A COMPUTATIONAL FRAMEWORK TO SIMULATE HUMAN AND SOCIAL BEHAVIORS FOR EGRESS ANALYSIS

Xiaoshan Pan[1], Chuck Han[2], Kincho H. Law[3], and Jean-Claude Latombe[4]

ABSTRACT

This paper describes a multi-agent based computational framework to simulate human and social behaviors for egress analysis. In order to simulate individual behaviors and social interactions, a perception-interpretation-action based autonomous agent model is proposed. A prototype system, MASSEgress, is implemented and has the ability to simulate some typical social behaviors commonly observed in emergency evacuations. To handle the non-deterministic nature of the problem, statistical methods, such as K-Means clustering analysis, are employed to synthesize simulation results and to deduce evacuation routes and patterns

KEY WORDS

human and social behavior, egress simulation, emergency evacuation, multi-agent system

INTRODUCTION

Among the regulations governing a facility design, safe egress is one of the key issues identified by facility planners, managers and inspectors. Although building codes contain "means of egress" provisions, adhering to the codes is often insufficient to ensure safe evacuations in case of emergencies. Each building is unique, and local geometries – shapes and sizes of spaces and obstacles – can pose significant challenges in designing effective egress systems. Human behavioral factors in emergency situations are often overlooked. For example, in most designs that follow the "Means of Egress" section of the International Building Code (ICBO, 2000), the methods for calculating the number of exits, width of exits, and the distribution of exits are based on an assumption that occupants would evenly utilize all available exits and channels to escape in case of an emergency. Such assumption unfortunately is hardly true in reality – for example, people tend to exit a building following the exits that they are familiar with, and ignore other exits. As a result, egress designs (e.g., the number of exits and their locations, exit width, corridors, and stairs) often fail to meet their functional expectations when a real emergency occurs (Shields and Proulx, 2000).

Human behaviors in emergency situations are complex phenomena, involving a broad range of physical, environmental, psychological and social factors, many of which, have not

[1] Graduate Student, Department of Civil and Environmental Engineering, Stanford University, CA 94305, Phone +1 650/723-1886, FAX +1 650/723-4806, xpan@stanford.edu

[2] Consulting Assistant Professor, Department of Civil and Environmental Engineering, Stanford University, CA 94305, chuck.han@stanford.edu

[3] Professor, Department of Civil and Environmental Engineering, Stanford University, CA 94305, law@stanford.edu

[4] Professor, Department of Computer Science, Stanford. University, CA 94305, Latombe@cs.stanford.edu

been well understood. This type of studies can be quite complicated since it often requires exposing real people to the actual, possibly dangerous, environment. As a result, generalizations about human behaviors in emergency situations for predictive purposes are difficult to draw. To date, not a coherent and comprehensive theory about human behaviors in emergency situations has emerged (Chertkoff and Kushigian 1999; Proulx, 2001). A computational tool which can take into consideration of a variety of human and social behaviors could be valuable towards gaining insights to the egress problem.

In this paper, we describe a Multi-Agent Simulation System for Egress analysis (MASSEgress). The purpose is to yield better understandings of human factors in emergency situation and to assist safe egress design. Although many computational tools for the simulation and design of emergency evacuation and egress are now available, due to the scarcity of human and social behavioral data, many of these computational tools rely on assumptions that have been found inconsistent or unrealistic. As pointed out by Santos and Aguirre (2004), current models have largely ignored insights regarding human and social behaviors from the fields of social psychology and social organization. Our computational framework adopts a multi-agent based simulation paradigm, which is well suited to incorporate and simulate evacuees' cognitive processes and behaviors and to explore emergent macro phenomena such as social or collective behaviors (which usually are not reducible to or understandable in terms of the micro properties of an individual).

RELATED WORKS

Most of current computational tools for the simulation and design of exits can be categorized into fluid or particle systems, matrix-based systems, and emergent systems. Fluid or particle systems couple fluid dynamic and "self-driven" particle models with discrete virtual reality simulation techniques to simulate how large groups of people moving out of buildings. An example is the panic model built by Helbing et al. (2000). A general critique on fluid or particle systems is that the fluid or particle analogies of people movement contradict with some observed human evacuation behaviors, since people do not follow the laws of physics (Still, 2000). The basic idea of a matrix-based system is to discretize a floor area into cells, and cells are used to represent free floor areas, obstacles or people. People transit from cell to cell based on occupancy rules defined for the cells. Two well known examples of the matrix-based systems are Egress (AEA Technology, 2002) and Pedroute (Halcrow Group Ltd., 2003), which have been applied to simulate evacuation in buildings and train stations. It was suggested that the existing matrix-based models suffer from the difficulties of simulating crowd cross flow and concourses, and many assumptions employed in these models are questionable when compared with field observations (Still, 2000). The concept of emergent systems is that the interactions among simple parts can simulate complex phenomena (Epstein and Axtell, 1996; Johnson, 2001). One example of the emergent systems is the Legion system (Still, 2000; Legion International Ltd., 2004). However, Legion was designed as an investigation tool for studying large scale interactive systems. Therefore, the behavioral representation of individual humans is oversimplified in the system. Nevertheless, the emergent concept is intriguing since it has the notion that crowd behavior is a collection of individuals'.

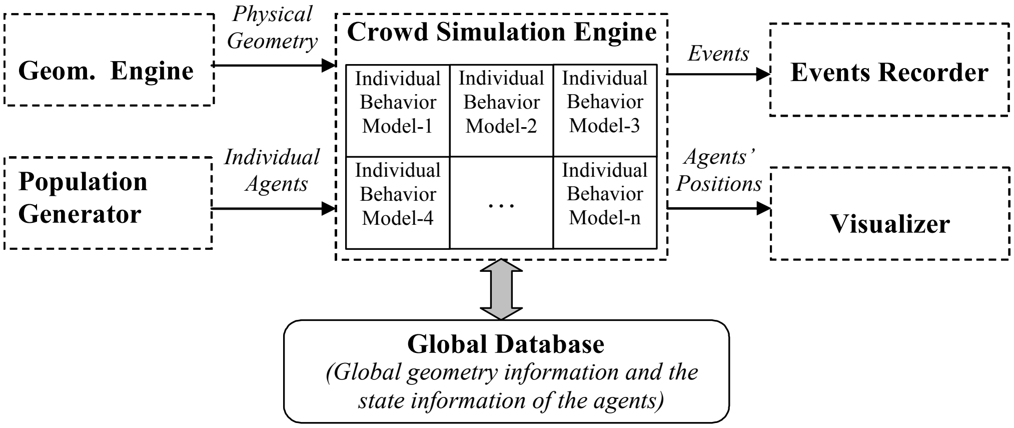

Geom. Engine
Physical Geometry
Crowd Simulation Engine
Individual Behavior Model-1
Individual Behavior Model-2
Individual Behavior Model-3
Individual Behavior Model-4
. . .
Individual Behavior Model-n
Population Generator
Individual Agents
Events
Events Recorder
Agents' Positions
Visualizer
Global Database
(Global geometry information and the state information of the agents)

Figure 1: System Architecture.

In summary, as noted by the Society of Fire Protection Engineers (2002), "(computational) models are attractive because they seem to more accurately simulate evacuations. However, due to the scarcity of behavioral data, they tend to rely heavily on assumptions and it is not possible to gauge with confidence their predictive accuracy (p. 52)." There have been increasing interests in studying human factors in emergency (Bryan, 1997; Shields and Proulx, 2000; Proulx and Richardson, 2002); however, the fundamental understanding of the sociological and psychological components of pedestrian and evacuation behaviors is left wanting (Galea, 2003; Santos and Aguirre, 2004). More detailed reviews of human and social behavior models relevant to egress simulation have been discussed by Pan et al. (2006a, 2006b).

SYSTEM ARCHITECTURE

The system architecture of MASSEgress is schematically depicted as shown in Figure 1. The system consists of six basic modules: a Geometric Engine, a Population Generator, a Global Database, a Crowd Simulation Engine, an Events Recorder, and a Visualization Environment.

- The purpose of the *Geometric Engine* is to produce the geometries representing the physical environments (e.g., a building or a train station, etc.). Spatial information, including obstacles, exits, spaces, spatial layouts, exit signs, etc., is most conveniently defined using CAD tools such as AutoCAD/ADT (Architectural Desktop Software).
- The *Population Generator* generates virtual agents to represent a crowd based on the distribution of age, mobility, physical size, type of facility (hospital, office building, train station, stadium, etc.) and other human factors. The population, its composition, and occupants' behavior would be different for different facility types. This module allows the user to easily generate occupants and specify space assignments.
- The *Global Database* is to maintain all the information about the physical environment and the agents during the simulation. It maintains the state information (mental tension, behavior level, location) of the individuals. The database is also used to support the interactions and reactions among the individuals.

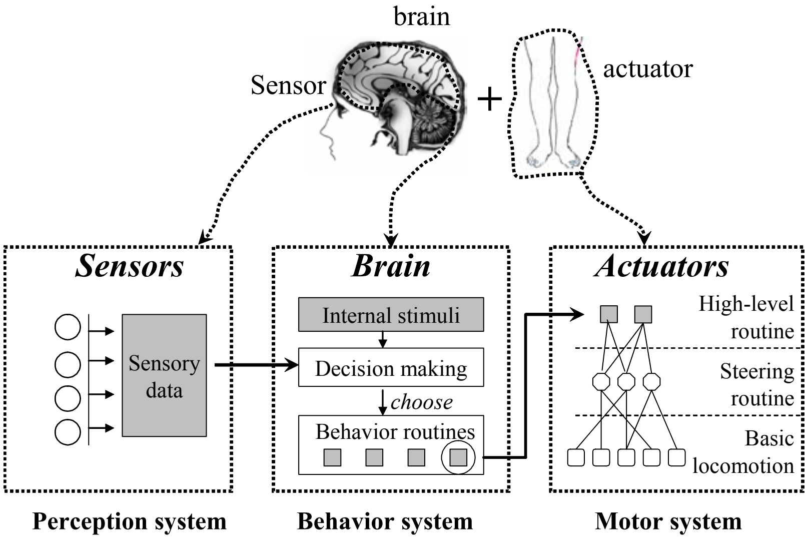


Figure 2: The three subsystems of an autonomous agent

- The *Events Recorder* is intended to capture the events that have been simulated for retrieval and playback. The simulated results can be recorded for further analyses, for example, to derive evacuation patterns and statistical information. The events captured can also be used to compare with known and archived scenarios.
- Visualization is important to display the simulation results. The *Visualizer,* which is currently built using OpenGL, receives the positions of agents, and then dynamically generates and displays 2D/3D visual images.
- The *Crowd Simulation Engine* is the key module of the multi-agent simulation system. Based on the behavior models and rules classified, each agent is assigned with an “individual behavior model” based on the data generated from the population generator. An Individual Behavior Model is composed of three subsystems – a Perception System, a Behavior System, and a Motor System, which will be discussed in details in the next section.

AUTONOMOUS AGENT

In MASSEgress, a human occupant is represented as an autonomous agent who interacts with the virtual environment and with other agents. As depicted in Figure 2, each agent senses the situation and the environment, makes decision and acts/reacts according to its behavior model.

PERCEPTION SYSTEM

The perception system of an agent consists of one or more sensors. A sensor contains three components: a set of input parameters, a sensory mechanism, and a set of output parameters to hold sensory data. The input parameters represent some aspects of the environment that a sensor attends to. The sensory mechanism is an algorithmic procedure that processes the input parameters and then produces sensory data for further processing by the behavior system.

There are many sensory information, such as visual, audio, and emotions or tensions etc., that could affect individuals' decision making in an evacuation situation. Currently, MASSEgress has implemented a set of computational methods to simulate the visual sensor of an agent. The input parameters of the visual sensor include: exits, assembly points, other agents, and obstacles. As for the sensory mechanism, we have adopted the concept of view volume (Figure 3), which is a visual cone defined by a perception range and a view angle. The view volume represents the basic constraint of an agent's visual perception in that an object is visible only if it falls within the view volume and is not occluded by any obstacle.

Computing the visibility of a virtual agent has been a subject of research in computer graphics and the modeling of digital actors, where many efficient methods have been developed. Based on experimental results, a hybrid approach which combines a point test algorithm and a ray tracing algorithm is adopted. The point test algorithm is used to determine whether or not an exit or an assembly point is visible to an agent. Given the eye position **O**, the view angle **ɵ** and the perception range **Vr** as shown in Figure 3, the procedure in testing the visibility of a point **P** can be described as follows:

```
PROCEDURE (Point Test):
IF (the distance |P – O| ≤ Vr) AND
     (the line OP lies within the view volume spanned by angle ɵ) AND
       (the point P is not occluded)
   THEN P is visible;
   ELSE  P is not visible;
```

The point test algorithm is simple and effective when an agent needs to test against exits and assembly points which are typically small in number for a floor layout.

The ray tracing algorithm is designed to compute the visibility of static obstacles. One observation is that a person does not always see all objects within his/her view volume simultaneously. Instead, depending on what draws the person's attention, the person usually focuses on one object at a time. Therefore, it is not necessary to compute the visibility of all objects that fall within an agent's view volume but only those objects that are relevant at the time. When surrounded by a large number of objects, those nearby objects closest to the person are to be tested first. Given the eye position **O** and the perception range **Vr**, and let **m** ϵ (0, 1] be a real number and **ε** be a small positive constant**,** the ray tracing algorithm for computing the visibility of obstacles can be described as follows:

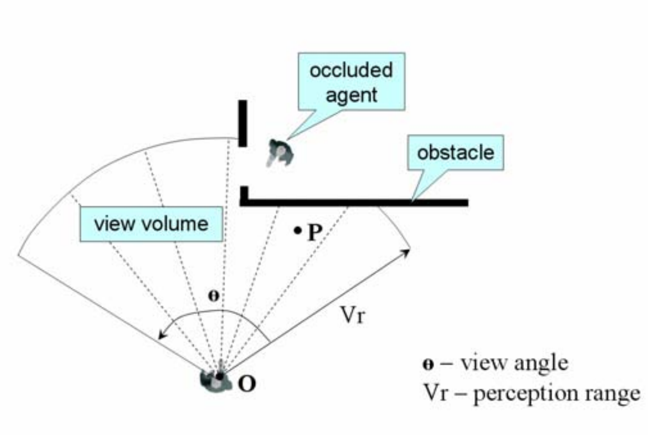


Figure 3: View volume

```
PROCEDURE (Ray Tracing):
Initialize m = 1;
WHILE simulation is not terminated, for each time step:
        From position **O**, cast three rays (left, middle and right) with length $\mathbf{LR} = \mathbf{m} * \mathbf{Vr}$;
        IF any ray intersects with an obstacle
        THEN record the intersection;
                set $\mathbf{m} = \mathbf{m}/2$;
        ELSE set $\mathbf{m} = \mathbf{m} + \boldsymbol{\varepsilon}$;
    END;
```

This algorithm works well for an agent to detect nearby obstacles as to avoid colliding with them while moving in a space. Since it emulates the 'attention-switch' aspect of a human cognitive system, the algorithm not only is efficient but also helps mimic human-like micro behaviors.

BEHAVIOR SYSTEM

The behavior system acts as the "brain" of an agent. Based on the sensory data received from the perception system, the agent then takes into consideration of some internal stimuli (i.e., psychological and sociological factors) and makes specific behavior decision. The main components of the behavior system are decision making rules which are organized as decision trees. A set of complex decision rules can be organized as a single decision tree, where a non-leaf node represents a condition and a leaf node represents a behavior decision. Different decision trees can be developed to represent different behavior models.

It should be pointed out that the decision-making rules and behavior routines can be tested independently before integrated into the system. The design has the flexibility and extensibility that allows modeling the psychological and sociological aspects of an agent's behaviors, and incrementally incorporating them into the system.

MOTOR SYSTEM

Once the behavioral system selects a behavior decision, the motor system executes the corresponding behavior routine. A behavior routine is composed of one or more steering behaviors, and each steering behavior in turn consists of a sequence of basic locomotion.

In MASSEgress, a behavior decision represents the intention of an agent, such as "determining an exit and then moving towards it". The execution of a behavior routine can be outlined as follows:

```
PROCEDURE (Behavior Routine):
IF a goal point is chosen
    THEN compute a steering angle toward the goal point;
            choose a steering behavior;
            execute the steering behavior;
    ELSE explore the space randomly.
```

The concept of steering behavior has been studied in robotics and artificial life. Steering behaviors are essential for an autonomous agent to navigate its virtual environment in a realistic and improvisational manner. Combining steering behaviors can be used to achieve higher level goals (Reynolds 1999), such as getting from here to there while avoiding obstacles. For each time step, the main function of a steering behavior is to "compute and execute a legitimate move", which can be described as follows:

```
PROCEDURE (Steering):
FOR i = 1, …, k (some constant integer)
    anticipate a move by calling a basic locomotion routine;
    IF the move is legitimate
        THEN execute the move;
               EXIT;
END;
ABORT operation.
```

Note that the number of search steps for a legitimate move is restricted to some constant k. This is because sometimes a legitimate move might not be possible (e.g., when an agent is surrounded by a dense crowd). The selection of locomotion routine is done according to the decision tree representing the logical steps of a particular steering behavior.

A particular steering behavior is made up of a sequence of locomotions. A basic locomotion represents the simplest movement that an agent can conduct through its actuators, such as "moving forward one step." The basic steps of locomotion can be described as follows:

```
PROCEDURE (Locomotion):
update agent speed;
compute agent body orientation according to the direction of a goal point;
compute the coordinates of the new position;
```

Locomotive types of an agent are defined according to some basic human movement that the system intends to simulate. In MASSEgress, eight basic locomotive types are defined. They include moving forward, turning left, turning right, making a U-turn, shifting left, shifting right, moving backward, and stopping.

SIMULATION OF HUMAN SOCIAL BEHAVIOR

Social behaviors are complex phenomena emerged from the interactions of a group of autonomous agents. In the prototype implementation, for each time increment, the agents are selected in a random fashion and do not have a pre-defined order of execution. At the microscopic perspective, each agent's behavior is essentially nondeterministic. Therefore, if the system is executed multiple times with the same initial setting, the agents would not behave exactly the same way each time due to the randomness embedded in the system. However, at a macroscopic level, certain behavioral patterns could be observed across the multiple simulations. These social behavioral patterns are called emergent phenomena. As of this writing, MASSEgress can demonstrate social emergent phenomena including bi-directional crowd flow, competitive, queuing, and herding behaviors (Figure 4).

The behavior of an agent is primarily driven by its decision rules. By defining the interactions among the agents through their behavior rules, social pattern is emerged through a hierarchically structured (from simple to complex) set of activities: locomotion, steering, and social. As an example, for a group of agents to form a queue at a narrow exit, the process could involve (1) the motion (such as moving a step) of an agent that takes place at the locomotion layer, (2) avoiding obstacle using a steering motion, which consists of a sequence of different locomotion, (3) exiting a door in an orderly manner as a type of social behavior that involves several steering activities.

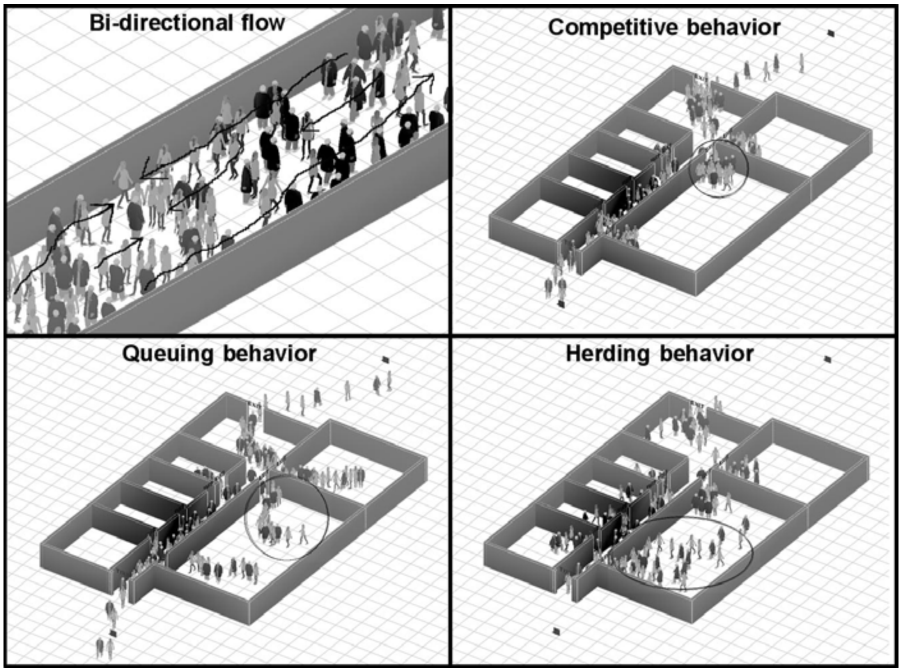
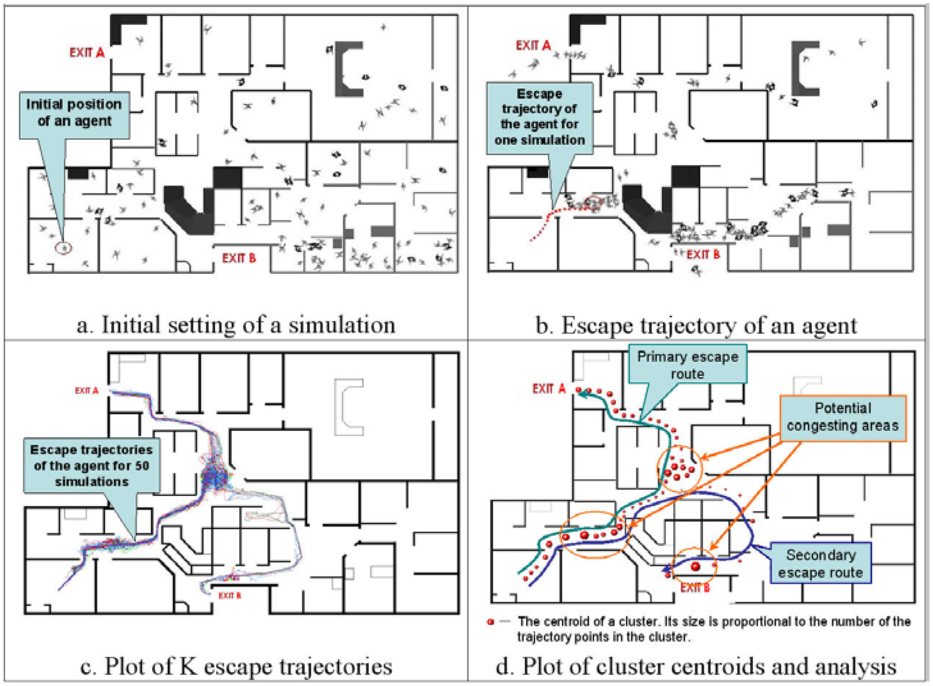


Figure 4: Social behavior simulation

STATISTICAL ANALYSIS OF EVACUATION PATTERN USING CLUSTERING ALGRORITHM

A computational tool like MASSEgress can potentially be used for many practical applications. One example is to facilitate egress analysis for building designs. When designing a floor plan of a building, although the intended usage of the space is usually known, it remains difficult to account for every possible scenario for safe evacuation, because of the uncertainties such as spatial distributions of the occupants and their behaviors. However, with the layout of a floor plan, some typical evacuation patterns can be drawn statistically by conducting multiple evacuation simulations with different occupant configurations.

To illustrate, Figure 5a shows a floor plan with two exits, A and B. At first, we place a 'test' occupant in a specific room with the presence of other occupants distributed randomly in other spaces. Evacuation simulations are then performed many times, with different spatial distribution of the occupant. That is, for each simulation, while fixing the location of the 'test' occupant, we randomize the locations and behavioral types of other occupants, so that the 'test' occupant would exhibit different evacuation behaviors for a range of different situations. Figure 5b shows an example escape trajectory of the 'test' occupant in one of the simulations. Figure 5c shows the trajectories of the 'test' occupant from 50 simulations. Using a K-Means clustering algorithm (MacQueen, 1967), the trajectory points are categorized into clusters represented by a set of centroids. The resultant centroids are plotted as shown in Figure 5d, and the size of each centroid reflects the number of trajectory points that the centroid contains. By analyzing the distribution of the centroids, we can identify the primary and the secondary escape routes of the occupant, the relative frequency for the usage


EXIT A
Initial position of an agent
EXIT B
EXIT A
Escape trajectory of the agent for one simulation
EXIT B
EXIT A
Escape trajectories of the agent for 50 simulations
EXIT B
Primary escape route
EXIT A
Potential congesting areas
Secondary escape route
EXIT B
The centroid of a cluster. Its size is proportional to the number of the trajectory points in the cluster.

a. Initial setting of a simulation

b. Escape trajectory of an agent

c. Plot of K escape trajectories

d. Plot of cluster centroids and analysis

Figure 5: Statistical analysis of escape routes and potential congestions

of the routes, and the potential congested areas during evacuations. By exploring different geometric configurations and re-arranging exit signs, a designer can modify the floor plan to alleviate congested areas and to provide more efficient egress routes.

SUMMARY AND DISCUSSION

This paper describes a multi-agent based computational framework, MASSEgress, that simulates human and social behavior during emergency evacuations. Such a computational framework not only can provide insight to current prescriptive and, often, ambiguous codes and provisions for egress design, but also can potentially serve as a means to study safety engineering, such as assessing building designs, testing safety and evacuation procedures, and assisting crowd management. Our future work will include incorporating other sensory information, behaviors (such as leader and follower relationships and affiliation) and physical models (such as knocking, pushing, etc.) and to further extend the tool to perform statistical analysis of evacuation patterns, times, flows and other design parameters.

ACKNOWLEDGMENTS

This research is partially supported by a seed research grant by the Center for Integrated Facility Engineering at Stanford University. The authors would like to acknowledge the software support from AutoDesk, Inc.

REFERENCES

AEA Technology, (2002). *A Technical Summary of the AEA EGRESS Code*. Technical Report, AEAT/NOIL/27812001/002(R), Issue 1.

Bryan, J. (1997). "Human Behavior and Fire." in Cote, A. (editor) *Fire Protection Handbook*, Eighteenth Edition, National Fire Protection Association, pp. 8.1-8.30.

Chertkoff, J. and Kushigian, R. (1999). *Don't Panic: The Psychology of Emergency Egress and Ingress*. Praeger, London.

Epstein, J. and Axtell, R. (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press, Cambridge, MA, 1996.

Galea, E., (editor)(2003). *Pedestrian and Evacuation Dynamics*. London, CMC Press, 2003

Gottschalk, S., Lin, M.and Manocha, D. (1996). "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection." *SIGGRAPH*, 1996.

Halcrow Group Ltd., (2003). *Pedroute*. (available at http://www.halcrow.com/pdf/urban_reg/pedrt_broch.pdf)

Helbing, D., Farkas, I., and Vicsek, T. (2000). "Simulating Dynamical Features of Escape Panic," *Nature*. 407:487-490.

ICBO, (2000). "Means of Egress." *2000 International Building Code*. pp. 211-247.

Johnson, S. (2001). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Simon & Schuster, New York, 2001.

Legion International Ltd. (2004). *Legion*. (available at http://www.legion.biz/)

MacQueen, J. (1967). "Some Methods for classification and Analysis of Multivariate Observations." *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press, 1:281-297, 1967.

Pan, X., Han, C., Dauber, K. and Law, K. (2006a). "Human and Social Behavior in Computational Modeling and Analysis of Egress." *Automation in Construction* (in press).

Pan, X., Han, C., Dauber, K. and Law, K. (2006b). "A Multi-agent Based Framework for the Simulation of Human and Social Behaviors during Emergency Evacuations." AI and Society (accepted for publication).

Proulx, G., (2001). "As of Year 2000, What Do We Know about Occupant Behavior in Fire?" *The Technical Basis for Performance Based Fire Regulations*. United Engineering Foundation Conference, San Diego, January 7-11.

Reynolds, C. (1999). "Steering behaviors for autonomous characters*." Proceedings of Game Developers.* San Jose, CA, March 15-19, pp. 763-782.

Santos G. and Aguirre, B. (2004). "A Critical Review of Emergency Evacuation Simulation Models." *Proceedings of Building Occupant Movement during Fire Emergencies*. Gaithersburg, Maryland, June 10-11.

Shields, T. and Proulx, G. (2000). "The Science of Human Behavior: Past Research Endeavors, Current Developments and Fashioning a Research Agenda." *Proceedings of the Sixth International Symposium on Fire Safety Science*. IAFSS, 2000, pp. 95-114.

Society of Fire Protection Engineers, (2002). *Engineering Guide to Human Behavior in Fire*. Technical Report.

Still, G. (2000). *Crowd Dynamics*. Ph.D. thesis, University of Warwick, UK.