedestrian movements \[14, 15, 26\]. To simulate the crowd dynamics in a building, Helbing et al. \[13\] provide a social force model based on Newton’s acceleration equation (Equation 1). The first term in Equation 1 represents socio-psychological force. The second and third terms represent human interaction (\(f_{ij}\)) (see Equation 2) and human to wall interaction (\(f_{iw}\)) (see Equation 3) forces.

\[
f = m_i a = \frac{d v_i}{dt} = m_i \left( v_i(t) e_i^0(t) - v_i(t) \right) + \sum_{j \neq i} f_{ij} + \sum_{w} f_{iw} \tag{1}
\]

Here, each pedestrian (i) of weight (\(m_i\)) likes to move with a certain desired speed (\(v^0_i\)) in a certain desired direction (\(e_i^0\)) through time (t), and therefore tends to correspondingly adapt his or her actual velocity (\(v_i\)) with a certain reaction time (\(\tau_i\)).

\[
f_{ij} = A_i \exp[(r_{ij} - d_{ij})/B_i] n_{ij} + kg(r_{ij} - d_{ij}) n_{ij} + kg(r_{ij} - d_{ij}) \Delta v^i \mu_{ij} \tag{2}
\]

In Equation 2, the first term \(f_{ij} = A_i \exp[(r_{ij} - d_{ij})/B_i] n_{ij}\) is a repulsive interaction force of two agents i and j to stay away from each other. \(A_i\) and \(B_i\) are constant that can reproduce the distance kept at normal desired velocities. Higher \(A_i\) produces greater repulsive forces with \(B_i\) range overlap between two agents. \(d_{ij}\) denotes the distance between center of two agents (\(H_i\) and \(H_j\)). \(r_{ij}\) is sum of radii \(r_i\) and \(r_j\) of two agents (\(H_i\) and \(H_j\)). \(n_{ij} = (H_j - H_i)/d_{ij}\) is the normalized vector pointing from agent j to i. If two agents touch each other (\(r_{ij} > d_{ij}\)), body compression \(kg(r_{ij} - d_{ij}) n_{ij}\) and relative tangential motion \(kg(r_{ij} - d_{ij}) \Delta v_i \mu_{ij}\) are considered. If they are apart from each other (\(r_{ij} < d_{ij}\)), \(g(x)\) is zero. \(k\) and \(\kappa\) determine the obstruction effect in cases of physical interactions.

\[
f_{iw} = A_i \exp[(r_{iw} - d_{iw})/B_i] n_{iw} + kg(r_{iw} - d_{iw}) n_{iw} + kg(r_{iw} - d_{iw}) (v_{iw} t_{iw}) \mu_{iw} \tag{3}
\]

Equation 3 captures the interaction between humans and walls. \(d_{iw}\) means the distance to wall \(W\). \(n_{iw}\) denotes the direction perpendicular to \(W\). \(t_{iw}\) is the direction tangential to \(W\). Other terms are identical to those in Equation 2.

To implement Equation 1 for human movement in building evacuation simulation, six behaviors of human agents are used: MYSTATE, MYROOM, MYEXIT, MAINFORCE, P2PFORCE, P2WFORCE, MOVE, and UPDATE. The MYSTATE action checks an agent’s states. MYROOM checks current room situation with room ID, which decides the agent’s desired speed (\(v^0_i\)). MYEXIT checks current doorway
to decide the desired direction ($e_0^i$) of the agent. MAINFORCE calculates the socio-psychological force (first term) in Equation 1. P2PFORCE calculates the human interaction force using Equation 2. P2WFORCE calculates the human to wall interaction force using Equation 3. MOVE changes the agent’s location based on the result velocity of Equation 1 that is sum of MAINFORCE, P2PFORCE and P2WFORCE. Finally, UPDATE updates the agent’s location and states.

18.5.3 Building Evacuation Simulation using Agent-Based Model

Among various agent-based modeling tools such as Swarm, MASON, Repast, and so on, Agent Analyst that is based on Repast\(^4\) is used in this study since it is tightly coupled with ArcGIS for data management and visualization. There are two types of agents (vector and generic agents) in the Agent Analyst. A generic agent is a non-spatial agent such as emergency announcements. A vector agent is a spatial agent that is stored as features in a shapefile. In our evacuation simulation, all input data including humans, rooms, walls, and doorways are modeled as vector agents.

In building evacuation model, the only moving agent is human and therefore the simulation is based on the human behavior model (see Equation 1) and the characteristics of the crowd dynamics. We take first five characteristics from Helbing et al.’s \cite{13} nine characteristics of crowd dynamics in order to simplify the situation: (1) people move or try to move considerably faster than normal; (2) individuals start pushing each other, and interactions among people become physical in nature; (3) moving and, in particular, passing through a bottleneck becomes uncoordinated; (4) at exits, arching and clogging are observed; (5) jams build up. We assume that all people in the building know its structure so that they know where the nearest exit is and how to get to it. Door data stores room connectivity from 3D geometric network, which provides the shortest path information.

To run the social force model (see Equation 1), we have specified the parameters as follows: a mass of $m$ is average 60kg. Initial desired speed ($v_0^i$) is 5m/s; initial desired direction ($e_0^i$) is toward exit; distributed pedestrian diameters $2r_i$ in the interval [0.5m, 0.8m] considering shoulder widths; the constant $A_i$ is 5000 and $B_i$ is 0.08m; the parameter $k$ is 750kg/s\(^2\) and $\kappa$ is 3000kg/ms. In the simulation, we consider 15 people ($N$) in three rooms to generate evacuation with and without a jam situation for each room.

The simulation shows the evacuation process through time (Fig. 18.10). People follow the shortest path to the outer exit through the door of their room. Figs. 10a to 18.10f show the evacuation situation in each room in the building. The number of moving people and the average speed are shown in Fig. 18.10g. When people start to jam in a room (Fig. 18.10b and Fig. 18.10c), the number of movements and the average speeds start to decrease (Fig. 18.10g). As people who are nearest to the door exit the room, the movements and the average velocity increase again. However, at the outer exit of the hallway, there is another jam because people out of different rooms meet at the exit, which decrease the average speed again (Fig. 18.10d and Fig. 18.10g).

\(^4\) Repast is an agent-based modeling tool and can download source codes and materials at repast.sourceforge.net.
Therefore, if there are many people in a building and they want to evacuate at the same time, jams can occur at several places such as doorways of each room, stairways, and outer exits of the building. The jams make more distant people take even longer to evacuate and may result in many casualties in an emergency situation such as a fire.

![Building Evacuation Simulation using Agent-based Model](image)

**Fig. 18.10.** Building Evacuation Simulation using Agent-based Model

In this study, we simulate egress movement simply in one story building in 2D display environment. We can easily extend it to multi-story building by the connection of multiple floors in 2D environment or may be directly visualized in 3D environment such as VRML. For evacuation modeling in multi-story building, we need to consider two aspects: vertical connection and agents’ speed in vertical movement. For vertical connection of floors, 3D geometric network already identifies stairwells, elevators, and escalators and stores vertical connectivity information. For vertical egress movement, the agents’ desired speed should be adjusted based on the type of space for vertical movement. In case of emergency, only stairwells may be considered for vertical movement and the desired speed should be reduced since agents could not move freely in stairwells with the same speed as in hallways.

**18.6 Conclusion**

An integration of agent-based modeling and GIS offers improvements to understand complex spatial phenomena. With agent-based models, GIS can represent feature dynamics such as temporal changes in spatial patterns for further time-series spatial analysis. However, current GIS are still nascent in the representation of 3D internal structure of buildings for micro-level spatial modeling such as building evacuation simulation.

In this study, we have discussed building evacuation simulation in GIS environment by integrating 3D geometric network construction procedure (3D GeoNet) with agent-based model (Agent Analyst). These tools are tightly coupled
with ArcMap that is used for visualization. 3D GeoNet generates the internal structure data of a building and 3D geometric network data and provides input data for Agent Analyst. Agent Analyst produces sequential results of agents’ egress movement. The sequential results of simulation are visualized in ArcMap.

Especially, the automated generation of 3D geometric network data is important for building evacuation in order to provide pedestrians’ routing information with room connectivity. In this study, 3D geometric network has been created from building blueprint data through wall extraction and 3D topology construction. We tested two wall extraction methods: vector-based MAT and raster-based thinning. For vector-based MAT, we utilized straight MAT to extract wall structure from wall polygon from CAD data. For raster-based thinning, we tested boundary peeling algorithm to extract wall structure from scanned blueprint. With wall structure extracted using both extraction methods, doorways and rooms were identified. By 3D duality, rooms were converted to nodes and the walls shared by rooms were converted to links in 3D topological structure. 3D geometric network was built by converting nodes in 3D topological structure to links if the nodes represent hallways. In particular, the topological connectivity and adjacency information in 3D geometric network were stored doorway and room data, respectively, which were used in building evacuation simulation.

With spatial data including rooms, walls, doors, and humans, a building evacuation was simulated in GIS environment using Agent Analyst. Especially, human agents’ behavior in building evacuation was modeled by adopting Helbing et al.’s social force model [13]. We tested simple evacuation with a jam situation by enforcing different numbers of people in three rooms. As expected, the results showed that the average velocity increased continuously before jams, decreased during jams at doorways and outer exits, and eventually increased again as individuals escaped from jams.

As we emphasized in the paper, 3D geometric network provides detailed sub-unit structure of a feature such as a building for observing and understanding micro-level navigation. Especially, the integration of 3D geometric network and an agent-based model provides current GIS with new methodology that allows a simulation for dynamic spatial process in a micro-level environment such as a building. Future building evacuation simulation will involve enhanced human behavior model and more complex building structure under disastrous situation such as fires.

### 18.7 Acknowledgments

This research was supported by a grant from 'Seoul R&BD Program (10592)' funded by the City of Seoul, South Korea

### References