Continuum Crowds

Adrien Treuille[1] Seth Cooper[1] Zoran Popović[1,2]

[1]University of Washington [2]Electronic Arts

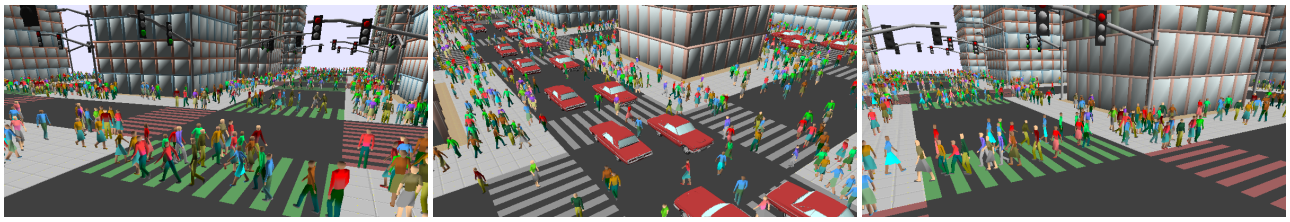


Figure 1: A variety of city scenarios simulated with our model.

Abstract

We present a real-time crowd model based on continuum dynamics. In our model, a dynamic potential field simultaneously integrates global navigation with moving obstacles such as other people, efficiently solving for the motion of large crowds without the need for explicit collision avoidance. Simulations created with our system run at interactive rates, demonstrate smooth flow under a variety of conditions, and naturally exhibit emergent phenomena that have been observed in real crowds.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Crowds, Human Simulation, Motion Planning

1 Introduction

Human crowds are ubiquitous in the real world, making their simulation a necessity for realistic interactive environments. Physically correct crowd models also have applications outside of computer graphics in psychology, transportation research, and architecture. In this paper we focus on real-time synthesis of crowd motion for thousands of individuals with intersecting paths. Our formulation is designed for large groups with common goals, not for scenarios where each person's intention is distinctly different.

Real-time crowd simulation is difficult because large groups of people exhibit behavior of enormous complexity and subtlety. A crowd model must not only include individual human motion and environmental constraints such as boundaries, but also address a bewildering array of dynamic interactions between people. Further, the model must reflect intelligent path planning through this changing environment. Humans constantly adjust their paths to reflect congestion and other dynamic factors. Even dense crowds are characterized by surprisingly few collisions or sudden changes in individual motion. It has proven difficult to capture these effects in simulation, especially for large crowds in real-time.

Virtually all previous work has been *agent-based*, meaning that motion is computed separately for each individual. The agent-based approach is attractive for several reasons. For one, real crowds clearly operate with each individual making independent decisions. Such models can capture each person's unique situation: visibility, proximity of other pedestrians, and other local factors. In addition, different simulation parameters may be defined for each crowd member, yielding complex heterogeneous motion. However, the agent-based approach also has drawbacks. It is difficult to develop behavioral rules that consistently produce realistic motion. Global path planning for each agent quickly becomes computationally expensive, particularly in real-time contexts. As a result, most agent models separate local collision avoidance from global path planning, and conflicts inevitably arise between these two competing goals. Moreover, local path planning often results in myopic, less realistic crowd behavior. These problems tend to be exacerbated in areas of high congestion or rapidly changing environments.

This paper presents a real-time motion synthesis model for large crowds *without agent-based dynamics*. We view motion as a per-particle energy minimization, and adopt a continuum perspective on the system. This formulation yields a set of dynamic potential and velocity fields over the domain that guide all individual motion simultaneously. Our approach unifies global path planning and local collision avoidance into a single optimization framework. People in our model do not experience a discrete regime change in the presence of other people. Instead, they perform global planning to avoid both obstacles and other people. Our dynamic potential field formulation also guarantees that paths are optimal for the current environment state, so people never get stuck in local minima.

We note that global path planning is frequently an unrealistic assumption. People often have limited vision and only partial knowledge of the terrain. Global knowledge is only an approximation to more accurate long-term planning with limited visibility and knowledge. Still, we found that the global planning assumption produces significantly smoother and more realistic crowd motion than the common and equally unrealistic assumption of strictly local knowledge. This is especially the case when people are threading through dense congestion. At the same time, our framework is not as general as agent-based methods: we trade-off individual variability for real-time planning of optimal crowd behavior with minimal computation per individual. To achieve the benefits of both models, agents can seamlessly be integrated with continuum crowds.

Contributions. This paper presents a new *type* of crowd simulator driven by dynamic potential fields which integrate both global navigation and local collision avoidance into one framework. Our

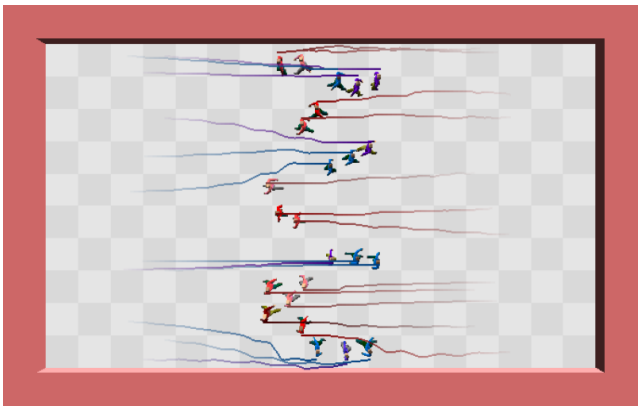
system uses two very simple ideas which are crucial to creating the effects we show: one is a velocity-dependent term which induces lane formation, and the other is a distance-based term which stabilizes the flow. We also show how individuals can integrate knowledge of the future to produce more intelligent behavior. The computational cost of our algorithm depends on the number of grid cells used to compute the dynamic potential. At coarse discretizations we can simulate crowds of over ten thousand people at several frames a second, while at fine discretizations we can simulate high resolution dynamics for several thousand people at real-time frame rates. We also show that our model exhibits important emergent phenomena observed in real crowds. In particular, people walking in opposite directions tend to form lanes, and group crossings can form vortices.

2 Related Work

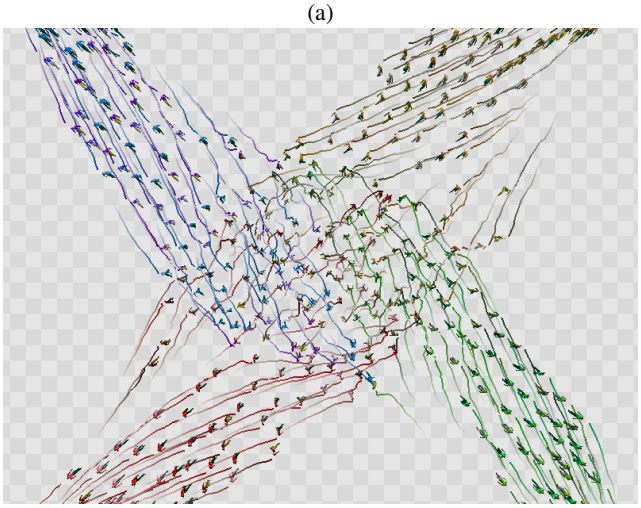
The most natural way to model locomotion of large human crowds is with agent-based methods in which each person plans individually. This is primarily because people have distinct characteristics and make decisions based on personal goals. Such dynamics have reached perhaps their most sophisticated form in the work of Funge *et al.* [1999], which models not only behavioral dynamics such as stimulus response but also cognitive aspects such as knowledge and learning. This model was further expanded with pedestrian visibility and path planning in [Shao and Terzopoulos 2005]. Massive Software, founded by Stephen Regelous, publishes a production-quality crowd simulator [Massive Software 2006] that gives the animator the freedom to author details describing each individual agent within the crowd. Though each of the above mentioned systems could be used to create ultra-realistic crowds, considerable expertise is still required to design the "correct" model for each individual. It has recently been observed that no existing methods can cover a comprehensive range of scenarios with accuracy suitable for safety engineering purposes [Still 2000]. More importantly, in order to accurately model human reasoning and congestion planning, the computation cost for each individual is large. For this reason the most accurate methods are impractical for real-time simulations of large crowds, which is the main focus of this paper.

To improve on the performance characteristics of agent-based methods, and to avoid the complexities of constructing a cognitive model for each agent, researchers have explored a number of simplifications to crowd models, including local methods, precomputed static path plans, and global path planning on a coarse environment graph. Locally controlled agents can be traced back to the seminal work of Reynolds [1987] who demonstrated that emergent flocking behavior can be generated from simple local rules. Reynolds's technique yields visually compelling flocks of birds and fish. Since this work, a considerable corpus of research has emerged focused on modeling human crowds with locally controlled agents. Reynolds [1999] has expanded the set of possible behaviors. Other work accounts for sociological issues [Musse and Thalmann 1997], psychological effects [Pelechano et al. 2005], geographically-based direction [Sung et al. 2004], social forces [Helbing et al. 1994; Cordeiro et al. 2005], and nontrivial motion dynamics [Brogan and Hodgins 1997], to name just a few.

Unlike real people, locally controlled agents do not perform any global or mid-range planning to avoid non-local congestion or cross-flowing pedestrians. Collisions are prevented locally by changing movement when other people get sufficiently close. Many interesting collision avoidance methods have been proposed including geometric models [Feurtey 2000], grid-based rules [Loscos et al. 2003], density-dependent techniques [Musse et al. 1998], behavioral models [Tu and Terzopoulos 1994], particle force interaction models [Heïgeas et al. 2003], and Bayesian decision processes [Metoyer and Hodgins 2004]. Ultimately, local collision avoidance



(a)



(b)

Figure 2: (a) Two groups pass, naturally forming lanes. (b) A vortex forms as four groups cross.

can only go so far in capturing the inherently long-term planning process that humans employ to smoothly avoid dynamic congestion. Because people continuously plan ahead, real pedestrians exhibit virtually no near-misses or "visual bumps" as they navigate through congestion. In this paper, we try to preserve this dynamic planning behavior.

Collision avoidance alone cannot model real crowds where people have an overall goal or objective. Consequently, many local crowd models have combined collision avoidance with global navigation. In general, global planning has taken the form of graph-based techniques or static potential fields. Lamarche and Donikian [2004] used sophisticated topological precomputations to enable real-time per-agent global path planning which takes into account visibility. They employed a separate collision avoidance process, so their agents were susceptible to "getting stuck" due to local minima under congestion. Bayazit *et al.* [2002] created visually pleasing results using coarse graph-based roadmaps for global planning together with collision avoidance. Probabilistic maps and decoupled planners were also combined with motion graphs and constraints to compute detailed locomotion of pedestrians [Sung et al. 2005]. Pettré [2005] uses navigation graphs of static scenes to find pedestrian paths in real-time, but does not account for collisions or congestion. Kamphuis and Overmars [2004] extended path planning so that agents could stay together. Li *et al.* [2001] assumed a leader/follower model; they reduced the expense of global planning by only planning for the leader at the cost of a less general system. Goldenstein *et al.* [2001] introduced a sophisticated three-

layer technique to handle local and global navigation, using a static potential field for maneuvering around fixed obstacles. Precomputed, static potential functions have also been used to study evacuation [Kirchner and Schadschneider 2002]. In general, static potential fields do not handle changing environments, such as changing traffic lights and exit congestion.

Agent-based crowd modeling has also been studied in transportation research for urban planning, path design, and evacuation [Helbing et al. 2003]. Helbing's social force model has been influential in this field, and has proven capable of reproducing specific crowd phenomena [Helbing et al. 2001]. A number of commercially available high-fidelity crowd models are available: the EGRESS system [Ketchell 2002] employs a cell-based approach to simulating macro behavior of crowds; Crowd Dynamics [Still 2000] uses a calibrated agent-based framework to accurately model a number of observed crowd phenomena.

Multiple-agent path planning has also been studied extensively in robotics, primarily for cooperative tasks of multiple robots. "Centralized" planners choreograph the motion of all agents through spacetime [Li and Chou 2003]. These methods are exponential in the number of robots, and are not appropriate for crowds, where agents optimize for personal goals. Closer to our work are "decoupled" planners where agents plan individually. Such algorithms require priority schemes to fix conflicting plans [Bennewitz and Burgard 2001]. Parker [1993] explores the trade-offs between local and global planning. Potential fields for navigation have also been explored in robotics, as by Arkin [1987].

Finally, an alternative approach to crowd synthesis has emerged from the fluid dynamics community. Hughes [2003] has developed a model which represents pedestrians as a continuous density field and presents a pair of elegant partial differential equations describing the crowd dynamics. Most importantly, this system is driven by an evolving potential function, defined so as to guide the density field optimally toward its goal. A full derivation can be found in [Hughes 2002], and in a subsequent application to a medieval battle analysis [Clements and Hughes 2004]. The 1D version of this model was also confirmed with real crowd data [Hongwan et al. 2003]. Alternative continuum models of 1D crowd flow have also been analyzed [Colombo and Rosini 2005]. In the graphics community, Chenney's flow tiles [2004] demonstrated that desirable crowd properties such as congestion avoidance can be achieved with divergence-free flows, although they do not address many of the other important aspects of crowds.

Our model was directly inspired by Hughes's work: we use a similar potential function to guide pedestrians towards their goal. We also adopt his technique of combining pedestrians into groups, and the use of "discomfort fields" to handle geographical preferences. However, Hughes only investigates analytic properties of his equations and does not discuss simulation. We address this by transforming Hughes's continuous crowd field into a particle representation which is better suited to rendering and forms the basis for our stable, efficient simulator. The particle representation is fully integrated throughout the dynamics, from the theoretical underpinnings to our grid discretization scheme, which is tailored for the difficult case of fine grid resolutions relative to the size of the people. We have also made numerous improvements to Hughes's model itself. Notably, we take into account crowd flow when computing individual speed. This allows our system to exhibit a number of visually interesting and empirically-confirmed phenomena such as lane formation when people cross in contrary directions. Also, we have changed the definition of path optimality to include a distance term, which inhibits the oscillations we observed in Hughes's dynamics.

3 The Governing Equations

In this section we develop a mathematical model of crowd dynamics. We begin with a set of observations about crowd flow, with each observation culminating in a precise hypothesis. Our model is directly derived from these hypotheses. We subsequently discuss the role of potential functions in our model, and present the exact form of the speed equation. We defer the discretization and simulation of these equations to Section 4.

The overarching force driving crowd flow is that people have a destination, or *goal*. Goals can be specific such as *go to 1549 35th street*, or general as in *go to the west side of town*. They can also be dynamic such as *chase this person*, *find a non-empty theater seat*, or *explore unseen parts of the environment*. Crowd scenarios without a definable goal such as browsing at the mall, or wandering aimlessly are not appropriate for the continuum crowd formulation. We consider goal selection to be an external parameter set by the animator.

HYPOTHESIS 1. Each person is trying to reach a geographic goal $G \subseteq \mathbf{R}^2$.

As people move towards their goal, the next most important consideration is their speed. In general, we assume that people move at the maximum speed possible given environmental conditions. For example, a city-dweller might walk at a brisk pace, while a medieval army might charge ahead at a fast run. The environment affects the speed in a number of ways: descending a slope causes people to increase their speed, while physical boundaries are impassible. Most importantly, the presence of other people affects speed. We assume people have difficulty moving against the current which is proportional to the local crowd density. In the extreme case, two people cannot intersect.

HYPOTHESIS 2. People move at the maximum speed possible. This can be expressed as a *maximum speed field* f such that a person at location $\mathbf{x}$ moving in direction θ has velocity:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \theta)\mathbf{n}_\theta, \tag{1}$$

where throughout this paper, $\mathbf{n}_\theta = [\cos\theta, \sin\theta]^T$ will denote the unit vector pointing in direction θ.

Even when people can move unobstructed, they may express preferences for certain paths. Pedestrians, for example, often do not cross a street until they reach a crosswalk. Also, people generally follow trodden paths when they exist, even if they yield longer routes. We represent this idea as follows.

HYPOTHESIS 3. There exists a *discomfort field* g so that, all things being equal, people would prefer to be at point $\mathbf{x}$ rather than $\mathbf{x}'$ if $g(\mathbf{x}') > g(\mathbf{x})$.

As with goal selection, the discomfort field is exogenous to our model. However, in Section 3.3, we discuss how dynamic discomfort can be used to enhance collision avoidance between people and other moving obstacles such as cars.

We now tie together the above hypotheses and describe how people choose paths. In general, people choose the minimum distance path to their destination. However, this preference is tempered by a desire to avoid congestion and other time-consuming situations. This can be seen as the classic trade-off between energy and time minimization. Additionally, people prefer to minimize their exposure to areas of high "discomfort." We summarize these ideas by assuming that people choose paths so as to minimize a linear combination of the following three terms:

- The length of the path.
- The amount of time to the destination.
- The discomfort felt, per unit time, along the path.

HYPOTHESIS 4. Let Π be the set of all paths from $\mathbf{x}$ to some point in the goal. Assuming that the speed field f, discomfort g, and goal G are fixed, a person at location $\mathbf{x}$ will pick the path $P \in \Pi$ minimizing

$$\underbrace{\alpha \int_P 1 ds}_{\text{Path Length}} + \underbrace{\beta \int_P 1 dt}_{\text{Time}} + \underbrace{\gamma \int_P g dt}_{\text{Discomfort}} . \tag{2}$$

Here α, β, and γ are weights for individual terms; ds means that the integral is taken with respect to path length while dt means the integral is taken with respect to time. These two variables are related by $ds = f dt$ where f is the speed. Using this equality, we may rewrite Equation (2) as

$$\alpha \int_P 1 ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds \tag{3}$$

which can be simplified to

$$\int_P C ds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f} \tag{4}$$

is the *unit cost field*.

3.1 Optimal Path Computation

We now show how potential functions can be used to find optimal paths given the path cost described in Equation (4). Suppose we have a function $\phi : \mathbf{R}^2 \to \mathbf{R}$ everywhere equal to the cost of the optimal path to the goal. Intuitively, it makes sense that for any person, the optimal strategy is to move opposite the gradient of this function, as this will decrease cost of the path most rapidly. Conversely, this function can be constructed by following the set of all optimal paths outwards from the goal, integrating cost along the way. Indeed, the *potential function* ϕ can be defined exactly this way: in the goal $\phi = 0$, and everywhere else ϕ satisfies the *eikonal equation*:

$$||\nabla \phi(\mathbf{x})|| = C, \tag{5}$$

where the unit cost C is evaluated in the direction of the gradient $\nabla\phi$. A theorem from the calculus of variations (see, for example [Kimmel and Sethian 2001]) guarantees that *all* optimal paths follow exactly the gradient of this function. In our system, this means that every person moves in the direction opposite the gradient, scaled by the speed at that point:

$$\dot{\mathbf{x}} = -f(\mathbf{x}, \theta) \, \frac{\nabla \phi(\mathbf{x})}{||\nabla \phi(\mathbf{x})||}, \tag{6}$$

where $\dot{\mathbf{x}}$ denotes the velocity, and $f(\mathbf{x}, \theta)$ is evaluated in the direction of motion.

Calculating a potential field may seem like a cumbersome way of finding an optimal path. We can, however, make a simplifying assumption that justifies this method. Suppose a group of people all share an identical speed field, discomfort, and goal. This is often the case in crowds when a number of people are trying to get to the same location at approximately the same speed. In this case, we need to calculate the potential function for the group only once, deriving optimal paths for all group members simultaneously. In reality, of course, people move at different speeds, have varied perceptions of discomfort, and seek different goals. Therefore, we divide the crowd into a set of groups, each with different characteristics. At each timestep we construct a potential function ϕ for each group and then move the people in that group according to Equation (6). Since each group affects the speed f and discomfort g across all groups, the entire crowd motion is coupled.

Solving Equation (5) is the slowest aspect of simulation. Thus, we prefer to have as few groups as possible. However, as we show in our results, even one group exhibits interesting dynamics, and realistic crowd congestion phenomena can be attained with few groups.

3.2 Speed

The speed field f measures the maximum permissible speed of movement for every point and every direction in the domain. Our speed model is ad-hoc, but very simple to compute, and it provides the physically plausible behavior for the crowd. We begin with an intuitive description. Speed is a density-dependent variable. At low densities, speed is dominated by the terrain, remaining constant on flat surfaces, but changing with the slope. At higher crowd densities, the speed becomes dominated by the movement of nearby people: movement is inhibited when trying to move against the flow, but is unaffected when moving with the flow. Modeling this latter effect is one important difference between our model and that of Hughes [2003]. Moreover, we found that this velocity-dependent term is crucial for modeling "lane formation" where people moving in opposite directions along a path naturally segregate into separate lanes, a phenomenon that has been widely noted in real crowds.

Since speed is density-dependent, we begin by describing the *crowd density field* ρ. We convert each person into an individual density field, denoted ρ_i for the ith person. This field should peak at the location of person i, and fall off radially. The specific form of this function is unimportant as long that it is no less than some threshold value $\bar{\rho}$ within a bounding disc of radius r, and no greater outside. The crowd density ρ is simply the sum of each individual density field. As we compute this, we simultaneously calculate the *average velocity field* $\bar{\mathbf{v}}$ which scales each person's density by his velocity, indicating the overall speed and direction of crowd flow:

$$\rho = \sum_i \rho_i, \quad \text{and} \quad \bar{\mathbf{v}} = \frac{\sum_i \rho_i \dot{\mathbf{x}}_i}{\rho}. \tag{7}$$

Here $\dot{\mathbf{x}}_i$ denotes the velocity of the ith person. Both sums are taken across all people in all groups.

We now describe specifically how density affects the speed. In areas of very low density ($\rho \leq \rho_{min}$ for some ρ_{min}), the speed f is equal to the *topographical speed* f_T. Assuming the terrain is bounded to lie within the minimum and maximum slopes s_{min} and s_{max}, the speed varies inversely with the slope:

$$f_T(\mathbf{x}, \theta) = f_{max} + \left(\frac{\nabla h(\mathbf{x}) \cdot \mathbf{n}_\theta - s_{min}}{s_{max} - s_{min}} \right) (f_{min} - f_{max}), \tag{8}$$

where $\nabla h(\mathbf{x}) \cdot \mathbf{n}_\theta$ is the slope of the height field h in direction θ.

In areas of high density ($\rho \geq \rho_{max}$ for some ρ_{max}), the speed f is equal to the *flow speed* $f_{\bar{\mathbf{v}}}$:

$$f_{\bar{\mathbf{v}}}(\mathbf{x}, \theta) = \bar{\mathbf{v}}(\mathbf{x} + r\mathbf{n}_\theta) \cdot \mathbf{n}_\theta. \tag{9}$$

The flow speed $f_{\bar{\mathbf{v}}}$ is essentially the average velocity $\bar{\mathbf{v}}$ evaluated at a distance r from the location $\mathbf{x}$. The offset causes people to evaluate the average velocity for the area *into* which they are trying to move. Indeed, if not for the offset, a person's speed would be dominated by their own previous speed, an undesirable effect. Also, the flow speed is clamped to be nonnegative, implying that the crowd can slow people down, but never carry them backwards.

At medium densities ($\rho_{min} < \rho < \rho_{max}$), we linearly interpolate between the topographical and flow speeds:

$$f(\mathbf{x}, \theta) = f_T(\mathbf{x}, \theta) + \left(\frac{\rho(\mathbf{x} + r\mathbf{n}_\theta) - \rho_{min}}{\rho_{max} - \rho_{min}} \right) (f_{\bar{\mathbf{v}}}(\mathbf{x}, \theta) - f_T(\mathbf{x}, \theta)) . \tag{10}$$

Note that, like the flow speed, the density ρ is evaluated at an offset of $r\mathbf{n}_\theta$, again for the same reason: we do not want a person's own contribution to the density field to self-obstruct their motion. Since we assumed that a person's contribution to the density field ρ is no greater than $\bar{\rho}$ outside the disc of radius r, we can be sure that in low congestion their speed will always be equal to the topographical speed so long as $\rho_{min} \geq \bar{\rho}$.

3.3 Models of the Future

The model presented in previous sections can be further improved by accounting for two important aspects of crowd motion that involve predictive models of the future:

Predictive discomfort. Individuals in our crowd simulator account for the future by path planning through a constantly updated static view of the environment. That is, the model takes into account moving obstacles, but not that the obstacles are moving. Often this simplification is justified: real pedestrians avoid those walking in the opposite direction and intelligently plan around areas of high congestion. In this model however, when two people cross perpendicularly, they fail to anticipate one another. We address this by adding a small amount of discomfort in front of each person, causing others to avoid this region. Specifically, we advance each person's position by their velocity for several timesteps depositing to the discomfort field an amount proportional to the density that would be calculated were the person there. In simulations with many intersecting paths, predictive discomfort yields smoother flow, effectively incorporating the salient aspects of short-horizon dynamic planning, while staying within our framework of fast, per-timestep, 2D planning. In section 5, we extend this idea to collision avoidance with other moving objects such as cars.

Expected periodic field changes. A similar issue arises when the field deterministically changes over time. Consider an environment with two exit doors rapidly opening and closing. An exiting crowd would continuously switch direction back and forth towards the currently open door. Similarly, when a traffic light turns red the speed field must be temporarily set to zero to prevent crossing. Individuals in our basic model would naively conclude that crossing is impossible, and pick another path. Here we offer a different solution. When computing the potential, we replace the actual inverse speed with the expected inverse speed computed over a longer time period. For traffic lights, the expected inverse speed is computed from the percentage of time the traffic light is green during a period. Similar computations are performed for other periodic changes (e.g. opening doors). By the linearity of expectations, the optimal path cost given in equation (2) can then be interpreted as the minimizing the *expected* time to the goal.

4 Implementation

To simulate our system, the model described in the last section must be discretized in time and space. The simulator advances through each timestep as follows:

For each timestep:
- Convert the crowd to a density field.
- *For each group*:
  - Construct the unit cost field C.
  - Construct the potential ϕ and it's gradient $\nabla\phi$.
  - Update the people's locations.
- Enforce the minimum distance between people.

To compute these fields, we discretize space into a regular grid, with physical variables defined at various locations within each grid cell. We store all physical fields as 2D arrays of floating point numbers according to the schema shown in Figure 4(a). All scalar fields are defined at the center of each grid cell. This is also true of the average velocity $\bar{\mathbf{v}}$, which is stored as a pair of floats. All anisotropic fields–those depending on both position and direction–are stored with four floats per cell corresponding to $\theta = \{0°, 90°, 180°, 270°\}$, that is the east, north, west, and south faces of each cell. Finally, the velocity $\mathbf{v}$, the gradient of the height ∇h and of the potential $\nabla\phi$ are stored at the faces of each cell, in a MAC-style arrangement (see [Fedkiw et al. 2001]). We now look more closely at how every step of the simulator is implemented.

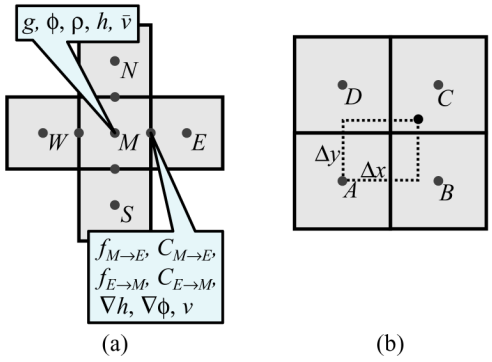


Figure 4: Discretized grid structure.

4.1 Density Conversion

We "splat" the crowd particles onto a density grid in order to compute the speed field, which is density dependent. We have two requirements of the density conversion function. First, the density field must be continuous with respect to the location of the people. Otherwise movement would cause sharp discontinuities in the density, and subsequently in the speed. Second, each person should contribute no less than $\bar{\rho}$ to their own grid cell, but no more than $\bar{\rho}$ to any neighboring grid cell. Intuitively, this requirements ensures that each individual is not affected by its own contribution to the density field. This requirement is the discrete analog to the disc of radius r surrounding each person described in Section 3.2.

The first requirement is standard, and could be satisfied by any number of splatting techniques, including bilinear and Gaussian. The second requirement is unusual, and we have defined a new density conversion technique to satisfy it. For each person, we find the closest cell center whose coordinates are both less than that of the person. We then compute the relative coordinates $[\Delta x, \Delta y]$ of that person with respect to the cell center, as shown in Figure 4(b). The person's density is then added to the grid as

$$\rho_A = \min(1-\Delta x, 1-\Delta y)^\lambda \qquad \rho_B = \min(\Delta x, 1-\Delta y)^\lambda$$
$$\rho_C = \min(\Delta x, \Delta y)^\lambda \qquad \rho_D = \min(1-\Delta x, \Delta y)^\lambda,$$

where the *density exponent* λ determines the speed of density falloff. This density conversion method is continuous with respect to the location of each person, and is defined so that each person contributes at least $\bar{\rho}$ to their grid cell, but no more than $\bar{\rho}$ to neighboring cells, with $\bar{\rho} = 1/2^\lambda$. Thus, our requirements are satisfied. As we compute the density field ρ, we simultaneously compute the average velocity $\bar{\mathbf{v}}$ according to Equation (7).

4.2 Unit Cost

There are two steps to computing the unit cost field C. We first compute the speed field f according to Equation (10), then calculate the cost field C using Equation (4). These fields are anisotropic so computing them involves not only iterating over each grid cell, but also iterating over each of the four directions within each cell. For example, for cell M in Figure 4(a) we must compute $f_{M\to i}$ and

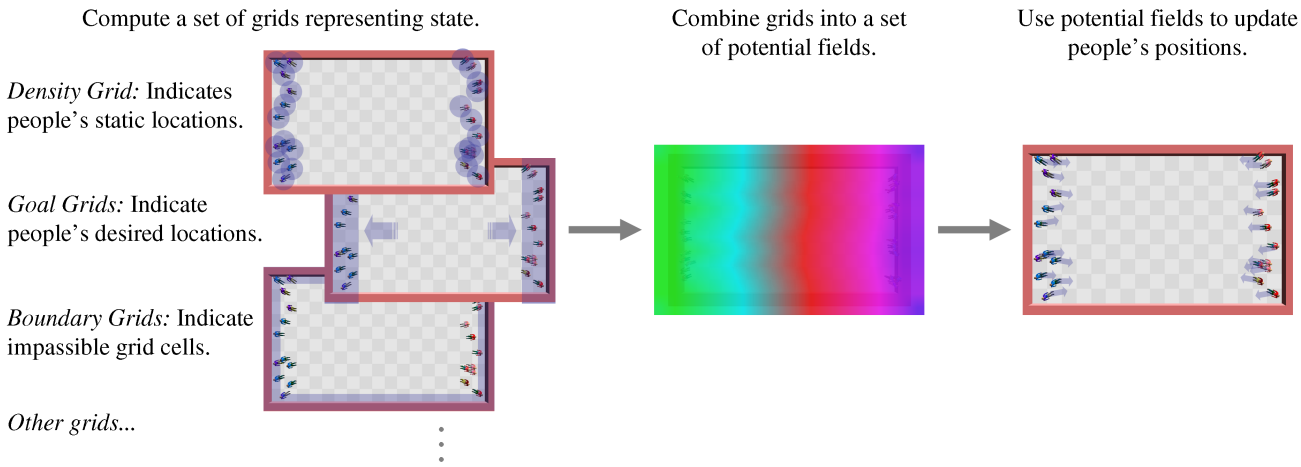


Figure 3: General algorithm overview.

$C_{M\to i}$ for $i \in \{E,N,W,S\}$. So that the pedestrians anticipate obstructions ahead, we evaluate the speed and discomfort at the cell *into which* the person would be moving if they chose that direction. For example, in the case of $f_{M\to E}$, we would plug the density ρ_E, discomfort g_E, and average velocity $\bar{\mathbf{v}}_E$ into Equation (10).

4.3 Dynamic Potential Field Construction

Constructing the dynamic potential is the most complex and time consuming step of the algorithm. Equation (5) defines the potential as an implicit eikonal equation, and hence cannot be calculated directly. However, efficient methods have been developed to solve this type of equation, notably the fast marching method [Tsitsiklis 1995] and the fast sweeping method [Tsai et al. 2005]. We have chosen the former, since the latter becomes inefficient when optimal paths must follow circuitous routes (because of obstacles, for example).

The fast marching algorithm is by now well known so we describe our implementation only briefly and refer the reader to [Tsitsiklis 1995] for more details. We begin by assigning the potential field ϕ the value of 0 inside the goal, and including these grid cells in the list of KNOWN cells; all other cells are UNKNOWN and set to ∞. Those UNKNOWN cells adjacent to KNOWN cells are included in the list of CANDIDATE cells and we approximate ϕ at these locations by solving a finite difference approximation to Equation (5). The CANDIDATE cell with the lowest potential is then included in the KNOWN cells, and its neighbors are introduced into the CANDIDATE set by re-approximating the potential at these cells. This process is repeated, propagating the KNOWN cells outwards from the goal until all cells are defined. A heap data structure efficiently handles the list of candidate grid cells and gives the algorithm its $O(N \log N)$ running time, where N is the number of grid cells.

We now describe the finite difference approximation to equation (5). Suppose we were solving the equation for grid cell M in Figure 4(a). We first find the less costly adjacent grid cell along the both x- and y-axes:

$$m_x = \operatorname*{argmin}_{i\in\{W,E\}} \{\phi_i + C_{M\to i}\} \qquad m_y = \operatorname*{argmin}_{i\in\{N,S\}} \{\phi_i + C_{M\to i}\}.$$

We then use these upwind directions to calculate a finite difference approximation to Equation (5) by solving for the larger solution to ϕ_M in the quadratic equation

$$\left(\frac{\phi_M - \phi_{m_x}}{C_{M\to m_x}}\right)^2 + \left(\frac{\phi_M - \phi_{m_y}}{C_{M\to m_y}}\right)^2 = 1. \qquad (11)$$

If either m_x or m_y is undefined because both neighbors have infinite cost, then we drop that dimension out of Equation (11). Note that, up to the discretization of the angles, this formulation correctly handles the anisotropy of the unit cost field C. Once we have computed ϕ_M, we take its difference with the neighboring grid cells in the upwind direction giving us $\nabla\phi$. We then renormalize the gradient, and multiply by the speed in the appropriate directions to compute the velocity field $\mathbf{v}$ at that point.

4.4 Crowd Advection

Having calculated the potential field ϕ, its gradient $\nabla\phi$, and determined the velocity field $\mathbf{v}$, we then simply update each person's position by interpolating the velocity field. Each person's position is displaced by their velocity, effectively computing an Euler integration of Equation (6). We have experimented with higher order Runge-Kutta integrators, but they do not appreciably affect the dynamics.

4.5 Minimum Distance Enforcement

In theory, our model ensures that no two people will intersect. Two people approaching one another will eventually experience density so high that the average velocity term dominates the speed equation and their speed towards each other will drop to zero. In practice, however, we can only resolve the dynamics up to the resolution of the grid, and two people in the same grid cell will sometimes intersect. This happens with relative infrequency as people naturally avoid one another in our system. However, when intersections do happen, they cause visually unpleasant artifacts.

To address this, we enforce a pair-wise minimum distance between the people. We simply iterate over all pairs within a threshold distance, symmetrically pushing them apart so that the minimum distance is enforced. This procedure does not strictly ensure that minimum distances are preserved, as modifying the position of one person changes that person's distance to all other people. Also, minimum distance enforcement occasionally introduces artifacts, such as when non-moving people are pushed by others. However, the vast majority of the time we found this method useful for eliminating intersection artifacts which could not be resolved by the grid. Note that minimum distance enforcement has linear time complexity if the crowd is first "binned" into a high-resolution neighbor grid.

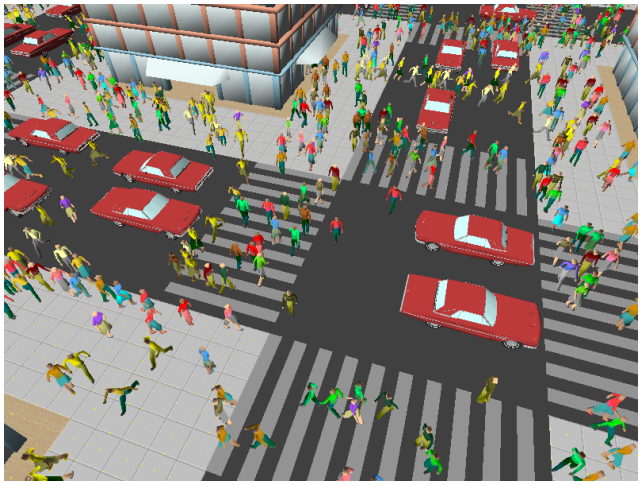


Figure 5: People in yellow shirts evacuate a building.

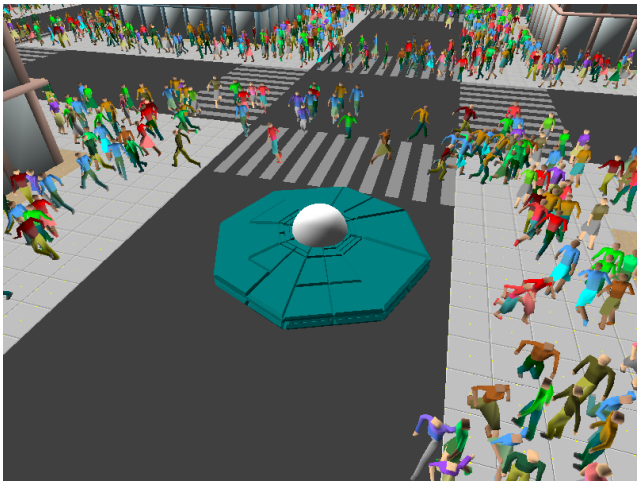


Figure 6: A continuum crowd reacts to a user-driven flying saucer agent in real-time.

5 Results

We have run a diverse set of crowd simulations with our system and found that the continuum method can produce smooth behavior for thousands of agents at interactive rates. We encourage the reader to view the animations in the companion video. Our implementation was in a mixture of Python and C++. All simulations ran on a 3.4GHz Pentium with a NVidia Quadro FX 3400 graphics card. Simulation updates took between 2 and 5 frames per second (fps). This running time includes a parallel rendering thread which calculated "in-between" frames at 24 fps (12 fps for our 10,000 person example). Because our simulations usually exhibited smooth motion, the decoupling of the renderer from simulator was not apparent, yielding real-time, interactive simulations. The most noticeable motion artifacts in our simulations were due to our simplified human animation. We used basic locomotion cycles with precomputed in-between poses to avoid popping artifacts between cycles. A more sophisticated off-line motion model would produce higher quality results, but would preclude interactive animation.

Figure 2(a) shows a simulation of twenty-four people crossing in a hallway and forming lanes to avoid collision. This phenomenon is ubiquitous in real crowds, and it can be observed in all of our simulations. In Figure 2(b), we show a simulation of four groups crossing one another to reach opposite corners of the domain. A vortex forms as the groups cross paths. This emergent behavior is an exotic form of lane formation, and it also serves to prevent collisions. Such vortices have been observed in real crowds [Helbing et al. 2003]. In these simulations, the checkered pattern on the floor indicates the grid size.

We have also run comparisons with two agent-based models which can be seen in the companion video. These models are the flocking model of Reynolds [OpenSteer 2006], and the particle-forces model of Heïgeas [2003]. We can see that in local models, people do not plan early for congestion, and thus do not avoid it until they are near. If we increase the local collision avoidance range, pedestrians avoid congestion earlier, but also tend to spread out, taking significantly longer paths towards their goal. With continuum crowds, people plan to avoid congestion in advance and the overall movement assumes a smoother, more optimal flow. We note that it is certainly possible for local models to display vortex-like crowd flow as reported by Heïgeas, but such emergent features occur only for specific parameters of the model and only after the simulation has reached a steady state. It is also possible to mix continuum crowds with agents, as shown in Figure 7 where a continuum crowd interacts with Reynolds agents.

One of the features of our system is that very large scale simulations are possible on coarse grids. To demonstrate this, we simulated a 2000 person army retreating from an 8001 person army on a 60×60 grid (Figure 8). Despite the coarse resolution, interesting effects are apparent in which people change paths as congestion conditions evolve across the mountainous terrain. This simulation ran at 5 fps before rendering "in-between" frames.

We also simulated a school quad as classes get out. People exit from eight doorways and are randomly assigned a different destination door. This simulation was challenging because people crossed paths at many different angles. For this reason, we used added ten timesteps of predictive discomfort, and the result is a smooth flow, even under the rapidly changing conditions.

Finally, we created a 16 square block city environment with crosswalks, streets, and sidewalks (Figures 1, 5, and 6). Our simulations contained between one and two thousand people on this 120×120 grid. By adding discomfort to the streets, people preferred to stay on the crosswalks, crossing the streets only when congestion was too great. We were also able to simulate the 9 traffic lights by setting the speed onto the street to zero when the light is red. Since the expected value of the traffic light turning green is included in the planning, pedestrians do not opt to go elsewhere once the light turns red. We also simulated a building evacuation (Figure 5), a scenario which is of interest for urban planning, and also served to highlight the interesting congestion phenomena that occur when groups travel at different speeds. Finally, we created

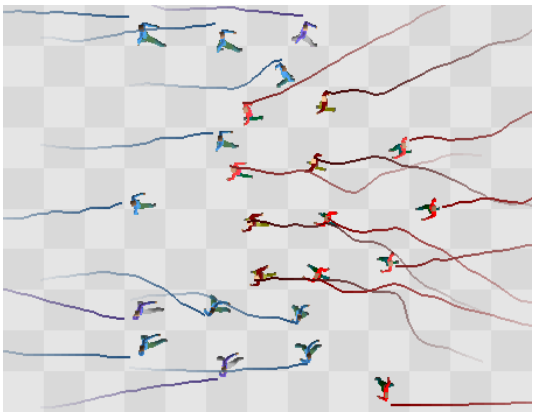


Figure 7: A continuum crowds group (blue) interacts with a group of Reynolds agents (red).

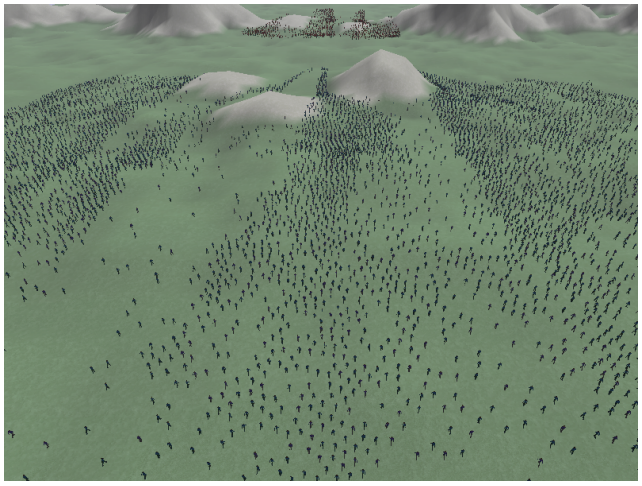


Figure 8: A large army chases a smaller one through mountainous terrain in real-time.

two simulations in which people predictively reacted to moving obstacles. We projected discomfort and increased speed in front of the obstacles so that people would move quickly out of the way. In our first example, cars are driving slowly through crowded streets. We consequently added only small amounts of discomfort and speed. In our second example (Figure 6), a user interactively flew a flying saucer agent through our simulation at much higher speeds. We correspondingly increased the amount and size of the region of increased speed and discomfort. Despite the 5 fps simulations rate, the pedestrians correctly avoided the flying saucer, and the 24 fps renderer provided smooth motion. The grid resolution is equal to the size of the sidewalk tiles shown in Figure 6.

6 Discussion and Future Work

In this paper we presented a novel crowd simulation framework based on a continuum perspective rather than per-agent dynamics. In this respect, we further develop the abstract model proposed by Hughes [2003]. We describe several enhancements to this model that produce more realistic crowd behavior. Most notably, we changed the continuous density field into a particle description of the crowd and developed a full set of continuous dynamics for this representation. We added a velocity-dependent term to the speed computation which is crucial for reproducing emergent phenomena that have been widely noted in real crowds. We also added a distance term to the optimal path computation which is psychologically plausible, and reduced the oscillations in Hughes's model. In addition, we described an algorithm for the efficient simulation of our continuum model. Finally, we showed how our model could be adapted to several interesting situations including traffic lights and predictive avoidance of moving objects.

Our model has numerous advantages over previous systems in graphics. First, our model unifies global planning and collision avoidance. This means that individuals in our simulations do not face conflicting requirements between local collision avoidance and global planning, and they exhibit smoother motion than has been previously reported. This brings our model closer to real crowds, which are characterized by surprisingly few sudden changes in direction. This has also allowed our system to capture a number of emergent phenomena that have been reported in the crowd literature, including lane formation and short lived vortices during turbulent congestion. By decoupling the simulator from the renderer, we were able to simulate these phenomena interactively.

It is possible to integrate our model with agent models. In our examples, the moving cars (driven by a simple rule) and the flying saucer (driven interactively) are all agents. We also show integration with more traditional agents in our video, where our model interacts with an "unaligned collision avoidance" behavior. Figure 7 shows a simulation of a continuum crowd interacting with Reynolds agents.

Like all models, our continuum approach makes simplifying assumptions. Most notably, we do not take into account visual occlusions or any other form of uncertainty, effectively assuming that people really know the dynamic properties of the environment. While this assumption may be reasonable for daily urban commuting, limited vision certainly influences real crowds in unknown environments. In those cases, global knowledge planning is an approximation, in a similar way that local proximity knowledge is an approximation. We believe that global knowledge approximation is justifiable in the cases where our model most directly applies: large crowds navigating open and familiar areas. More importantly, compared to alternative approaches and other approximations, the use of global knowledge provides more smooth crowd behavior and allows us to plan for everyone at once, achieving high simulation speeds. We also think that it is possible to incorporate partial visibility and we hope to address this issue in the future.

We also assume that people can change direction without respect to inertia. While this is justified for walking crowds, it becomes unrealistic when people run. This could be addressed by giving each person a state consisting of position and velocity, and solving a 4D eikonal equation through this state space; however this method would not be real-time. Another useful extension would be to solve for the potential function across nonuniform grids. This would allow sparse data points in areas with few people, and finer discretizations in areas of congestion, which might yield a substantial speedup.

Our system does not have the flexibility and individual variability of the full agent-based approach. Continuum crowds are designed specifically to model multiple large homogeneous groups of people who are moving in order to reach a specific goals. Since very large crowds are observed at a distance where it is harder to notice details of individual behavior, we felt that smooth movement, lane formation and planned motion were more important features to preserve than individual variability. At the same time, occasional variability can be achieved by inserting a number of agents within the continuum crowds. The continuum crowd model treats agents as moving obstacles and naturally handles the unpredictability of their behavior. We suspect that this hybrid model of continuum crowds with embedded agents may be the most natural use of continuum crows in larger interactive frameworks including games.

The continuum crowd model is not appropriate for all crowd behavior. For example, it does not take into account the regime where people are so tightly packed that contact forces between them dominate the physics. It is also limited by the requirement that people move with a common goal. As a result, browsing without a specific goal, such as visiting an art gallery, is not well suited for the continuum formulation. At the same time, the continuum framework can be applied to wide range of scenarios that at first do not seem to meet the common goal hypothesis. These scenarios are achieved by dynamically changing goals and discomfort fields. For example, environment exploration can be modeled by setting goals to collectively unseen regions. We have similarly produced examples of "posse chasing" behavior, where goals are set to the location of target people, and avoidance behavior, where goals are set to a large-radius circle around people that should be avoided. The unassigned-seat theater-filling example can also be modeled by removing each "seat goal" as they get occupied.

We are encouraged that our algorithm has recently been evaluated and licensed by Electronic Arts for use in next-generation games. According to their feedback our method appears to be uniquely attractive for a number of reasons: it produces more

natural crowd threading behavior than the agent-based methods evaluated; it doesn't "get stuck" even during very long simulations; real-time performance is possible with very large crowds, and it integrates well with autonomous agents.

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by the UW Animation Research Labs, NSF grants EIA-0121326, CCR-0092970, IIS-0113007, Alfred P. Sloan Fellowship, NSF Graduate Research Fellowship, Electronic Arts, Sony, and Microsoft Research.

References

ARKIN, R. 1987. In *1987 IEEE International Conference on Robotics and Automation.*, vol. 4, 264–271.

BAYAZIT, O. B., LIEN, J.-M., AND AMATO, N. M. 2002. Better group behaviors in complex environments with global roadmaps. In *Int. Conf. on the Sim. and Syn. of Living Sys. (Alife)*, 362–370.

BENNEWITZ, M., AND BURGARD, W. 2001. Finding solvable priority schemes for decoupled path planning techniquesfor teams of mobile robots. Proceedings of the 9th International Symposium on Intelligent Robotic Systems (SIRS).

BROGAN, D. C., AND HODGINS, J. K. 1997. Group behaviors for systems with significant dynamics. In *Autonomous Robots*, 137–153.

CHENNEY, S. 2004. Flow tiles. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 233–242.

CLEMENTS, R. R., AND HUGHES, R. L. 2004. Mathematical modelling of a mediaeval battle: the battle of Agincourt, 1415. *Math. Comput. Simul. 64*, 2, 259–269.

COLOMBO, R. M., AND ROSINI, M. D. 2005. Pedestrian flows and nonclassical shocks. *Mathematical Methods in the Applied Sciences 28*, 13.

CORDEIRO, O. C., BRAUN, A., SILVEIRA, C. B., MUSSE, S. R., AND CAVALHEIRO, G. G. H. 2005. Concurrency on social forces simulation model. First International Workshop on Crowd Simulation.

FEDKIW, R., STAM, J., AND JENSEN, H. 2001. Visual Simulation of Smoke. In *Computer Graphics (SIGGRAPH 2001)*, ACM, 15–22.

FEURTEY, F. 2000. *Dept. of EE.* Master's thesis, Univ. of Tokyo.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 29–38.

GOLDENSTEIN, S., KARAVELAS, M., METAXAS, D., GUIBAS, L., AARON, E., AND GOSWAMI, A. 2001. Scalable nonlinear dynamical systems for agent steering and crowd simulation. *Computers & Graphics 25*, 6 (Dec.), 983–998.

HEÏGEAS, L., LUCIANI, A., THOLLOT, J., AND CASTAGNÉ, N. 2003. A physically-based particle model of emergent crowd behaviors. In *GraphiCon*.

HELBING, D., MOLNÁR, P., AND SCHWEITZER, F. 1994. Computer simulations of pedestrian dynamics and trail formation. In *Evolution of Natural Structures*, 229–234.

HELBING, D., MOLNÁR, P., FARKAS, I. J., AND BOLAY, K. 2001. Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design 28*, 361–383.

HELBING, D., BUZNA, L., AND WERNER, T. 2003. Self-organized pedestrian crowd dynamics and design solutions. *Traffic Forum 12*.

HONGWAN, L., WAI, F. K., AND CHOR, C. H. 2003. A study of pedestrian flow using fluid dynamics. Tech. rep.

HUGHES, R. L. 2002. A continuum theory for the flow of pedestrians. *Transportation Research Part B 36*, 6 (july), 507535.

HUGHES, R. L. 2003. The flow of human crowds. *Annu. Rev. Fluid Mech. 35*, 169–182.

KAMPHUIS, A., AND OVERMARS, M. H. 2004. Finding paths for coherent groups using clearance. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 19–28.

KETCHELL, N. 2002. A technical summary of the aea egress code. Tech. Rep. 1, AEA Technology.

KIMMEL, R., AND SETHIAN, J. A. 2001. Optimal Algorithms for Shape from Shading and Path Planning. *Journal of Mathematical Imaging and Vision 14*, 237–244.

KIRCHNER, A., AND SCHADSCHNEIDER, A. 2002. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. In *Physica A*, vol. 312, 260–276.

LAMARCHE, F., AND DONIKIAN, S. 2004. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum 23*, 3 (Sept.), 509–518.

LI, T.-T., AND CHOU, H.-C. 2003. Motion planning for a crowd of robots. In *IEEE International Conference on Robotics and Automation*.

LI, T.-Y., LENG, Y.-J., AND CHANG, S.-I. 2001. Simulating virtual crowds with a leader-follower model. In *Proceedings of 2001 Computer Animation Conference*.

LOSCOS, C., MARCHAL, D., AND MEYER, A. 2003. Intuitive crowd behaviour in dense urban environments using local laws. In *Theory and Practice of Computer Graphics (TPCG'03)*.

MASSIVE SOFTWARE, 2006. http://www.massivesoftware.com.

METOYER, R. A., AND HODGINS, J. K. 2004. Reactive pedestrian path following from examples. *The Visual Computer 20*, 10, 635–649.

MUSSE, S. R., AND THALMANN, D. 1997. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation '97*, 39–51.

MUSSE, S. R., BABSKI, C., CAPIN, T., AND THALMANN, D. 1998. Crowd modelling in collaborative virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, 115–123.

OPENSTEER, 2006. http://opensteer.sourceforge.net.

PARKER, L. E. 1993. Designing control laws for cooperative agent teams. In *IEEE International Conference on Robotics and Automation*, 582–587.

PELECHANO, N., OBRIEN, K., SILVERMAN, B., AND BADLER, N. 2005. Crowd simulation incorporating agent psychological models, roles and communication. First International Workshop on Crowd Simulation.

PETTRÉ, J., LAUMOND, J.-P., AND THALMANN, D. 2005. A navigation graph for real-time crowd animation on multilayered and uneven terrain. First International Workshop on Crowd Simulation.

REYNOLDS, C. W. 1987. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, vol. 21, 25–34.

REYNOLDS, C., 1999. Steering behaviors for autonomous characters.

SHAO, W., AND TERZOPOULOS, D. 2005. Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 19–28.

STILL, G. 2000. *Crowd Dynamics.* PhD thesis, University of Warwik, UK.

SUNG, M., GLEICHER, M., AND CHENNEY, S. 2004. Scalable behaviors for crowd simulation. *Computer Graphics Forum 23*, 3 (Sept.), 519–528.

SUNG, M., KOVAR, L., AND GLEICHER, M. 2005. Fast and accurate goal-directed motion synthesis for crowds. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 291–300.

TSAI, R., ZHAO, H., AND OSHER, S. 2005. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM Journal of Numerical Analysis 42*, 6.

TSITSIKLIS, J. N. 1995. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control 40*, 9 (Sept.), 1528–1538.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, 43–50.