Pomona College
Department of Computer Science

Optimizing Leader Proportion and Behavior in Building Evacuations

Paul McCormack

April 7, 2012

Submitted as part of the senior exercise for the degree of
Bachelor of Arts in Computer Science

Professor Tzu-Yi Chen, advisor

Abstract

The modeling of human crowds has applications ranging from entertainment to public safety. Health and safety officials rely on agent-based simulations to plan emergency exit routes, to anticipate congestion, and to predict ways to minimize evacuation casualties. Beginning with an investigation of current crowd models, we build a simulation to investigate how the proportion, behavior, and knowledge of trained leaders affects evacuation success. We base our simulation on the Helbing Social Force Model and incorporate other algorithms to facilitate collision avoidance and high-level wayfinding. New features include an extended collision avoidance algorithm and a high-level wayfinding algorithm that allow agents to avoid large crowds and to minimize time to exit rather than distance to exit. As expected, we found that increasing leader proportion speeds evacuation. In particular, we found that an evacuee to leader ratio of 20:1 is optimal and that leaders should guide rather than lead agents out of the building. Providing leaders with real-time communication and surveillance can help them optimize evacuee flow, especially in large buildings. These results are consistent with previous research and guidelines set by the Occupational Safety and Health Adminstration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Spurred by a rapidly growing list of applications in entertainment, virtual training, transportation planning, and public safety, crowd modeling has developed a large base of research over the last 15 years (see [Cha10]). Crowd modeling initially branched off of traffic modeling, which helps infrastructure planners design highways and roundabouts to maximize flow using one-dimensional simulations. As noted in [BND11], "the mathematical structure to be used for the modeling of crowd dynamics is analogous to that of vehicular traffic, but in a vector form to take into account the fact that pedestrians move in more than one space dimension" (427). Crowd modeling has been less studied and, until Helbing's work in 2000 ([HFV00]), had experienced few major developments. In fact, until recently, large-scale crowd simulations were not possible due the complexity of human interactions with other pedestrians and their environment. Even now, with much greater computing resources, researchers struggle to imitate human behavior in individual agents. Some effort has been made to simulate the human decision-making process with the belief-desire-intention model and others described in [LZC+08]. Most simulations select several features of human behavior that, when injected into many agents, cause the agents to exhibit desired emergent behaviors. By building models that are sufficiently accurate and complex, we are able to investigate how many different types of crowds will behave in varying environments and scenarios. Our model incorporates sufficient agent traits that we observe emergent behaviors and are able to make conjectures about how leader behavior affects an evacuation.

Our work is motivated by the recent rise in the number of high-occupancy buildings, which has increased the need for evacuation and safety planning. Such planning emphasizes the need for evacuation drills, warden training, and exit route planning but has also influenced building architecture and design. The Occupational Safety and Health Act of 1970 formalized and strengthened building requirements, specifying that any public location must have alternate exit routes and that evacuation route signs must be posted. According to OSHA (Occupational Health and Safety Adminstration):

> The employer should assure that an adequate number of employees are available at all times during working hours to act as **evacuation wardens** so that employees can be swiftly moved from the danger location to the safe areas. Generally, **one warden for each twenty employees** in the workplace should be able to provide adequate guidance and instruction at the time of a fire emergency. The employees selected or who volunteer to serve as wardens should be trained in the complete workplace layout and the various **alternative escape routes** from the workplace [Occ80].

Federal guidelines suggest evacuation drills involving all employees in addition to the detailed training of "evacuation wardens", or what we call *trained leaders*. We design an agent-based evacuation simulation to determine an optimal ratio of leaders to agents, the most suitable leader behavior, and the ideal level of leader knowledge about alternate escape routes and realtime building congestion information. We then compare these guidelines to our findings.

Beginning with an investigation of current crowd models, we build a simulation to determine how the proportion, knowledge, and behavior of trained leaders affects evacuation success. Similar to in [PB06], all agents know at least one evacuation strategy, whether it be following exit signs and maps, or departing in the same way they entered the building. Therefore, in our simulation, all agents have a complete knowledge of the floorplan, but not the hazards in it. Leaders communicate with each other by radio and can inform agents within hearing distance of hazards throughout the building. The percentage of leaders in the population and their behavior can vary. New features in our model include an improved collision avoidance algorithm that incorporates both slowing and turning as well as short-term path optimization that allows agents to avoid dense crowds ahead of them. We expand on Pelechano's work in [PB06] by adding high-level wayfinding, where agents and, in particular, leaders, can plan evacuation paths based on observed crowd densities and flows. Pelechano mentions this idea as a possible extension in her future work section. Thus, our model extends previous work in the area by combining the idea of leaders with more complex leader behavior and knowledge.

Like [PB06], we base our simulation on the Helbing social force model, described in [HFV00], with some additions from [HFV02] to simulate panic propagation. After including corrections to the model enumerated in [LKF05], we incorporate collision avoidance, [KZ11], and high-level pathfinding. We also add "cognitive maps" to each agent, giving trained agents an increased real-time knowledge of the map by way of radio communication with other leaders.

We validate our model by observing emergent behaviors, such as doorway crowding and herding, as was done in [PAB07] and [HFV00]. In addition, we qualitatively compare our graphical results to those found in [PB06].

These new features in our model allow us to explore how this additional complexity enhances leaders' ability to facilitate evacuations. Not only do we seek to discover an optimal, yet practical, percentage and knowledge level of trained agents in the population to expedite agent evacuation, but also, for example, we investigate how having leaders adopt a *directing* vs. *leading* strategy helps or hinders an evacuation. Such knowledge could lead to better evacuation drill planning and emergency communication systems.

We find that the optimal proportion of evacuees to leaders is 20:1. This leader presence decreased evacuation time by 40%. Our findings indicated that these leaders should guide agents (i.e. give them directions) rather than leading their followers out of the building. The latter behavior produces clustering and crowding around the leaders and impedes flow, thus increasing evacuation time by about 25%. While leader knowledge of realtime crowd dynamics has less of an impact on crowd evacuations in small, single-level floorplans, knowledge of alternate exits is critical. The new features in our simulation allow for better evacuation prediction because they incorporate complexities of human behavior such as mutual cooperation and long-term planning. Future work might indicate that realtime congestion knowledge helps considerably in larger buildings. Finally, as OSHA suggests, leaders should conduct regular evacuation drills to minimize panic and increase communication between non-leader agents. This communication can almost halve the total building evacuation time. These results are consistent with the guidelines from OSHA, which were created to expedite evacuations during disasters.

Chapter 2

Background

A pedestrian simulation attempts to describe how people interact with each other and their environment. The underlying model ranges in complexity based on the application. At the largest scale, a simulation might rely on flow equations, such as the Navier-Stokes equation, to determine pedestrian flow between adjacent cells in a cellular automata grid. Pure cellular automata models can simulate the motion of large, dense crowds with up to 100,000 agents, and adding some statistical contributions can further increase their accuracy. See [BND11] for an overview of crowd modeling approaches. The primary benefit of these models is that they are very fast, but they lack the ability to accurately model low-density crowds. For this reason, [NGCL09] and others create hybrid models that rely on this type of "aggregate flow" simulation combined with a more individualistic model for low densities.

To add a bit of granularity between agents, some researchers represent individuals as particles, interacting according to systems like Smoothed Particle Hydrodynamics described in [Aue09]. Similar to aggregate flow models, particle simulations are quick to calculate but lack the individualism apparent in real pedestrians. Even though people are represented as individual particles, each particle behaves according to the same rules or equations as every other one. This limits the ability of these types of simulations to represent real crowds of varying densities, since people are actually individuals that interact according to personal attributes and complex decision-making patterns.

In the past 15 years, computers have become fast enough to represent individuals as agents in the simulation. The most important contribution of this type of model is that agents can be unique – one might be impatient while another might feel strong altruism toward several others in his family. An agent interacts with other agents and obstacles according to rules or forces based on these properties. These simulations do not perform as well as larger-scale models for big crowds partly because they can be extremely slow to calculate and partly because people in high-density crowds do tend to behave like particles or an aggregate flow. For example, see the figure in [NGCL09] of the Mecca. However, for smaller crowds, it allows for a richer representation of pedestrians that increases result accuracy.

In this study, we used an agent-based model to simulate crowds of about 1000 pedestrians. Our primary goal was to investigate how trained leader agents can affect evacuation time, so the individualism provided by agent-based models was necessary.

2.1 Agent-Based Modeling

In [HFV00], Helbing developed his well-known Social Force model, which formed the basis for a multitude of later extensions. As the father of modern crowd modeling, he looked to work by social psychologists to derive his list of nine characteristics of panicking human crowds. These include 1) moving faster than normal, 2) pushing, 3) bottlenecks, 4) arch-shaped clogging at exits, 5) jams, 6) dangerous interpersonal pressure, 7) slowing due to obstacles, 8) mass behavior, or herding, and 9) overlooking alternative exits. He describes several of these behaviors in detail in his later work, [HFV02].

From these principles, he developed an agent-based model that describes each person, p_i with a mass, m_i, having a desired speed, v_i, in direction $\mathbf{e_i}$, and an instantaneous velocity, $\mathbf{v_i}$. Each agent will adjust its instantaneous velocity to match its desired velocity with a characteristic time τ_i. In addition, each agent interacts with other agents according to a repulsive interaction force. Contact and "sliding friction forces" are applied if agents are sufficiently close to each other. He observed that agents tend to either crowd near exits in a herd-like manner or to behave completely individually and find other exits. Neither situation is ideal, he argues, as one would like the evacuation flow to be equally divided among all exits.

Five years later, Lakoba addressed some issues in the Helbing model in [LKF05]. First, he criticizes the approximation that Helbing made in calculating inter-personal forces. Helbing, he describes, neglected to include a "maximum squeezing distance", d_0, for agents in the crowd, which means pedestrians can overlap each other. Instead, Lakoba used a social force function between people with an artificially high coefficient that would simulate these collisions. However, at equilibrium distances of 50 cm, this function behaves inappropriately, exerting an outward force of 5 N on each agent involved. For large crowds, this approximation may be sufficiently accurate, but it does not give good results for less dense crowds where individual interactions matter more. For this reason, Lakoba investigates other collision algorithms in an attempt to find one more appropriate for crowd modeling. One such model creates a vertical asymptote in the force as the distance between agents approaches $2d_0$. This modification effectively creates an infinite repulsive force between two completely squished agents and so prevents overlaps. Unfortunately, this model requires smaller and smaller time steps as the agents become closer, making it computationally inefficient. In addition, all existing algorithms required solving for all interactions between agents, even though the interaction matrix is sparse since agents only need to interact with others within three meters. If the model has n agents, the algorithm needs to calculate $O(n^2)$ forces instead of $O(n)$ in each simulation iteration.

To correct the Helbing model, Lakoba creates an "overlap eliminating algorithm". First, the most overlapped agent is identified. It is moved out of the wall, if it overlaps, and set to be "stationary", meaning that it cannot be moved for the rest of the round. All agents overlapping the stationary agent are moved and assigned new velocities corresponding to it. The algorithm then performs the same procedure until no overlapping agents are found but no more times than the number of pedestrians in the room. The simulation time step is determined by the distance between the closest pair of agents. The algorithm drastically reduced the running time of the simulation and allowed [LKF05] to adjust the interpersonal force coefficient to more accurately reflect human crowds.

Several types of crowd modeling stemmed from Helbing's novel work. Pelechano et al. ([PAB07]) group these into three categories: force-based, rule-based, and flow-based. Helbing and Lakoba both focused on primarily force-based algorithms; we also use a force-based algorithm in our model. In these models, individuals behave like particles with some element of social force between them. The researchers in the following years lamented the lack of individualism in the agents, and some even tried to model the human decision-making process.

2.2 Model Specialization

With the advent of more powerful computers and a greater demand for realistic models, works such as [BMdOB03], [Hu06], and [LZC$^+$08] built increasingly complex simulations to describe how human behavior affects crowd dynamics. In [BMdOB03], Braun et al. introduce altruism and social dependence into their model. As crowds are not generally composed of equally selfish individuals, they aimed to observe how changing the composition of the agent population affected evacuation success. Altruism is represented as a force drawing an agent towards its group (such as family or friends). Dependence reflects how capable an agent is of dealing with stress: a child or disabled person has high dependence and may tend to stop moving altogether in an emergency. Braun's test results exhibited group formation and less efficient evacuation in general.

Moving past the force-based model standard, [Hu06] characterizes agents with *behavioral context* in a rule-based state model. Individuals make adaptive choices based on their environment. Each individual occupies a single behavioral state at any time, and, at a higher level, maintains a behavioral context state. Behaviors include casual walk, explore point, maintain personal space, follow crowd, and flee to exit. Each behavior contains *inhibitory coefficients* that prevent an agent from moving into that state from another. These coefficients can be changed by the behavioral context state. For example, a panicked agent will be more likely to "flee" (a panicked, semi-irrational behavior) than one not so afflicted. Hu's paper made important contributions in the area of rule-based simulations by incorporating behavioral states.

Luo in [LZC$^+$08] extends these discoveries to build a three-layer generic model that aims to "naturally reflect human decision making due to external stimuli". His daunting goal demonstrates the increasing focus on individualistic agents. The top level is the crowd level, modeling social relationships such as altruism. The second layer is the individual level, where an agent processes sensory inputs and accumulates situation awareness. This level sends information to the third, physical layer which forms a list of basic actions to interact with the world. In addition to this complex system, Luo et al. include a plethora of other variables such as energy level, emotional attraction, and panic.

Not all researchers developed more individualistic agents. Narain et al. in [NGCL09] instead formulated large-scale crowd modeling algorithms for use with 50,000 to 100,000 agents. Previous simulation techniques were infeasible due to the complexity of agent interactions and collision detection. Narain develops an aggregate flow cell-based model. He relies on an assumption he terms *unilateral incompressibility*, meaning that agents in an extremely dense crowd behave like an incompressible fluid. Each cell contains a pressure gradient and flow, so he avoids most of the agent collision detection. The global planner determines individual velocities, these velocities are "splatted" to a cell, the model performs the unilateral incompressibility calculation, and the actual agent velocities are extracted. Not wishing to sacrifice individual movement or over-constrain less dense crowds, he determines each agent's velocity based on an interpolation between the preferred velocity and flow velocity depending on the crowd density. For example, in an extremely dense crowd (such as at the Mecca), agents must move entirely with the flow whereas agents at a large campground can move almost independently. In this way, the model uses a mixture of Eulerian and Lagrangian principles. Pelechano in [PAB07] also experiments with large, high density crowds but incorporates some individual behavior such as politeness and impatience. Her model creates an attractor point in each room which draws all agents. She successfully simulated behaviors such as queueing, pushing, and panic propagation.

In an earlier paper, [PB06] tackle a novel concept in evacuation modeling: that all agents do not have equal information. They combine Helbing's social force model with high-level wayfinding and varied agent roles. In general, an individual in an evacuation may not have a complete *cognitive map* of the building and so will either proceed to the nearest known exit or will explore unknown areas. In addition to this possibly incomplete internal blueprint, each agent has an orientation (where it is), and an ability to navigate or explore. It can communicate its knowledge to other agents with varying degrees of ability. All agents in the model also have some personality traits as described in previous papers. More importantly, some agents will initially have greater knowledge. These "leaders" can inform other agents of nearby exits or alternative evacuation routes. Pelechano et al. investigate how the presence of these leaders affects evacuation efficiency. They determine that even a few such agents can drastically reduce the casualties in an emergency.

As Helbing notes in [HFV00], "One of the most disastrous forms of collective human behaviour is the kind of crowd stampede induced by panic, often leading to fatalities as people are crushed or trampled". This reasoning, as well as interest in convincingly portraying crowds in entertainment and video games, has brought crowd modeling to the front of current simulation research. Beginning only 11 years ago, Helbing introduced his novel social force model, which served as the basis for countless others. Some moved in a more individualistic direction and others toward large-scale simulation. However, all may be important in modeling evacuation scenarios.

Chapter 3

Our Model

Of the many models developed to simulate crowd dynamics, we selected a few that gave rise to emergent behaviors presented in [LKF05] and allowed us to investigate how the presence of leaders affects evacuations. These models were selected and incorporated to give a force-based, rather than rule or flow-based, model that is relatively simple to implement and likely to be robust enough to exhibit emergent behavior. In addition, force-based models do not require as many esoteric parameters or state-transition rules, instead depending on intuitive quantities like personal space radius and other functions described in [HFV00].

At the base of our model is the path-finding algorithm that agents use to find exit paths. Overlaying that are a variety of agent-interaction models such as the Helbing social force model, the overlap eliminating algorithm, a panic propagation model, a collision avoidance algorithm, and a novel high-level wayfinding method.

3.1 Path Finding

Agents in our simulation rely on a visibility graph to plan minimum-distance paths to exits, in contrast to the portal-based approach in [PB06]. During pre-processing of the floorplan image, the program identifies convex corners as vertices in the graph and creates edges between all pairs of vertices within line-of-sight, as shown in figure 3.1. We chose this approach because it facilitates running simulations on arbitrary floorplans. In many portal-based algorithms, graph vertices are rooms rather than corners. So, when the connection between two rectangular rooms is particularly wide, an agent cannot optimize its path by choosing which side of the "door" it passes through without our implementing inconvenient extensions to the algorithm.

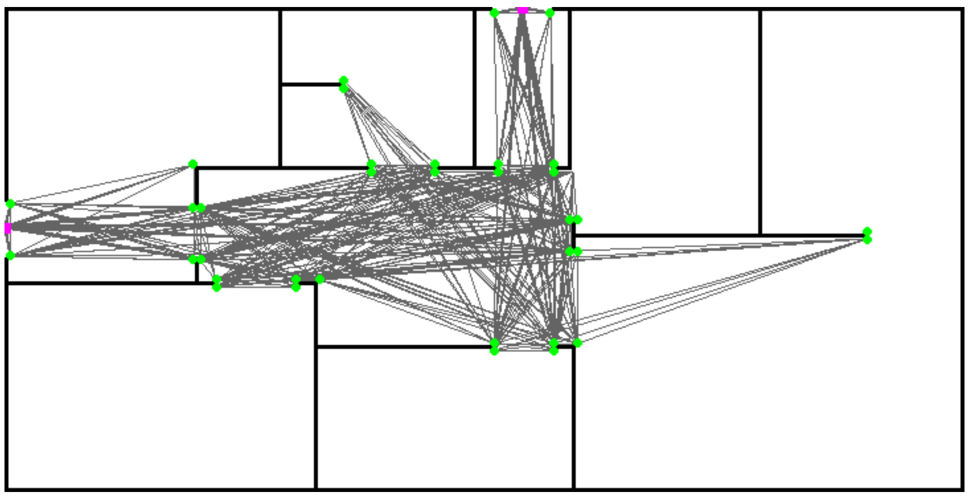

Figure 3.1: Visibility graph for simple floorplan. Lime green dots denote vertices, and gray lines are edges. Magenta boxes are destination points, or exits in the floorplan. This floorplan has 41 vertices and 400 edges.

An agent determines an optimal sequence of edges to traverse using a single-source, shortest path algorithm. Interactions with other agents may cause this agent to deviate from its planned path enough so that the next vertex in its path is obscured by a wall. An agent would then need to recalculate its exit path. In addition, an agent does not need to actually reach each vertex along its exit path; it only needs to follow each edge until the next edge in its path is visible. Several optimizations can be made to the algorithm by reusing previously calculated parent maps, recalculating new paths only when described above, and precalculating the visibility of each vertex from each pixel in the floorplan. In practice, only the last technique is possible. The dependence of exit *time* on current crowd conditions severely hampers optimization, and using individualized cognitive maps limits the effectiveness of caching previous results. Because crowd conditions can change during any time step, an agent must recalculate its path every time. Visibility precalculations of each vertex from each pixel in the floorplan greatly speeds up the simulation at the cost of a memory usage. The program stores one bitmap for each vertex in the floorplan. So, if there are n vertices and the floorplan has m pixels, nm bytes will be required. During each iteration of the simulation, an agent needs to consider whether it can still see the next vertex in its exit path. In addition, for two agents to interact, one needs to be able to see the other's next point to ensure both are on the same side of a wall. A naive approach might trace a path between each agent and its next vertex at every iteration, but storing precalculated visibilities gives a much faster implementation.

3.2 Agent Interactions

During an evacuation, agents navigate around walls and through doorways along an exit path. Each agent must consider nearby agents, obstacles, and hazards to effectively minimize its own evacuation time. Our simulation relies on four primary models and algorithms.

3.2.1 Helbing Social Force Model

The Helbing Social Force Model describes the quantitative motions of crowding agents using the concept of *personal space*. Helbing in [HFV00] describes personal space as "a psychological tendency of two pedestrians to say away from each other" by a repulsive interaction force whose magnitude is given by $Ae^{-d/B}$, where d is the distance between two interacting agents and A and B are constants. A tangential friction force resists the sliding motion of both agents while the normal force protects the agent's personal space. We choose A and B based on results from Helbing's work and determine similar constants for agent-wall interactions to ensure that an agent cannot push another through a wall. During each time step, an agent accelerates from his initial velocity, $\mathbf{v_i}$ to meet his desired velocity, $\mathbf{v_0}$ with a specified time constant, τ.

$$\frac{d\mathbf{v}}{dt} = \frac{\mathbf{v_0} - \mathbf{v_i}}{\tau} + \text{other forces}$$

As noted in the background, this model has several flaws, particularly in calculating collisions. We address these by adding some modifications detailed in [LKF05]. The resulting model forms the basis of all agent interactions in our model.

3.2.2 Overlap Eliminating Algorithm

In [LKF05], Lakoba et al describe an Overlap Eliminating Algorithm (OEA) as an alternative to Helbing's collision model. The authors reject the idea of modeling interpersonal collisions with a simple exponential function, $Ae^{d/B}$, where d is the distance between them, noting that people should never actually overlap. They explore using asymptotic functions to model interpersonal forces. Since the force between two touching agents will be infinite, it guarantees that agents will never overlap. However, solving for the motion of agents requires an impractically small time step, so they develop the OEA to govern collisions instead. During each simulation iteration, the following algorithm is run:

1. Find the most overlapped pedestrian.
2. If that pedestrian overlaps a wall, move him away from the wall and set the component of his velocity toward the wall to be 0.
3. Mark the pedestrian as "stationary", meaning that he cannot be moved again during this iteration.
4. Move all non-stationary pedestrians away from him to eliminate their overlap.
5. Repeat steps 1-4 until no overlapping pedestrians are found, but no more times than the number of agents.

In almost all cases, all overlaps will be eliminated. It is conceivable that an agent in a corner will be unable to eliminate overlaps with agents around him if those agents are already stationary.

This can be remedied by prioritizing agents that overlap walls, even if other agents may be more overlapped, which is a danger with large time steps. This ensures that pedestrians are not trapped against walls (or pushed through them, as the case may be). At reasonable densities of at least 2 sq. ft per agent, our implementation of the algorithm gives reasonable-looking results, especially after some position averaging to remove jitter. Position averaging simply displays agents at a position which is the running average of their actual position over several time steps. Agents form arch-shaped clogs at bottlenecks (see figure 3.2) and push each other out of the way. For example, an agent at the outer edge may squeeze between two agents, pushing them both farther from the door. Both pushing and clogging are emergent behaviors noted in [HFV00].

3.2.3 Collision Avoidance Algorithm

Depending on the density of a crowd, agents may attempt to avoid collisions by making small adjustments to their velocity. Our method is loosely based

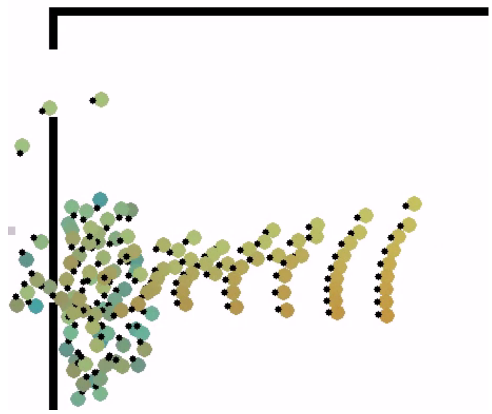

Figure 3.2: Arch-shaped clogging at an exit. Agents appear to be overlapped because of position averaging.

on a description in [KZ11], which models agents interactions through sensory input. Figure 3.3 illustrates how, in Koh's model, an agent perceives its surroundings through contact, personal space (hearing), and vision. Koh projects rays from the agent throughout its field of view. It can pay attention to a fixed number of objects at any time and will attempt to avoid collisions with these objects with a success rate dependent on its level of panic and the required deviation in velocity. In a less dense crowd, personal space increases, and agents will make greater adjustments in velocity to stay away from others. In dense crowds, agent inhibitions against contact are diminished, particularly during panicked evacuations. Interestingly, using the low-density parameters for a high-density crowd causes some agents to following winding paths through a maze of agents, almost tripling their evacuation times.

[KZ11] does not detail algorithms and methods for avoiding collisions, but it suggests that an agent will *either* turn *or* slow down. In our model, an agent will perform both according to its panic and desired speed. [HFV02] describes that panic makes agents move more quickly and impedes their ability (and desire) to avoid collisions. Therefore, a panicked agent will turn rather than slow down, but not as much as it would under normal circumstances. In figure 3.4, agent 1 will turn slightly to the right and slow down in order to minimize the change to his velocity but avoid the collision. For example, turning left would require a much greater deviation.

Koh indicates that people can psychologically only pay attention to a few objects at a time. These "attention points" are ranked according to

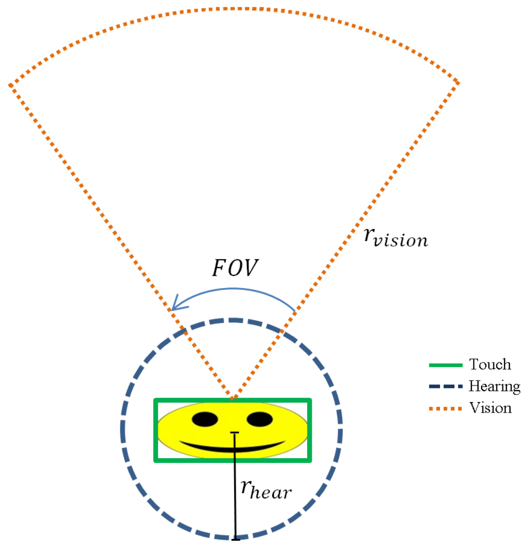

Figure 3.3: One agent's perception of its environment through vision, hearing, and touch.

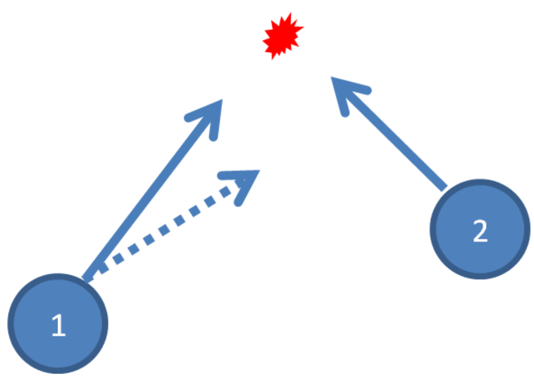

Figure 3.4: Agent 1 will turn right and slow slightly to avoid the anticipated collision with Agent 2.

collision risk and relative position and velocity. We make the addition that an agent can still be aware of agents farther away that have no chance of colliding with it. For example, a person can usually tell you the location of large clusters of people, even if they do not have any recollection of the individuals within the group. Suppose an agent anticipates a direct head-on collision with another agent. Clearly, he must turn to avoid the collision, but since turning either left or right will suffice, he considers several other factors. The first is cultural. People will generally follow traffic guidelines and prefer to pass on the right side (in the United States at least), all else equal ([KZ11]). In addition, they may consider the room's larger-scale dynamics and choose to turn to avoid passing through dense crowds. For example, see figure 3.5.

Our model is also novel in its attempt to simulate unspoken agreement between agents about which will make a change in velocity. In real crowds, people use eye contact to decide which person will turn to avoid the other, and this method is usually successful. We decided that the accommodating person tends to be the one that turn in most collisions. Similarly, a person that constantly stares at the ground will not be the one to change velocity to avoid others. So, we assign each agent a random number at the beginning of the simulation. Before an anticipated collision, the agent with the lower number will usually make the change in velocity. The method does not allow agents to always avoid collisions without drastic reductions in speed, but even real people cannot always predict the motion of others.

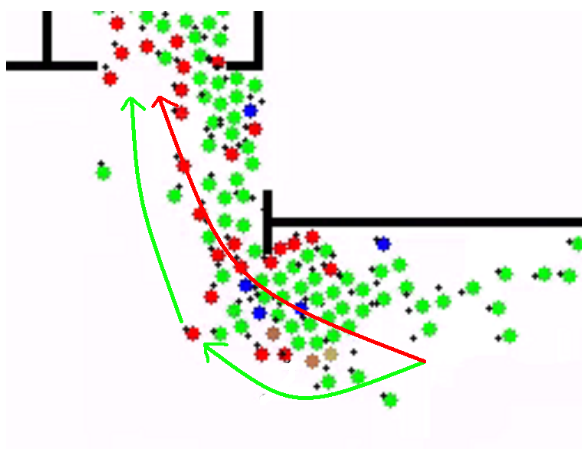

Figure 3.5: An agent will likely follow the green path to avoid agents immediately ahead of it to optimize its travel time. The figure also demonstrates how an agent can overtake several others in a typical human behavior.

3.2.4 High-level Path Optimization

To optimize its path through a single room, an agent needs to plan a route based on observed crowd motion that minimizes travel *time* rather than simply travel *distance*. For example, an agent may decide to take an entirely different exit path after observing the locations of crowds throughout his field of view. A less-crowded, longer path may allow for a speedier evacuation. Unfortunately, the average agent can only factor in crowd dynamics within his field of view. Consider figure 3.6 to compare how high-level individual planning can optimize agent flow.

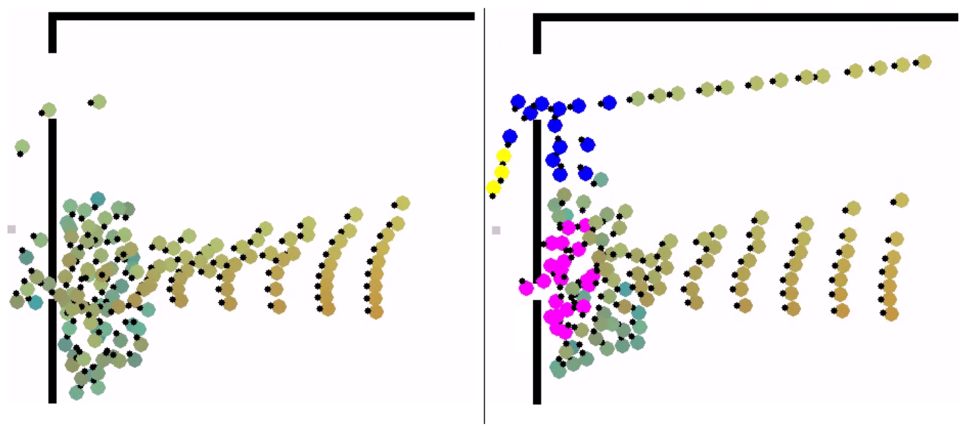

Figure 3.6: On the left, agents crowd unnecessarily around a single exit. On the right, high-level path optimization allows them to predict a short exit time through the upper door.

This feature is implemented in the simulation by creating a "path" under each edge of some predetermined, fixed width (see section 4.2). A single agent could potentially be on multiple paths at the same time, since edges overlap and their paths certainly do. From an agent's location, the simulation determines which paths the agent is on and records average path speed and density for observation by other agents. In addition, since crowds tend to form near vertices, the number of agents within a fixed crowd width (see parameters) is counted to determine density at each vertex. Agents account for densities near edge endpoints to predict their time of exit.

3.3 Hazards and Leaders

In a realistic evacuation, exits may be blocked, and hazards may impede motion throughout the building. In our simulation, agents observe these fixed-location hazards and plan new exit paths around them. [PB06] gives each agent an individual cognitive map where it can store information it learns. With only hundreds of agents and 50-100 rooms, the space requirement is small. However, in a simulation with 1000s of agents that uses a visibility graph, several gigabytes are necessary to store the most basic information in each cognitive map. Instead, in our simulation, each agent maintains a list of "disaster edges" that are impassable due to some hazard. They use these to share information and plan exit strategies.

We investigate how varying levels of communication between agents affects evacuation speed. A complete lack of communication between agents means that every agent must observe a hazard with his or her own eyes before planning a new route. This low level of communication is unrealistic – even if no verbal communication takes place, an agent is unlikely to run into a room experiencing a mass exodus of panicked people. However, at the other end of the spectrum, the exact location of all hazards cannot spread instantaneously between all evacuees. A more realistic model that we developed is that an agent hears from another agent within close quarters about some hazard with some probability. In addition, that agent's confidence in the information is less than that of the agent that communicates it. If an agent's confidence is low enough, it will not trust the information that is heard. In [PB06], Pelechano allows agents in a single room to share information about their observations of the floorplan.

In order to expedite the evacuation process, leaders equipped with radios and an admirable selflessness can direct or lead "followers" out of the building. Because these leaders share a global knowledge (due to the radios), they can inform nearby agents of hazards and guide them on alternate evacuation routes. Moreover, depending on their level of knowledge, they may know relative congestion throughout the building and optimize paths using that information. In addition to varying their proportion and distribution within the population, we vary their behavior. One one hand, they can lead small groups toward exits, and on the other they can act more as semi-stationary directors of traffic.

3.3.1 Panic and Impatience

In an evacuation scenario, panic can lead to the stampeding crowds Helbing describes. Our model relies on work in [HFV02] to model panic. An agent's panic level is increased by 1) the sight of a hazard or blocked path, 2) moving too slowly, and 3) panicking agents in the vicinity. Some factors that decrease an agent's level of panic include motion and the presence of leaders. Due to their extensive experience and training, leaders do not panic. Our model records panic level on a scale from 0 to 100 for each agent. During each time step, each of the aforementioned situations will change an agent's panic level according to their respective weights in the model. We assign weights rather arbitrarily in a way that makes intuitive sense: each panicked neighbor adds 1, a leader decreases it by 7, movement decreases it by 3, lack of movement increases it by 4, and seeing a disaster increases it by 15. As mentioned briefly before, panic generally makes an agent move faster, ignore better routes, collide more with its surroundings, and exhibit herding behavior. For example, an agent will choose to turn slightly rather than slowing slightly to avoid a collision. In the simulation, these effects are model by increasing speed, decreasing observation-based path optimization, decreasing willingness to change velocity to avoid another pedestrian, and increasing desire to follow other agents. This model of panic allows our simulation to consider how communication and leaders affect individual agent behavior.

Chapter 4

Experimental Framework

4.1 Technical Simulation Resources

We implemented our model in C++ using the Microsoft Visual C++ 2010 Express development environment. Graphics were created using the standard Windows GDI+ libraries. All other features were implemented using the standard template and C libraries. Input floorplan images were on the order of 300 pixels to a side. A single simulation run with about 1000 agents and all features takes about 25 minutes to run. All of our simulations ran on a single processor. The processor was an Intel Core i7 Nehalem E5530 (2.4 GHz, 256 kB L2, 8 MB L3). The computer had 16 GB of 1333 MHz DDR3 RAM.

4.2 The Simulation Parameters

We ran the simulation using a similar floorplan as was used in [PB06]. Figure 4.1a displays this floorplan, where hazards are shown as red boxes. Hazards could be fires, toxic gases, or other impassable obstacles. We added obstacles to the floorplan according to figures in [PB06]. Since the hazards are distributed evenly throughout the building and remain stationary, they might represent rubble after an earthquake, for example. Running the simulation with fewer hazards should decrease the importance of leaders and their knowledge. Our simulation begins with 1005 agents evenly distributed in a grid throughout the building. Leaders are also evenly distributed. Notice that agents in the room near the top right corner will not be able to evacuate. In addition to the "maze" floorplan from [PB06], we used several very simple floorplans and a floorplan of a hotel (`www.martinexecutivesuites.com`, retrieved Feb 2012) to observe emergent behavior (see figure 4.1b).

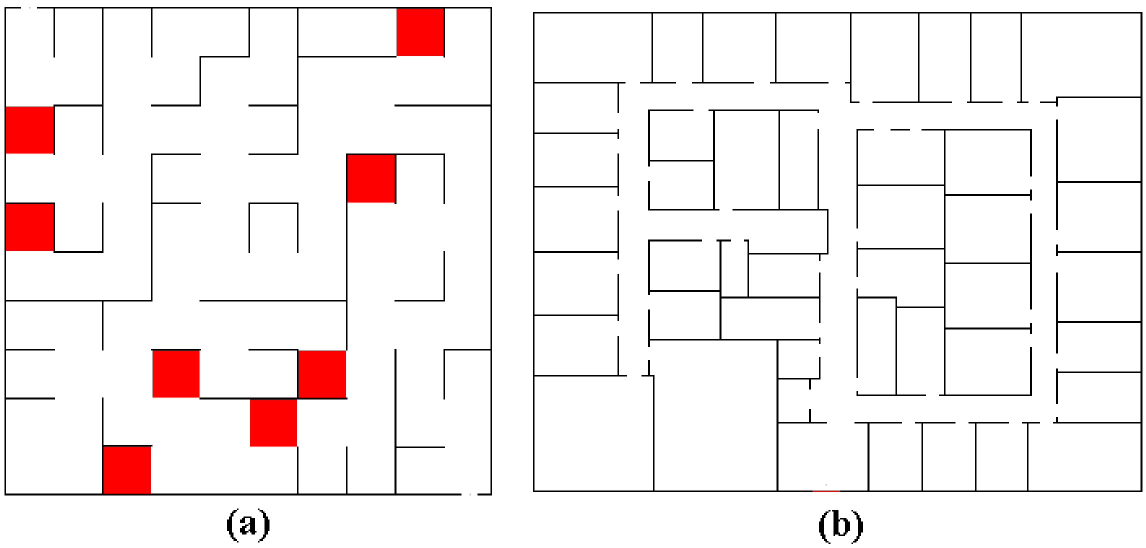

Figure 4.1: Two floorplans used for testing, validation, and final runs. Red boxes denote hazards. Hotel floorplan (b) is from www.martinexecutivesuites.com.

Our simulations test how changing a number of variables affects simulation success. The default configuration is specified below, and leader proportion is set in each experiment to highlight findings.

- No communication between non-leaders. Agents will not verbally inform other agents of hazards.

- Leaders have knowledge of crowd densities and average speeds. This knowledge should help them plan more efficient exit paths.

- Leaders guide followers out of the building rather than simply directing traffic while stationary.

- Agents do not crowd leaders but rather listen to instructions given and proceed accordingly toward the exit.

- Agents panic, and panic affects judgement and behavior.

In addition, our model depends on a number of constants, as described in table 4.1. Model parameters were determined either from background

Parameter	Value
Initial avg agent density	16 - 64 sq ft/agent
Simulation Time step	.05 sec
Pixels per foot	2 pix
Path width (High-level Path Optimization)	4 ft
Crowd width (High-level Path Optimization)	7 ft
Minimum number of agents on a path to consider avoiding	10 agents
Minimum squeezed agent diameter	1 ft ([LZC+08])
Wall-agent force (d is feet from wall)	$0.04(10-d),\ 0<d<10$ pix/timestep^2
Agent-agent force	$A=68, B=-.52$ ft. ([HFV00])
Personal space radius	3 ft
Field of view	80^o
Max viewing distance to see individuals	15 ft
Default speed distribution (uniform)	3.7 ± 1 ft/s
Time constant, τ (double if low density crowd or if panicked)	0.3 s ([HFV00])
Leader communication distance	13 ft (stationary), 8 ft (moving)

Table 4.1: Simulation Parameters

research where noted or by observation and trial and error. For instance, we set an arbitrary minimum number of agents along a path for other agents to consider them a potential obstacle to be 10 because we found that produced reasonable-looking wayfinding displays. Because the goal of our simulation was not to predict real times of evacuation, the interaction between parameters was more important than the exact values. For example, we set the wall-agent force to balance the agent-agent force when an agent was next to a wall. Most other parameters were chosen based on common sense, as these are parameters that are easily understood. For example, the width of a typical crowd near a doorway or field of view for a pedestrian are both intuitive parameters. We note that varying these parameters within reasonable bounds had very little effect on the simulation.

We determined a good value for the time step experimentally. The value was observed to be too high if agents were pushed entirely through walls before the OEA could eliminate their overlap. A very low value would make the simulation too computationally costly for experimental purposes. We found that using 0.05 seconds per time step was a reasonable compromise.

Chapter 5

Simulation Results

After developing our simulation, we validated it by comparing our results to those in [PB06] and by observing emergent behavior. We varied the proportion, knowledge, and behavior of leaders to determine which parameters are ideal for a real evacuation. The graphs we created show the number of agents left inside the building at a given time. We determined the time to *total evacuation* by the number of time steps until all agents have exited the building. In some cases, a hazard may trap agents inside the building, and they are ignored in final calculations. Approximating the actual evacuation time requires multiplying by the time step conversion factor (see table 4.1). For example, if a total evacuation takes 2600 time steps, we could predict that the actual evacuation would take about 2 minutes.

5.1 Validation

We validate our model primarily by observing emergent behaviors noted in [LKF05] and [PB06]. Several other tests validate individual components of our model like collision avoidance and the Overlap Eliminating Algorithm.

Arch-shaped Clogging Figure 3.2 demonstrates how arch-shaped clogs form when the desired pedestrian flow out of the room exceeds the doorway's capacity.

Herding and Aversion to taking detours Particularly when panicked, pedestrians tend to follow other pedestrians, even if that route is not the fastest. Similarly, [HFV02] observes that pedestrians avoid taking detours. In figure 5.1, leaders do not consider crowd density and average speed in path planning, so agents tend to follow the crowd, rather than taking the detour to their right, even when that strategy may lead to a slower exit time.

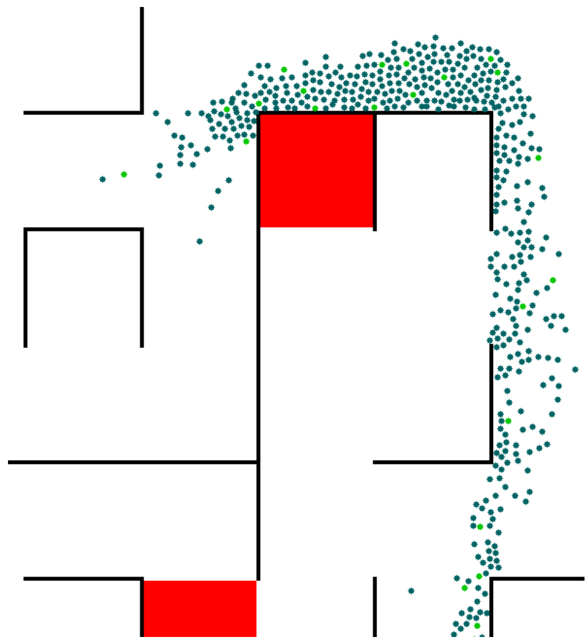

Figure 5.1: When leaders do not consider crowd density and average speed in path planning, agents tend to follow the crowd, even when that strategy may lead to a slower exit time.

Leader Crowding As observed in the simulations by [PB06], agents cluster around leaders during an evacuation. See figure 5.2. We experiment with various leader behaviors to minimize this clustering, since it can restrict leader motion. A slower leader means a slower evacuation for the entire group. Leaders that instead simply give directions can optimize flow somewhat without becoming obstacles.

Avoiding Collisions To test the collision avoidance algorithm, we have 20 agents move between each other toward opposite sides of the room, similar to in [KZ11]. Figure 5.3 exhibits how the algorithm allows agents to avoid colliding with each other.

Maintaining Personal Space In low crowd densities, maintaining personal space becomes more important to agents. To test their ability to both maintain this psychological barrier around themselves and avoid

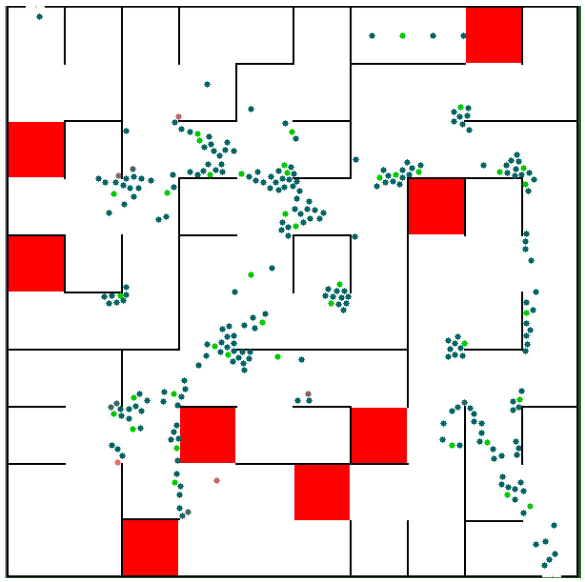

Figure 5.2: Agents cluster around leaders during an evacuation, sometimes impeding a leader's ability to guide them out. Note that the agents in the upper right are trapped by an obstacle.

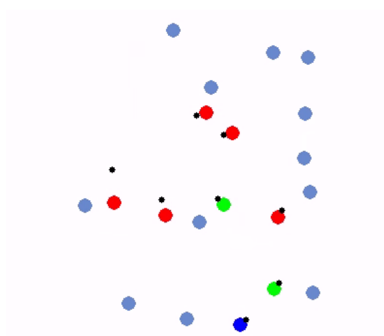

Figure 5.3: Agent moving toward opposite sides of a room avoid collisions. The "noses" in front of agents denote their instantaneous direction.

collisions, we used low-density crowd parameters in a high-density crowd. Figure 5.4 demonstrates this result.

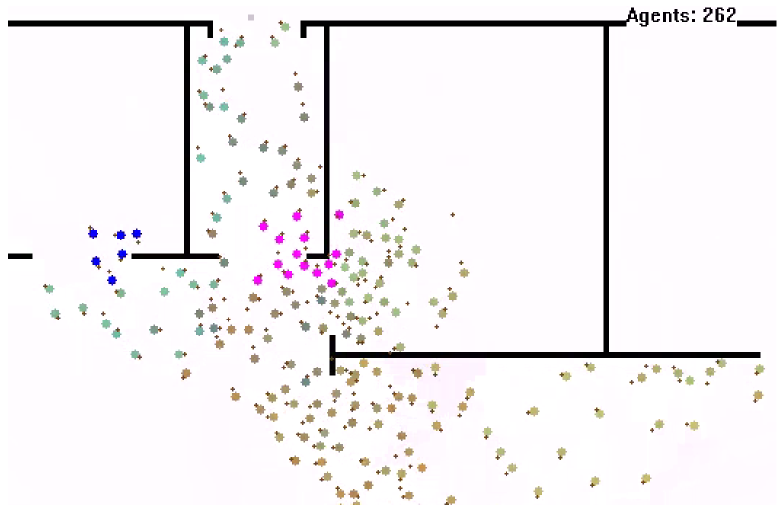

Figure 5.4: Agents preserve personal space even in a high-density crowd with adjusted parameters. Some agents seem to be heading in unusual directions to avoid collisions with other agents. The small brown dots represent agent noses which indicate instantaneous direction.

5.2 Effect of Leader Proportion

We investigated how the proportion of leaders affects evacuation success. Leaders begin evenly distributed throughout the building, like security guards in a museum or police in a subway station. In the graph, figure 5.5, the number of agents in the building is plotted against time step. In general, a greater number of leaders means that the evacuation proceeds more quickly. When there are no leaders, many agents are trapped exploring the building, looking for an unblocked exit path. The introduction of leaders allows more agents to proceed directly to an exit. Interestingly, the evacuation proceeds the most quickly during the first 1000 time steps when there are *no* leaders. Without leaders, agents closer to the exit can proceed directly out – the leader knowledge does not help them evacuate more quickly. The presence of even a few leaders will cause some clustering, not necessarily around the leaders, but along the path the leaders dictate is the best. At any instant, all agents within hearing range of a leader will decide to proceed along the same exit path. For example, a leader agent cannot tell half its followers at any one time to go one way and half to go another. This lack of granularity in leader instructions creates more congestion than if all agents decide individually how to exit. Since agents only listen to the closest leader, training more leaders can minimize path congestion.

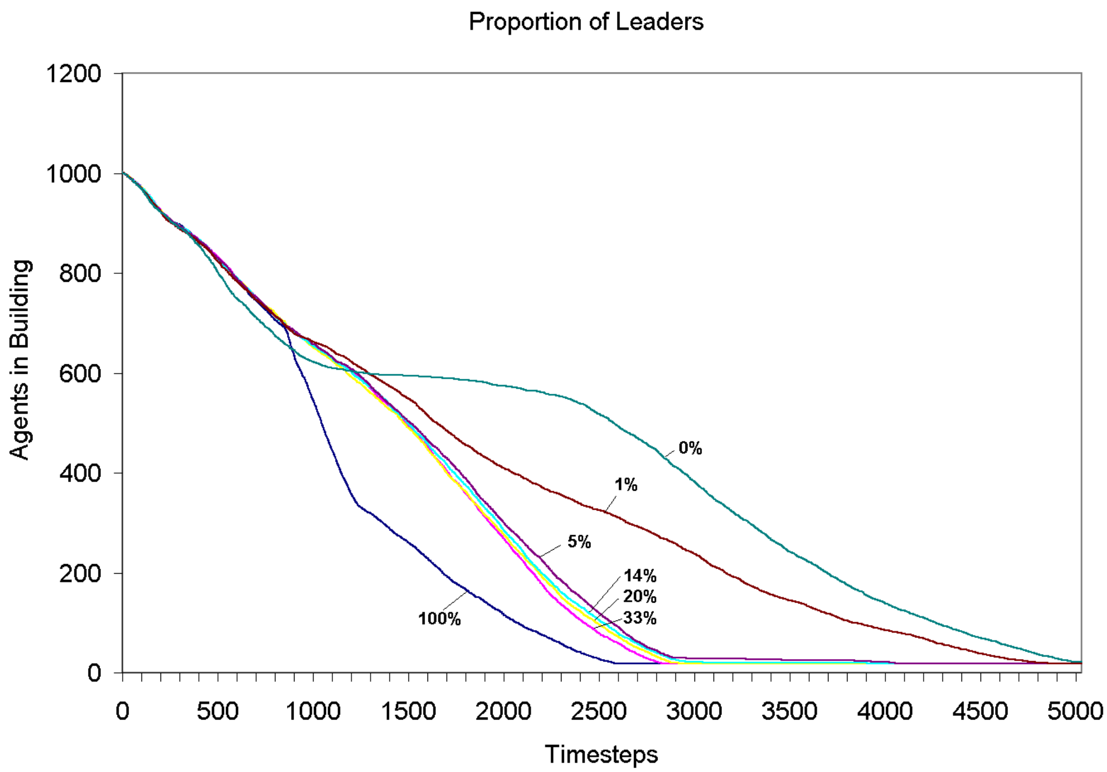

Figure 5.5: Analyzing how the proportion of leaders within the agent population speeds evacuation.

As expected, a greater proportion of leaders speeds the evacuation. However, since any proportion above 20% is infeasible for real evacuation scenarios, the 5% gives the most effective leader presence ratio. Indeed, having 33% leaders instead of 5% leaders gives little, if any, benefit.

5.3 Effect of Non-leader Communication

Communication between non-leaders greatly diminishes evacuation time. In both our simulation and in [PB06], evacuation times are halved. An agent that sees a blocked path will inform other nearby agents of his knowledge. Each time the information is spread, confidence in its accuracy and probability it will be trusted decreases. In these trials, we use a 1% leader-agent ratio.

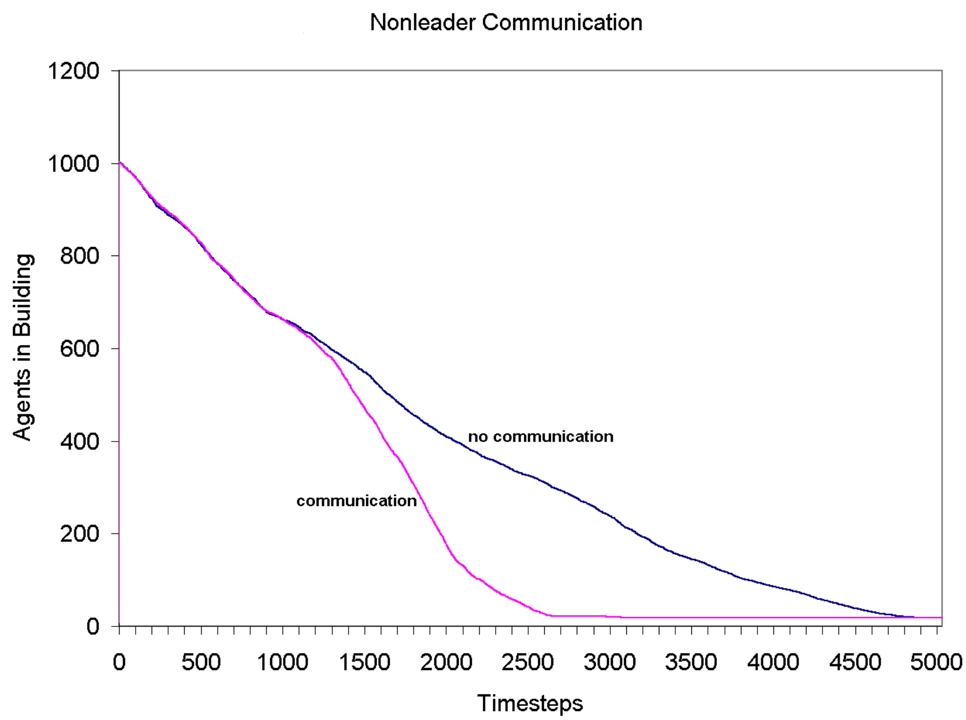

Figure 5.6: Communication between non-leader agents can halve the evacuation time.

Clearly, figure 5.6 demonstrates that communication between agents can speed the evacuation process considerably. However, this is not usually a factor a safety planner can control. Pedestrians in an evacuation are focused primarily on self-preservation, especially when panicked. Some ways to increase interagent cooperation may be to have leaders present to reduce panic levels and to have evacuation drills.

5.4 Effect of Leader Density Knowledge

Using radios, leaders could communicate information about crowd densities and movement throughout the building to each other. Using this information, leaders could plan evacuation routes more effectively to minimize exit time. While non-leader agents do have knowledge of crowds they can see, they do not know about crowd dynamics observed by all leader agents. In figure 5.7, leaders optimize crowd flow to reduce evacuation time by 10%. In these trials, there is a 1:5 leader to agent ratio.

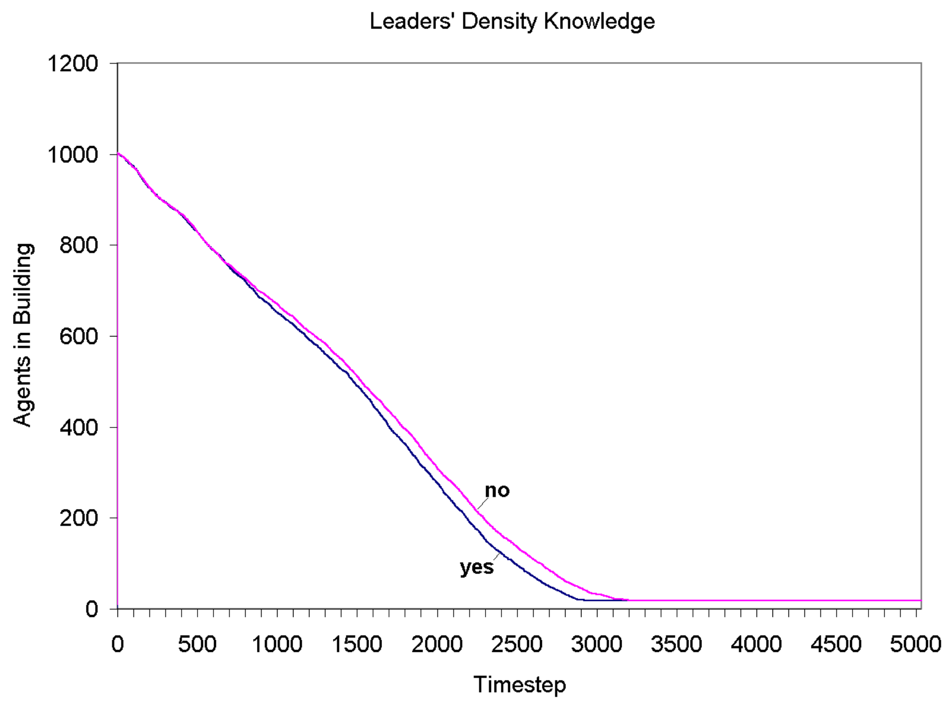

Figure 5.7: Knowledge of densities observed by all leaders helps maximize flow out of the building.

Theoretically, leaders should be able to communicate at this level of detail with the proper training. For example, they could refer to specific rooms and hallways with unique identifiers known to all agents. However, leaders do *not* have the ability to solve complex network-flow problems in their head to maximize flow out of the building. For this reason, our simulation does not rely on such algorithms; instead, each leader guides his current group of followers out along a path that approximately minimizes their exit time according to his knowledge. In a real evacuation, a leader that knows a certain hallway is very congested might guide the agents following him out along an alternate path.

5.5 Effect of Leader Behavior: Directing or Guiding

We experimented with two modes of leader behavior. In the first, leaders *lead* groups of followers out the building by walking along the exit path with them. In the other, they *guide* agents by informing them how to exit instead of leading them out. In the extreme case, all leaders remain in the building until all agents have left. In figure 5.8, these two types of behaviors are compared. In both cases, leaders move toward the exit at their own desired velocities (rather than remaining stationary). All of our simulations assign leaders' desired speed using the same distribution as is used for normal agents. The only variation in leader behavior is in what instructions they communicate to their followers. In these trials, 14% of agents are leaders.

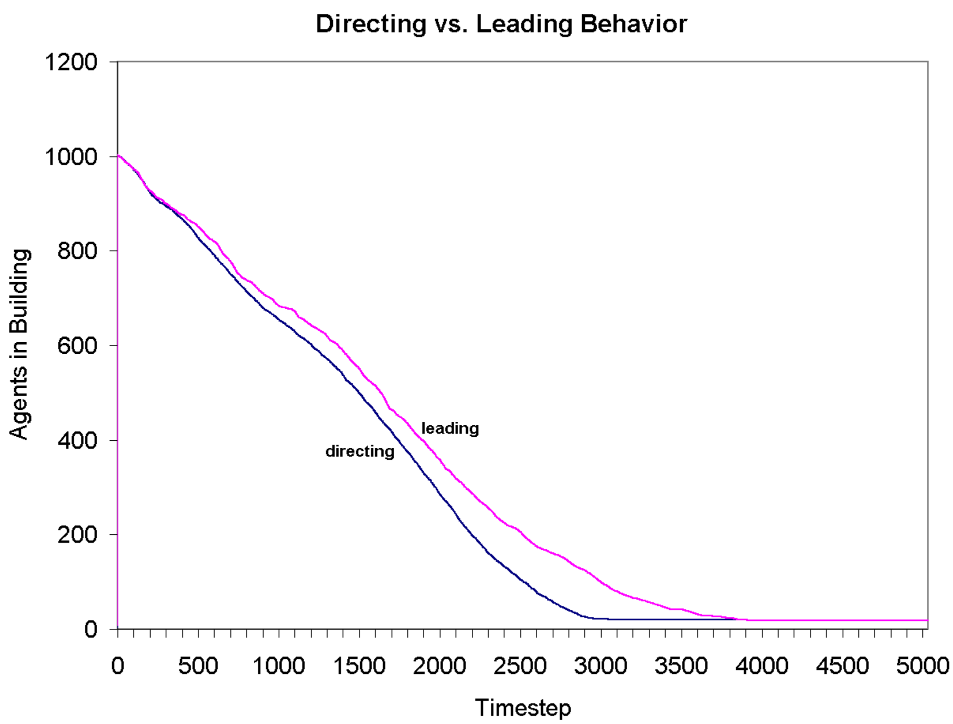

Figure 5.8: Leaders *guide* agents out of the building by giving directions. In this case, they also move at the same speed as the agents toward the exit. The *leading* behavior means agents follow them out of the building, often clustering around them and slowing the evacuation.

It appears that *guiding* rather than *leading* gives a much faster evacuation (almost 25% faster). The primary benefit of the first behavior is that agents do not crowd around leaders, thus preventing them from moving. Panic only exacerbates this phenomenon because agents pay even less attention to avoiding collisions other agents.

In trials where leaders were stationary until all agents had evacuated, the evacuation times were slower because leaders got in the way of crowds. One benefit of this extreme strategy is that no agents were left behind, looking for an exit. Helping people to realize the importance of staying in groups during drills may help eliminate these stragglers but may also increase congestion and unwillingness to follow detours. Perhaps a better strategy is for leaders to stay near walls to allow crowds to pass them.

Chapter 6

Conclusions

Our simulation emphasizes that the level of preparation for evacuations can have dramatic effects on their success. We found that training 5% of agents within the population to be leaders decreases evacuation time by up to 40% while minimizing panic and encouraging communication between all agents can decrease it by almost 50%. This discovered coincides with the OSHA guideline of a 1:20 leader to agent ratio. Leaders should have knowledge about hazards and crowd dynamics throughout the building. Sufficient training will allow leaders to quickly communicate this information between each other during an evacuation.

Leader behavior should depend on the population composition; if agents naturally travel in groups (for example, a large party a friends or closely-knit workplace), leaders can exit the building normally while guiding agents along optimal paths without worrying about stragglers. If not, agents should remain close to walls to avoid impeding crowd flow while they ensure all agents evacuate successfully. Choosing the optimal behavior could increase evacuation rate by another 11%.

Even though the simulation does account for individual differences like walking speed, patience, and knowledge, future work might model even more individual aspects like altruism between family members. These unique attributes could drastically affect an evacuation. Some work has already been done in this area by [LZC$^+$08]. In addition to testing simulations by observing emergent behavior, several researchers have performed Hidden Markov Model video analysis to compare their models to real crowd data. Such testing and validation is important for the field to show that simulations can predict the best way to organize and run evacuations. With that confidence, building planners and architects may use simulations to analyze and plan effective evacuation routes. In particular, future work might investigate typical office building or warehouse design to see if changes to the floorplan facilitate evacuations.

Bibliography

[Aue09] Stefan Auer. Realtime Particle-Based Fluid Simulation. *Technische Universität München*, 2009.

[BMdOB03] A. Braun, S.R. Musse, L.P.L. de Oliveira, and B.E.J. Bodmann. Modeling individual behaviors in crowd simulation. In *16th International Conference on Computer Animation and Social Agents*, pages 143 – 148, May 2003.

[BND11] Bellomo, Nicola, and Christian Dogbe. On the Modeling of Traffic and Crowds: A Survey of Models, Speculations, and Perspectives. *SIAM Review*, 53.3:409–463, 2011.

[Cha10] Panayiotis Charalambous. Simulating Virtual Crowds. *UCY Computer Graphics Lab: Lectures*, pages 1–38, 2010.

[CMS+11] A. Carmi, L. Mihaylova, F. Septier, S . K. Pang, and S. Godsill. MCMC-Based Tracking and Identification of Leaders in Groups. In *IEEE Workshop on Modeling, Simulation and Visual Analysis of Large Crowds (in conjunction with ICCV)*, Nov. 2011.

[Eva11] Evacmod.net – Evacuation Modeling Portal, October 2011.

[HFV00] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulating Dynamical Features of Escape Panic. *Nature*, 407:487–490, 2000.

[HFV02] Dirk Helbing, Illes Farkas, and Tamas Vicsek. Simulation of Pedestrian Crowds in Normal and Evacuation Situations. *Pedestrian and Evacuation Dynamics*, pages 21–58, 2002.

[Hu06] Xiaolin Hu. Context-Dependent Adaptability in Crowd Behavior Simulation. In *IEEE International Conference on Information Reuse and Integration*, pages 214–219, Sept. 2006.

[IEE11] IEEE. 1st IEEE Workshop on Modeling, Simulation and Visual Analysis of Large Crowds. In *Aggregate Dynamics for Dense Crowd Simulation.*, Barcelona, Spain, 2011.

[KZ11] Wee Lit Koh and Suiping Zhou. Modeling and simulation of pedestrian behaviors in crowded places. *ACM Trans. Model. Comput. Simul.*, 21:20:1–20:23, February 2011.

[LCHL07] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group Behavior from Video: a Data-driven Approach to Crowd Simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 109–118, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[LKF05] Taras I. Lakoba, D. J. Kaup, and Neal M. Finkelstein. Modifications of the Helbing-Molnair-Farkas-Vicsek Social Force Model for Pedestrian Evolution. *Simulation*, 81(5):339–352, 2005.

[LZC+08] Linbo Luo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, Feng Tian, Yongwei Wang, Xian Xiao, and Dan Chen. Agent-based Human Behavior Modeling for Crowd Simulation. *Computer Animation and Virtual Worlds*, 19(3–4):271–281, 2008.

[MOS09] R. Mehran, A. Oyama, and M. Shah. Abnormal Crowd Behavior Detection using Social Force Model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 935 – 942, June 2009.

[NGCL09] Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C. Lin. Aggregate Dynamics for Dense Crowd Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5):122:1–122:8, 2009.

[Occ80] Occupational Safety and Health Adminstration. Exit Routes, Emergency Action Plans, and Fire Prevention Plans: Means of Egress. *Occupational Safety and Health Standards*, 19(E.1910), 1980.

[PAB07] N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling Individual Agents in High-Density Crowd Simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 99–108, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[PB06] N. Pelechano and N.I. Badler. Modeling Crowd and Trained Leader Behavior during Building Evacuation. *Computer Graphics and Applications, IEEE*, 26(6):80 –86, Nov. – Dec. 2006.

[PPD07] Sébastien Paris, Julien Pettré, and Stéphane Donikian. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. *Computer Graphics Forum*, 26(3):665–674, 2007.

[RSLA11] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven Crowd Analysis in Videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[SGC04] Mankyu Sung, Michael Gleicher, and Stephen Chenney. Scalable Behaviors for Crowd Simulation. *Computer Graphics Forum*, 23(3):519–528, 2004.

[SVLS06] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. Crowd Simulation for Emergency Response using BDI Agent Based on Virtual Reality. In *Proceedings of the 38th Conference on Winter Simulation*, WSC '06, pages 545–553. Winter Simulation Conference, 2006.

[UT01] Branislav Ulicny and Daniel Thalmann. Crowd Simulation for Interactive Virtual Environments and VR Training Systems. In Nadia Magnenat-Thalmann, Daniel Thalmann, W. Hansmann, W. Purgathofer, and F. Sillion, editors, *Computer Animation and Simulation*, Eurographics, pages 163–170. Springer Vienna, 2001.