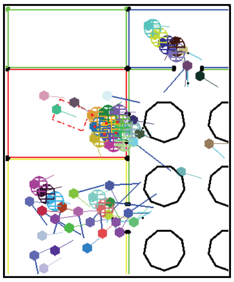
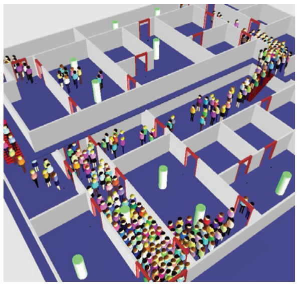
Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2007)
D. Metaxas and J. Popovic (Editors)

Controlling Individual Agents in High-Density Crowd Simulation





N. Pelechano, J.M. Allbeck and N.I. Badler

University of Pennsylvania, USA

Abstract

Simulating the motion of realistic, large, dense crowds of autonomous agents is still a challenge for the computer graphics community. Typical approaches either resemble particle simulations (where agents lack orientation controls) or are conservative in the range of human motion possible (agents lack psychological state and aren't allowed to 'push' each other). Our HiDAC system (for High-Density Autonomous Crowds) focuses on the problem of simulating the local motion and global wayfinding behaviors of crowds moving in a natural manner within dynamically changing virtual environments. By applying a combination of psychological and geometrical rules with a social and physical forces model, HiDAC exhibits a wide variety of emergent behaviors from agent line formation to pushing behavior and its consequences; relative to the current situation, personalities of the individuals and perceived social density.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1 Introduction

Animating motion for large crowds has been an important goal in the computer graphics, movie and video games communities. There has been a considerable effort on locomotion, path planning, navigation in large virtual environments, and realistic behavior simulation using cognitive models.

We classify crowd agent motions by three main approaches: *social forces* models, *rule based* models and *cellular automata* models. Although much effort has gone into improving the behavioral realism of each of these approaches, none of the current models can realistically animate *high-density* crowds. Social forces models tend to create simulations that look more like particle animation than human movement. Cellular automata models limit agent spatial movements and tend to expose the underlying checkerboard of cells when crowd density is high. Finally, rule based models either don't consider collision detection and repulsion at all or adopt very conservative approaches through the use of waiting rules, which work fine for low densities in everyday life simulation, but lack realism for high-density or panic situations.

Figure 1 shows a taxonomy for crowd simulation and compares our model (HiDAC: High-Density Autonomous Crowds) with the main models in the literature along the dimensions of animation realism and crowd density.

HiDAC addresses the problem of simulating high-density crowds of autonomous agents moving in a natural manner in dynamically changing virtual environments. Our solution to the problem of realistically simulating local motion under different situations and agent personalities uses psychological, physiological and geometrical rules combined with physical forces. Since applying the same

Symposium on Computer Animation 2007, San Diego, CA, August 03-04, 2007.
© 2007 ACM 978-1-59593-624-4/07/0008 $5.00

rules to all agents leads to homogeneous behavior, agents are given different psychological (e.g., impatience, panic, personality attributes) and physiological (e.g., locomotion, energy level) traits that trigger individual heterogeneous behaviors. Each agent is also endowed with perception and reacts to static and dynamic objects and other agents within the nearby space.

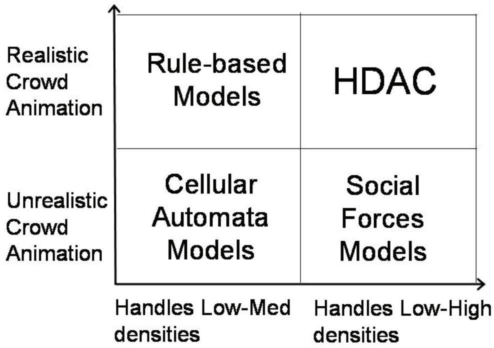


Figure 1: *Current models framework and our approach for low-level motion (HiDAC).*

Realistic movement may be defined as the emergence of crowd behaviors consistent with real observed crowds, and appropriate individual collision avoidance and collision response. We achieve such realism through contextual application of physical and geometric algorithms. Over longer distances tangential forces gently steer agents around obstacles, while over shorter distances collision response is applied to avoid overlapping. Pushing behavior between agents arises from varying the long/short personal space threshold of each individual. Agents in a hurry will not respect others' personal space and will appear to push their way through the crowd. In contrast, more 'polite' agents will respect lines and wait for others to move first.

Each agent has an influence disk (region) in front of them that triggers waiting behavior. Relaxed agents temporarily stop when another agent moves into their path, while impatient agents do not respond to this feedback and tend to 'push'. Our model stops impatient agents from appearing to 'vibrate' as they try to force their way through dense crowds: we add temporal stopping states to prevent the agent from trying to move during a short interval of time although it can still be pushed by others.

Our agents' behavior is determined by a high-level algorithm (including: navigation in complex virtual environments, learning, communicating and decision-making) [PB06], [POS05] and low-level motion controllers. Here we focus on a new approach for high-density crowds and so we will not explain in depth the algorithm for setting attractor points that drives high-level navigational behaviors.

The reminder of this paper begins with a review of related work. We then present an architectural overview of HiDAC. Section 4 describes the methods by which we combine a forces model with a set of psychological, physiological and geometrical rules to achieve realistic crowd movement. Finally, we present results and conclusions.

2 Related Work

Many crowd simulation methods derive from Helbing's empirical Social Forces model [HFV00] which applies repulsion and tangential forces to simulate interactions between people and obstacles, realistic 'pushing' behaviors and variable flow rates. The main disadvantage of this approach is that agents appear to 'shake' or 'vibrate' unnaturally in high-density crowds. There has been much work done using particle simulation approaches for low-density crowds. Particle systems and dynamics have been used for modeling the motion of groups with significant physics [BH97]. Individualism has been used to extend the social forces model [BMO*03]. Some recent work has focused on extending Helbing's model [HBJ*05], [LKF05] but has resulted in equations that are not applicable in real-time simulations. Crowd simulation systems have been described based on continuum dynamics instead of agent rules and run at interactive rates [TCP06].

Rule-based models [Rey87], [Rey99] achieve more realistic human movement for low and medium density crowds, but cannot handle contact between individuals and therefore fail to simulate 'pushing' behavior. These models usually adopt a conservative approach by avoiding contact and, when densities are high, applying 'wait' rules to enforce ordered crowd behavior without the need to calculate collision detection and response. Cognitive models have been used in combination with rule-based models to achieve more realistic behaviors for pedestrian simulation [ST05]. Different behavioral rules can be applied to the crowd, group or individuals to achieve more believable overall crowd behavior [OCV*02], [SBC*06], [TMK99].

Cellular-automata models [Che04], [KNN03], [TLC*01] are fast and simple to implement, but do not allow for contact between agents. Floor space is discretized and individuals can only move when the adjacent cell is free. Higher-level behaviors can be incorporated by pre-computing paths towards high-level goals and storing them within the grid [LMM03].

In order to navigate a complex environment, we need to have some high-level representation of the environment. Among the most popular techniques for crowd navigation are: cell and portal graphs [LCC06], [PB06], [PLT05], potential fields [Che04], and roadmaps [BLA02], [KSL*96], [SKG05]. Information can be embedded in the high-level representation of the virtual environment to achieve real-time crowd simulation [FBT99], [PB06], [TD00].

In multi-agent systems, each agent needs to sense the environment to perceive changes and react to them [TT94]. Perception is often simulated by casting a set of rays and finding their intersection with obstacles around the object [BNT94], [PHL05], [ST05]. Massive SW has also developed a crowd simulation system with vision-based behavior [MS05].

3 Architecture Overview

HiDAC is a multi-agent system without a centralized controller. Each agent has its own behavior based on personality variables that represent physiological and psychological factors observed in real people. Agent behaviors are computed at two levels:

- High-level behavior: navigation, learning, communication between agents, and decision-making. [PB06]
- Low-level motion: perception and a set of reactive behaviors for collision avoidance, detection and response in order to move within a bounded space.

Figure 2 shows the interaction between the two levels. The High-Level module receives information about bottlenecks and door changes that have been perceived by the agent and makes decisions based on that information and its current knowledge of the environment. Once the high-level decides the next room to walk to, it sends the next attractor point to the Low-Level module to carry out the required motion to reach it. When the Low-Level module reaches the attractor, it queries the High-Level module for the next attractor in its path towards the destination.

The Motion sub-module queries the Perception sub-module about positions and angles of obstacles, crowd density ahead of the agent, and velocity of dynamic obstacles. Based on information perceived and the internal state of the agent (current behavior, panic, impatience, etc.), the Motion sub-module calculates the velocity and next position of the agent, and sends a message to the Locomotion sub-module to execute the correct feet movements.

Both high-level and low-level agent behavior are affected by psychological and physiological attributes. The high-level is affected by changes in psychological state (panic or impatience), thus altering the decision-making process. Agent memory and orientation abilities are also affected by psychological states.

The low level is also affected by changes in the agent's psychological state which trigger modification of its speed, fall probability, pushing thresholds, etc. The psychological model needs to have as input information about environment events detected by the agent's perception system and information obtained through communication. This information is combined with the agent's current emotional state, possibly modifying it, to provide updated input to both low and high-level modules.

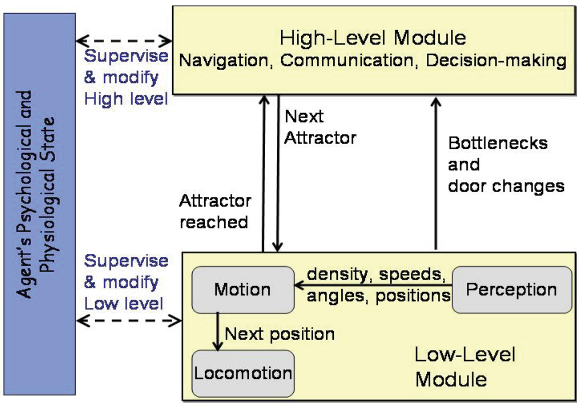


Figure 2: *Architecture Overview.*

4 Local Motion Approach

Local agent motion is based on a combination of geometrical information and psychological rules with a forces model to enable a wide variety of behaviors resembling those of real people. HiDAC uses psychological attributes (panic, impatience) and geometrical rules (distance, areas of influence, relative angles) to eliminate unrealistic artifacts and to allow new behaviors:

- Preventing agents from appearing to vibrate
- Creating natural bi-directional flow rates
- Queuing and other organized behavior
- Pushing through a crowd
- Agents falling and becoming obstacles
- Propagating panic
- Exhibiting impatience
- Reacting in real time to changes in the environment

4.1 The HiDAC Model

HiDAC is a parameterized social forces model that depends on psychological and geometrical rules. Its High-Level module determines which attractor point (waypoint or portal) an agent walks to within a room. [PB06]. Collision avoidance, detection and response are performed only with the people in the same room, and with static elements of that room (walls and obstacles). When people are crossing portals, care must be taken to avoid intersection between agents leaving and agents entering. HiDAC keeps track of the people currently crossing a portal, so that when an agent is near a door, collision detection is performed against agents in the room and agents crossing the doorway.

Collision detection and response must be performed with those agents that are overlapping the agent from any direction. In contrast, collision avoidance is only performed against individuals that appear in the desired direction of movement, and therefore are relevant to an agent's future position.

The movement of agent i ($\mathbf{F}_i^{To}$) depends on the desired attractor ($\mathbf{F}_i^{At}$), while avoiding walls w ($\mathbf{F}_{wi}^{Wa}$), obstacles k ($\mathbf{F}_{ki}^{Ob}$) and other agents j ($\mathbf{F}_{ji}^{Ot}$) and trying to keep its previous direction of movement to avoid abrupt changes in its trajectory ($\mathbf{F}_i^{To}[n-1]$). All these forces are summed together with different weights w_i that are the result of psychological and/or geometrical rules, and determine the importance of each force on the final desired direction of movement:

$$\mathbf{F}_i^{To}[n]=\mathbf{F}_i^{To}[n-1]+\mathbf{F}_i^{At}[n]w_i^{At}+\sum_w \mathbf{F}_{wi}^{Wa}[n]w_i^{Wa}+ +\sum_k \mathbf{F}_{ki}^{Ob}[n]w_i^{Ob}+\sum_{j(\neq i)} \mathbf{F}_{ji}^{Ot}[n]w_i^{Ot} \qquad (1)$$

The force vector is therefore:

$$\mathbf{f}_i^{To}=\frac{\mathbf{F}_i^{To}}{\left|\mathbf{F}_i^{To}\right|} \qquad (2)$$

And finally the new desired position $\mathbf{p}_i[n+1]$ for agent i is calculated as:

$$\mathbf{p}_i[n+1]=\mathbf{p}_i[n]+\alpha_i[n]v_i[n]\left(\left(1-\beta_i[n]\right)\mathbf{f}_i^{To}[n]+\beta_i[n]\mathbf{F}_i^{Fa}[n]\right)T+\mathbf{r}_i[n] \quad (3)$$

where:

- $v_i[n]$ is the magnitude of the velocity in the simulation step n. The velocity at each time step is calculated as:

$$v_i[n]=\begin{cases} v_i[n]=v_i[n-1]+aT & \textit{if } v_i[n]<v_i^{MAX} \\ v_i^{MAX} & \textit{otherwise}\end{cases}$$

where a is a constant that represents the acceleration of the agent when it starts walking until it reaches v_i^{MAX} .

- v_i^{MAX} is the agent's maximum walking velocity. It can be set to depend on agent capability (normal, handicapped) and modified dynamically if the agent enters panic mode or is injured.
- $\mathbf{r}_i$ is the result of the repulsion forces that affect the agent when it overlaps with a wall, obstacle or another agent; these will be introduced in section 4.1.2.
- α represents whether the agent will move in this step in its desired direction of movement or instead be pushed by a repulsion force.

$$\alpha_i=\begin{cases} 0 & \textit{if } |\mathbf{r}_i|>0 \vee \textit{StoppingRule} \vee \textit{WaitingRule} \\ 1 & \textit{otherwise}\end{cases}$$

The *StoppingRule* and *WaitingRule* are used to avoid shaking behavior and to allow for line formation, respectively. These rules will be explained in sections 4.1.3 and 4.1.4.

- β_i is used to give priority to avoiding fallen agents on the floor:

$$\beta_i=\begin{cases} 0.5 & \textit{if distance to fallen agent} < 2m \\ 0 & \textit{otherwise}\end{cases}$$

- $\mathbf{F}_i^{Fa}$ is the avoidance force to avoid fallen agents and will be explained in detail in section 4.1.6.
- T is the increment in time between simulation steps.

4.1.1 Avoidance Forces

Autonomous agents need to perceive the environment to avoid static and dynamic obstacles while walking to a attractor. HiDAC provides efficient perception through a cell and portal graph. Each cell corresponds to a room, and contains information about all the static objects within it. As the agents traverse the environment, the lists of dynamic objects within each room are rapidly updated; thus an agent can obtain obstacle data by querying the cell.

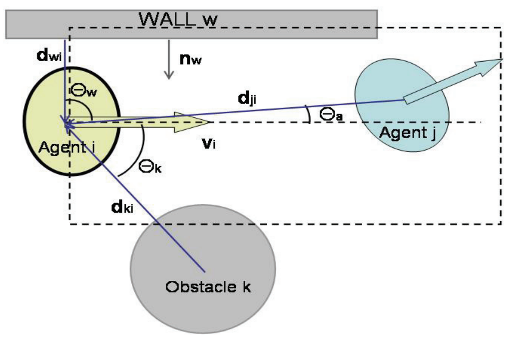


Figure 3: *Perception for the yellow agent.*

For each obstacle, wall and agent we need to calculate its distance to agent i and, if it is close enough, then we calculate the angle between agent i's desired direction and the line joining the center of agent i and the obstacle. This information is used to determine whether it falls within the rectangle of influence (Figure 3). The distance and the angle provide enough information to establish how relevant that obstacle is to the trajectory. As they navigate the environment, agents also update their perceived density of the crowd ahead which will be necessary to their decision-making process.

Wall and Obstacle Avoidance:

Avoidance forces are calculated only for relevant obstacles, walls and agents: those falling within the rectangle of influence.

The avoidance force for obstacle k is:

$$\mathbf{F}_{ki}^{Ob}=\frac{(\mathbf{d}_{ki}\times\mathbf{v}_i)\times\mathbf{d}_{ki}}{|(\mathbf{d}_{ki}\times\mathbf{v}_i)\times\mathbf{d}_{ki}|} \quad (4)$$

The avoidance force for wall w is:

$$\mathbf{F}_{wi}^{Wa}=\frac{(\mathbf{n}_{w}\times\mathbf{v}_i)\times\mathbf{n}_{w}}{|(\mathbf{n}_{w}\times\mathbf{v}_i)\times\mathbf{n}_{w}|} \quad (5)$$

Other Agent Avoidance: Overtaking and bi-directional flow:

To exhibit realistic counterflows and overtaking behaviors, we include rules that modify some parameters of the forces model. This approach allows us to simulate human behavior by setting parameters related to real human movement. The parameters that affect the tangential forces for obstacle avoidance are:

- Distance to obstacles
- Direction of other agents relative to agent i's desired velocity vector ($\mathbf{v}_i$).
- Density of the crowd

If an agent appears in the rectangle of influence, then tangential forces (described below) will be applied in order to slightly modify the direction of movement and make a curve in the trajectory to avoid collision.

The angle between two agents' velocity vectors determines whether their movements are confluent or opposed. This angle is also used to simulate human decision-making of how to react to an imminent collision. For example, if we are walking on the left side of a corridor, and another person walks towards us on our right, none of us would change direction, but if we are both walking in the middle of the corridor, the majority of people have a tendency to move towards their right side. Therefore, when the velocity vectors are almost collinear, the tangential forces will point to the right.

Suppose an agent i detects agent j and agent l as possible obstacles (Figure 4). We calculate the distance vector towards agent i for each of them ($\mathbf{d}_{ji}$ and $\mathbf{d}_{li}$). Agent j is farther away than l, but since it is moving against agent i, the perception algorithm establishes this obstacle as having higher priority. We select an agent to be avoided if it falls within the influence rectangle, unless that agent is walking in the opposite direction and with distance smaller than D_i-1.5, where D_i is the length of the rectangle.

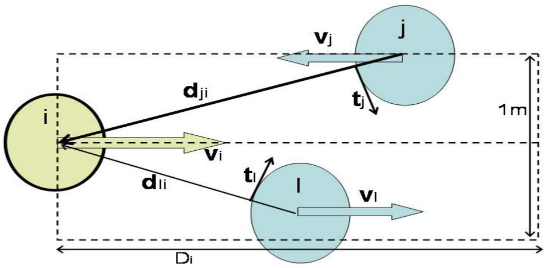


Figure 4: *Collision Avoidance rectangle of influence.*

The tangential force ($\mathbf{t}_j$) that will steer agent i to avoid j is:

$$\mathbf{t}_j = \frac{(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}}{\left|(\mathbf{d}_{ji} \times \mathbf{v}_i) \times \mathbf{d}_{ji}\right|} \qquad (6)$$

Next, the normalized tangential vector is multiplied by two scalar weights to obtain the final avoidance force

$$\mathbf{F}_{ji}^{Ot} = \mathbf{t}_j w_i^d w_i^o \qquad (7)$$

where w_i^d is the weight due to the distance between agents, and increases as the distance between the two agents becomes smaller and thus the agent i trajectory will change more abruptly as the distance to agent j decreases:

$$w_i^d = (d_{ji} - D_i)^2 \qquad (8)$$

and w_i^o is the weight due to the difference in orientation of the velocity vectors. It distinguishes whether the perceived agent is moving in the same direction as agent i or against it, and thus the magnitude will be higher to avoid counter flow.

$$w_i^o = \begin{cases} 1.2 & if(\mathbf{v}_i \cdot \mathbf{v}_j) > 0 \\ 2.4 & otherwise \end{cases} \qquad (9)$$

The last parameter to consider is the crowd density, which each agent perceives at any given time. If the crowd is very dispersed, then people look for avoidance from far away and keep their preference for the right hand side of the space (D_i=3m); but when the crowd is very dense, then the right preference is not so obvious and several bi-directional flows can emerge (D_i=1.5m). Modifying the length of the collision avoidance rectangle and reducing the angle for right preference based on perceived density achieves this behavior.

Figure 5 shows different bi-directional flow-rate formation for low and high densities. Figure 5b shows the result if the length of viewing rectangle and right preference parameters are not affected by density. The emergent behavior shows an unrealistic "triangle" of people moving in opposite directions, and awhile later in the simulation, two perfectly formed groups of people appear to move in opposite directions, which is less common in real high-density crowds.

HiDAC produces an interesting emergent counterflow behavior for high-density crowds (Figure 5c): the formation of lanes of people moving in the same direction intermingled among lanes moving in the opposite direction. This is a behavior that is often observed in real crowds, and it emerges here even though it is not explicitly implemented.

a)
b)
c)

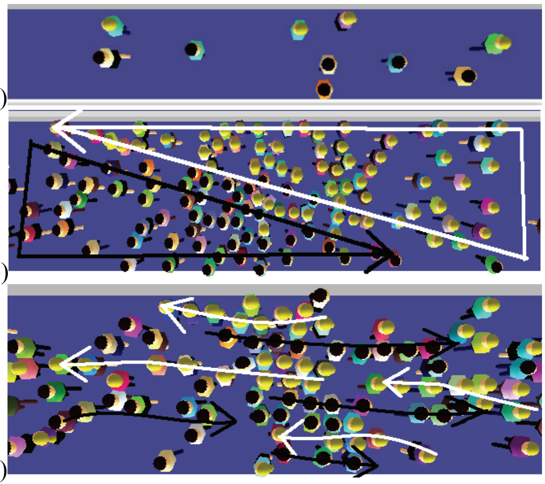


Figure 5: *Bi-directional flows. People with blonde hair walk towards the left, while dark-haired people walk towards the right. (a) low-density flows, (b) high-density without altering the viewing rectangle and right preference, (c) high-density with HiDAC.*

4.1.2 Repulsion Forces

When an agent's position overlaps with any static or dynamic obstacle, wall or agent then a collision response force applies. The repulsion force $\mathbf{r}_i$ from equation (3) is calculated as:

$$\mathbf{r}_i[n] = \sum_w \mathbf{F}_{wi}^{R\_Wa}[n] + \sum_k \mathbf{F}_{ki}^{R\_Ob}[n] + \lambda \sum_{j(\neq i)} \mathbf{F}_{ji}^{R\_Ot}[n] \qquad (10)$$

where $\mathbf{F}_{wi}^{R\_Wa}$ is the repulsion force from wall w, $\mathbf{F}_{ki}^{R\_Ob}$ is the repulsion force from obstacle k and $\mathbf{F}_{ji}^{R\_Ot}$ is the repulsion force from another agent j:

$$\mathbf{F}_{wi}^{R\_Wa}[n] = \frac{\mathbf{n}_w(r_i + \varepsilon_i - d_{wi}[n])}{d_{wi}[n]} \qquad (11)$$

$$\mathbf{F}_{ki}^{R\_Ob}[n] = \frac{(\mathbf{p}_i[n] - \mathbf{p}_k[n])(r_i + \varepsilon_i + r_k - d_{ki}[n])}{d_{ki}[n]} \qquad (12)$$

$$\mathbf{F}_{ji}^{R\_Ot}[n] = \frac{(\mathbf{p}_i[n] - \mathbf{p}_j[n])(r_i + \varepsilon_i + r_j - d_{ji}[n])}{d_{ji}[n]} \qquad (13)$$

where p_i is the position of agent i, p_j is the position of agent j and p_k is the position of obstacle k. Radii r_k, r_i, and r_j belong to obstacle k and agents i and j, respectively. Similarly, d_{ji} and d_{ki} are the distances between the centers of agent i and j, and the centers of agent i and obstacle k; d_{wi} is the shortest distance from the center of agent i to the wall w.

λ in equation 10 is used to set priorities between agents (that can be pushed) and walls or obstacles (that cannot be pushed). If there is repulsion from walls or obstacles, then λ is set to 0.3 to give preference to avoiding intersection with walls or obstacles over agents that can be pushed away.

Finally ε_i and ε_j are small personal space thresholds that the agents have and are used for the purpose of assigning different pushing abilities based on personality (discussed in section 4.1.5).

4.1.3. Solution to "shaking" problem in high-densities

When an agent encounters a bottleneck in a high-density crowd, applying a basic forces model leads to an unnatural behavior where agents appear to vibrate continuously. This behavior must be avoided (We have verified that this phenomenon is not based on our physics simulation implementation or its stepsize). In HiDAC we incorporate "stopping rules." These rules are applied based on the personality of the agent, direction of movement of other agents, and current situation (panic vs. normal).

When repulsion forces from other agents appear against the agent's desired direction of movement, and the agent is not in panic state, then the *stopping rule* applies:

$$\text{If}\left(\left(\mathbf{v}_j \cdot \mathbf{F}_i^{R\_Ot}[n]\right) < 0\right) \wedge (\neg panic) \text{then}$$

StoppingRule=TRUE

In order to avoid deadlocks, a timer is set to a random value within a small range, and when the timer reaches 0, the agent will set *StoppingRule=FALSE,* so that in the next simulation step the agent will try to move again

When *StoppingRule* is *true*, the parameter α_i in equation (3) is set to 0, which implies that the agent will only change position if it is pushed by other agents; otherwise it will inhibit the intention to move for several simulation steps. This effect drastically reduces the shaking behavior observed in the social force model without increasing the computational time of the algorithm.

Only forces directed backward are relevant (Figure 6).

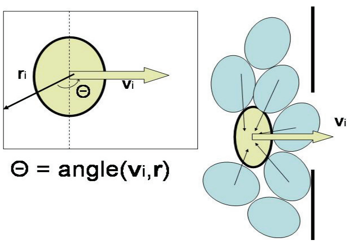


Figure 6: *Example of repulsion forces which are necessary to apply braking forces.*

If the forces appear to be towards our desired movement, we cannot decrease their intensity by not moving forward and therefore no reaction is necessary.

This method succeeds in reducing shaking behavior, while still allowing body contact and thus pushing behavior. Since stopping rules do not apply when the agent is being pushed forwards, this achieves the desired emergent result of people appearing to be pushed through doorways when there is a high-density crowd behind them.

4.1.4. Organized behavior - queuing

In a "normal" (non-panic) situation, people will respect lines and wait for others to walk first. Such organized behavior emerges by adding influence disks ahead of each agent that drive the temporal waiting behavior; they work similar to the *stopping rules*. Figure 7 shows the area that triggers waiting behaviors in a non-panicked agent *i* in a high-density crowd when another agent *j*, walking in the same direction, falls within the disk: agent *i* sets *WaitingRule=TRUE* and a timer starts. Agent *i* moves again when its area of influence does not satisfy the conditions for waiting, or when the timer reaches the value 0 to avoid deadlocks. The radius of the influence disk depends on personality (different people tend to respect different distances) and type of behavior desired: e.g., panicking agents will not respect these distances.

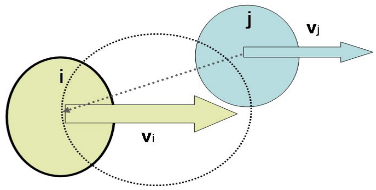


Figure 7: *Area of influence for waiting behaviors.*

For simulations of "normal" situations (e.g., individuals leaving a cinema after a movie) all the agents exhibit waiting behavior when there is no available space ahead of them. The emergent behavior observed corresponds to queuing. Since agents use tangential forces to move within a crowd while avoiding others, the strength of those tangential forces will lead to narrow or wide queues, as can be observed in Figure 8. The user can specify those tangential forces to be minimum, medium or maximum.

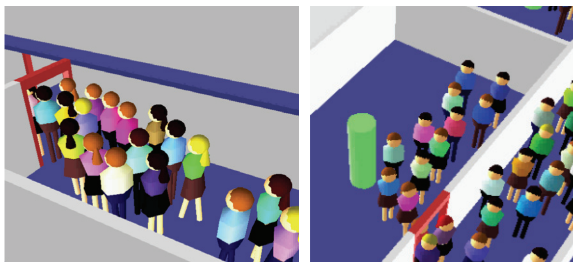


Figure 8: *Examples of wide and thin queues emerging when animating a "normal" scenario.*

4.1.5. Pushing behavior

Pushing behavior emerges because HiDAC can handle not only collision avoidance but also collision detection and response. Agents have different behaviors that can be triggered at any time. During an organized situation, individuals wait for space available before moving, but when in panic, they try to move until they collide with other individuals who impede forward progress. By combining both behaviors simultaneously for a heterogeneous crowd, we observe an emergent behavior where some individuals that do not respect personal space will get very close to other agents and push them away in order to open a path through a dense crowd. The effect of being pushed away is achieved by applying collision response forces and different personal space thresholds (ε_i and ε_j from the repulsion equations 11, 12 and 13).

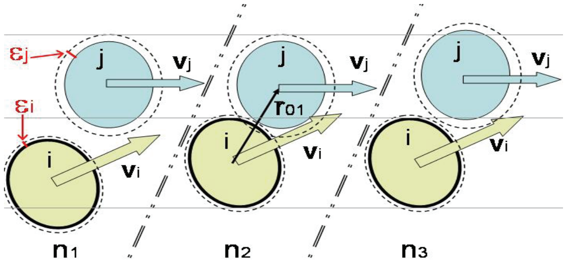


Figure 9: *Pushing forces.*

An agent suffers a repulsion force from another agent when its personal space is overlapped. Figure 9 shows a sequence of simulation steps, where a smaller personal space threshold ε_i allows agent *i* to get closer to agent *j* who has a larger personal space threshold ε_j. Thus agent *i* can push away agent *j* while agent *i* is not being pushed and can continue with its desired trajectory.

Figure 10 shows an example where the top left room has been filled with panicked people (represented by red-heads) who will tend to push others away, while the other three rooms contain individuals following more organized behaviors. After a few seconds of simulation, the red-headed people have managed to almost empty their room by pushing others away in the corridor in order to reach the exit faster. Individuals in the other rooms are calmly waiting for their turn to get through the door.

4.1.6. Falling and becoming obstacles

A benefit to a physical social force model is that one might use it to gauge potential injury arising from high-density situations. When the majority of pushing forces affecting one individual are approximately in the same direction, the agent will receive a sum of forces with magnitude high enough to make it lose equilibrium. At this moment the person may fall and become an obstacle for the rest of the crowd.

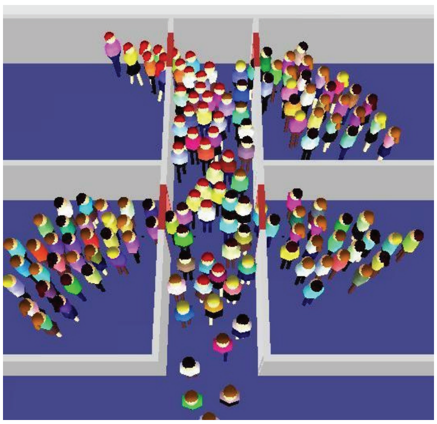


Figure 10: *Red-headed people exhibit panic behavior and push others to open their way through the crowd.*

Fallen agents represent a different type of obstacle because, unlike walls and columns, a body on the floor is an obstacle that should be avoided, but if necessary (or unavoidable) can be stepped over. In HiDAC, fallen individuals become a rectangular obstacle (a bounding box covering the torso and head, but not the legs since other individuals can easily step over that part of the agent). When other agents approach this new obstacle, weak tangential forces are applied in order to walk around the fallen agent ($\mathbf{F}_i^{Fa}$ in equation 3), but repulsive forces are not applied. Therefore, when the crowd is extremely dense and the pushing forces from behind are strong, the result is that agents may walk over the body on the floor, as has been observed in actual extreme situations. Figure 11 shows an example of this behavior (where the crowd density is artificially low for visibility).

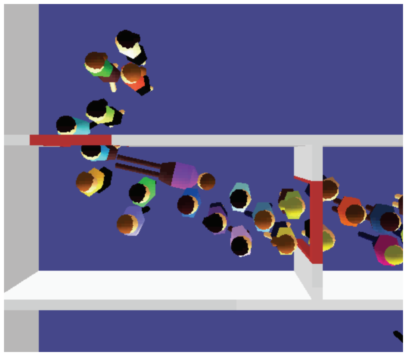


Figure 11: *Agents avoiding a fallen agent.*

4.1.7 Panic Propagation

HiDAC can simulate an emergency evacuation. When an alarm goes off some agents will start in the panic mode. While in panic they tend to move faster, push, and exhibit agitated behavior. All these behaviors depend on the agent personality and levels of panic. As the agents start running, they may provoke panic in other agents whose behavior will be modified in turn. To propagate panic, we use either communication between agents (managed by the High-Level behavior module), or perception to detect relevant changes in low-level behaviors, such as increasing crowd densities and number of people pushing or both.

4.1.8 Avoiding bottlenecks and interactive changes in the environment

When dealing with high-density crowds in buildings, bottlenecks can appear in the portals. HiDAC incorporates a high-level decision process that will allow impatient agents to react to this situation by finding an alternative path. As the low-level algorithm detects the bottleneck, it sends that information to the high-level which will try to find an alternative route based on what the agent can perceive from its current position (doors, obstacles) and the knowledge that the agent has about the internal connectivity of the building. If an alternative path is available, the high-level chooses a new portal as the goal and sets an attractor point to change the direction of movement.

Figure 12 shows a bottleneck and how impatient individuals (represented by the blonde people) have sought and walked toward an alternative door.

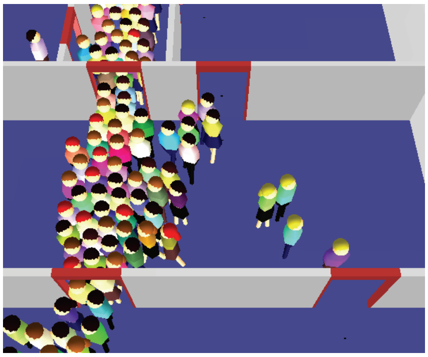


Figure 12: *Impatient people avoiding bottlenecks.*

When a change occurs in the environment (e.g., a door is blocked by fire) agents perceive and react to it. For an access change, the High-Level module needs to make a new wayfinding decision. The agent detects this change in real time and sets its destination to the new attractor set by the High-Level.

Figure 13 shows a sequence where dynamic wayfinding is forced by opening and closing doors, and agents must search for alternative paths. All low-level behaviors are still active during these activities.

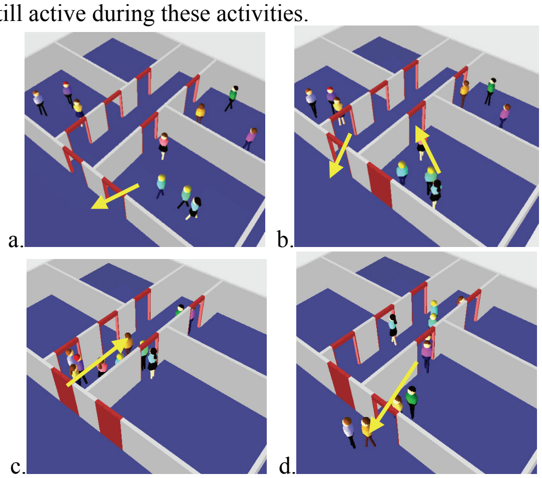


Figure 13: *Interaction with dynamic changes in the environment. Agents react to doors being closed and opened during the simulation.*

These examples show the interaction between High-Level and Low-Level modules to achieve realistic simulations with dynamic changes in the environment geometry.

Figure 14 shows the 2D and 3D view of a high density crowd. On the 2D view we can observe the red rectangle of influence for one of the agents (affects agent avoidance forces), the agents that affect the perceived density (with red points in the center of each agent), the avoidance forces with obstacles and walls in cyan, the avoidance forces with other agents in dark blue, and the stopping rules represented by circles of the same color as the agent waiting. The vector in the same color as the agent indicates the velocity direction.

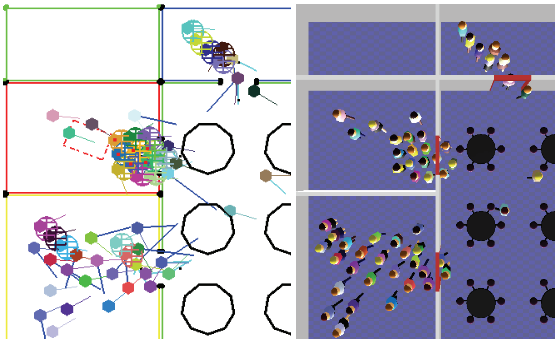


Figure 14: *2D and 3D view of a high density crowd.*

5 Results

We have presented a number of simulations that show HiDAC's visual output, and described methods for achieving many goals that enable realistic simulation of high-density crowds:

Goal	**Method**
Fast perception of environment	Influence rectangles, distances, angles and directions of movement are used to prioritize obstacles.
Eliminate shaking behavior	Apply stopping rules to forces model.
Natural bi-directional flow	Variable length influence rectangles and different 'right' preferences.
Queuing behavior	Influence discs triggering waiting behavior based on agent direction.
Pushing behavior	Collision response based on variable 'personal space thresholds'.
Falling agents becoming new obstacles	Apply tangential forces for obstacle avoidance but not repulsion forces.
Panic propagation	Modify agent behavior based on personality and perception of other agents' level of panic.
Crowd impatience	Dynamically modifying route selection based on environmental changes.

We have run simulation tests on a 2.99 GHz Intel Xeon with 2GB of RAM measuring frame rates both for simulation only and for simulation and 3D rendering. When doing only simulation, HiDAC can handle up to 1800 agents with a frame rate of 25Hz. Simulation and 3D rendering using an NVIDIA Quadro FX 3400/4400 graphics system can achieve 25 frames/second (not using GPU rendering) for up to 600 simple 3D virtual human figures ("crayon figures") each with about 100 vertices. For the frame rate tests, we used a large complex environment with 85 rooms and 53,448 vertices overall.

6 Conclusions

HiDAC can be tuned to simulate different types of crowds, ranging from extreme panic situations (fire evacuation) to high-density crowds under calm conditions (leaving a cinema after a movie). Also we allow for heterogeneous crowds where a number of different behaviors can be exhibited simultaneously.

Unlike cellular automata and rule-based models, HiDAC can realistically simulate an individual trying to force its way through a crowd by pushing others, and unlike social forces models, our agents can exhibit more respectful behavior when desired and make decisions in terms of letting others walk first and queuing when necessary. These emergent behaviors are driven by the combination of psychological and physiological rules together with a social forces model. "Impatience" has been integrated in order to avoid the sheep-like behavior that many crowd simulation models exhibit.

We have shown novel extensions to social forces models by adding stopping rules and influence region controls that mitigate agent vibration while not increasing computational time. Our system uses the best features of both rule-based and social forces systems, while eliminating their disadvantages. The implementation allows real-time simulations for hundreds of individualized agents.

The social force model extensions also mitigate combinatorial problems associated with the possible geometric arrangements of large numbers of agents. Rather than analyze all possible spatial configurations or force agents into discrete cells, HiDAC uses general behaviors based on surrounding social forces and crowd density perception to limit influences and consequences to a small number of nearby agents.

We have expanded on our previously reported work that included higher level concepts such as leadership, agent communication, levels of environmental knowledge, and way finding, by greatly improving lower level agent interactions, such as reduced vibrations, natural bi-directional flows, queuing behavior, pushing, falling, panic propagation, impatience, and real time reactions to changes in the environment, thereby increasing the heterogeneity of the crowd.

While HiDAC's combination of a social forces model, rule-based model, and unique extensions enable a variety of behaviors, achieving these behaviors for different scenarios (e.g. a typical mall scene versus a building evacuation) requires a user to set a few low level parameters. Though these parameters are limited in number and would not be overwhelming to a user, they currently would require the user to understand some of the lower level methodologies of HiDAC in order to achieve the desired behaviors. We are currently working on mapping these parameters to agent properties that would enable a user to create a crowd simulation based on the properties of individuals in the crowd instead of lower level parameters. In addition to expanding this personality model, we are also working to add in agent actions other than locomotion. Working toward an integrated model of crowd behavior given an individual's personality, changing physiological state, and personal goals and values is imminently feasible [POS05]. Because the model already uses some psychological, physiological, and social factors, the simulation can use this dynamic information to further select and animate specific agent task actions with the environment and interactions with other agents.

6 Acknowledgements

This research was partially supported by the National Science Foundation grant IIS-0200983, Office of Naval Research Virtual Technologies and Environments grant N0001 4-04-1-0259, Army Research Office grant N61339-05-C-0081, and a Fulbright scholarship. We also thank Autodesk and Cal3D for their software and Jeff Wajcs and everyone in HMS for their assistance. Special thanks to Joan Pelechano for his rigorous review of our physics equations.

References

[BLA02] BAYAZIT, O.B., LIEN, J.-M. and AMATO, N.M., Roadmap-based flocking for complex environments. in *Proceedings of Pacific Conference on Computer Graphics and Applications*, (2002), 104-113.

[BNT94] BOULIC, R., NOSER, H. and THALMANN, D., Automatic derivation of curved human walking trajectories from synthetic vision. in *Proceedings of Computer Animation*, (1994), IEEE Computer Society Press, 93-103.

[BMO*03] BRAUN, A., MUSSE, S.R., OLIVEIRA, L.P.L. and BODMANN, B.E.J., Modeling Individual Behaviors in Crowd Simulation. in *Proceedings of Computer Animation and Social Agents*, (2003), 143-148.

[BH97] BROGAN, D. and HODGINS, J. Group Behaviors for Systems with Significant Dynamics. *Autonomous Robots*, *4*, 1997. 137-153.

[Che04] CHENNEY, S., Flow Tiles. in *ACM SIGGRAPH/ Eurographics Proceedings of Symposium on Computer Animation*, (2004), 233-242.

[FBT99] FARENC, N., BOULIC, R. and THALMANN, D., An informed environment dedicated to the simulation of virtual humans in urban context. in *Proceedings of Eurographics*, (1999), 309-318.

[HBJ*05] HELBING, BUZNA, JOHANSSON and WERNER Self-Organized Pedestrian Crowd Dynamics. *Transportation Science*, *39* (1), 2005. 1-24.

[HFV00] HELBING, D., FARKAS, I. and VICSEK, T. Simulating dynamical features of escape panic. *Nature*, *407*, 2000. 487-490.

[KSL*96] KAVRAKI, L., SVESTKA, P., LATOMBE, J. and OVERMARS, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transaction on Robotics and Automation*, *12* (4), 1996. 566-580.

[KNN03] KIRCHNER, A., NAMAZI, A., NISHINARI, K. and SCHADSCHNEIDER, A., Role of Conflicts in the Floor Field Cellular Automaton Model for Pedestrian Dynamics. in *2nd International Conference on Pedestrians and Evacuation Dynamics*, (2003), 51-62.

[LKF05] LAKOBA, KAUP and FINKELSTEIN Modification of the Helbing-Molnar-Farkas-Vicsek Social Force Model for Pedestrian Evolution. *Simulation*, *81* (5), 2005. 339-352.

[LCC06] LERNER, A., CHRYSANTHOU, Y. and COHEN-OR, D., Efficient cells-and-portals partitioning. in *Computer Animation & Virtual Worlds*, (2006), Wiley, 21-40.

[LMM03] LOSCOS, C., MARCHAL, D. and MEYER, A. Intuitive crowd behaviour in dense urban environments using local laws. *IEEE Theory and Practice of Computer Graphics*, 2003. 122.

[MS05] MASSIVE_SOFTWARE_INC. 3D animation system for crowd-related visual effects, http://www.massivesoftware.com, 2005.

[OCV*02] O'SULLIVAN, C., CASSELL, J., VILHJALMSSON, H., DOBBYN, S., PETERS, C., LEESON, W., GIANG, T. and DINGLIANA, J., Crowd and Group Simulation with Levels of Detail for Geometry, Motion and Behavior. in *Third Irish Workshop on Computer Graphics*, (2002).

[PHL05] PAN, X., HAN, C.S. and LAW, K.H., A Multi-agent Based Simulation Framework for the Study of Human and Social Behavior in Egress Analysis. in *The International Conference on Computing in Civil Engineering*, (Cancun, 2005).

[PB06] PELECHANO, N. and BADLER, N. Modeling Crowd and Trained Leader Behavior during Building Evacuation. *IEEE Computer Graphics and Applications*, *26* (6), 2006. 80-86.

[POS05] PELECHANO, N., O'BRIEN, K., SILVERMAN, B. and BADLER, N., Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. in *First International Workshop on Crowd Simulation. (V-CROWDS '05)*, (2005).

[PLT05] PETTRE, J., LAUMOND, J.-P. and THALMANN, D., A Navigation Graph for real-time crowd animation on multilayered and uneven terrain. in *First International Workshop on Crowd Simulation*, (2005), 81-90.

[Rey87] REYNOLDS, C., Flocks, herds, and schools: A distributed behavior model. in *Proceedings of ACM SIGGRAPH*, (1987), 25-34.

[Rey99] REYNOLDS, C., Steering behaviors for autonomous characters. in *Game Developers Conference*, (1999), 763-782.

[ST05] SHAO, W. and TERZOPOULOS, D., Autonomous pedestrians. in *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, (2005), 19-28.

[SBC*06] SILVERMAN, B., BHARATHY, G., CORNWELL, J. and O'BRIEN, K. Human Behavior Models for Agents in Simulators and Games: Part II - Gamebots for a Foreign Culture. *Presence*, *15* (2), 2006. 163-185.

[SKG05] SUNG, M., KOVAR, L. and GLEICHER, M., Fast and accurate goal-directed motion synthesis for crowds. in *Symposium on Computer Animation*, (2005), 291-300.

[TLC*01] TECCHIA, F., LOSCOS, C., CONROY, R. and CHRYSANTHOU, Y., Agent behavior simulator (ABS): A platform for urban behavior development. in *Proceedings of ACM/EG Games Technology Conference*, (2001).

[TMK99] THALMANN, D., MUSSE, S.R. and KALLMANN, M., Virtual Humans' Behavior: Individuals, Groups, and Crowds. in *Proceedings of Digital Media Futures*, (1999), 13-15.

[TD00] THOMAS, G. and DONIKIAN, S., Virtual Humans Animation in Informed Urban Environments. in *Proceedings of Computer Animation*, (2000), 112.

[TCP06] TREUILLE, A., COOPER, S. and POPIVIC, Z., Continuum Crowds. in *ACM Transactions on Graphics (SIGGRAPH 2006)*, (2006), 1160-1168.

[TT94] TU, X. and TERZOPOULOS, D., Artificial Fishes: Physics, Locomotion, Perception, Behavior. in *Proceedings of ACM SIGGRAPH*, (1994), 43-50.