Aggregate Dynamics for Dense Crowd Simulation
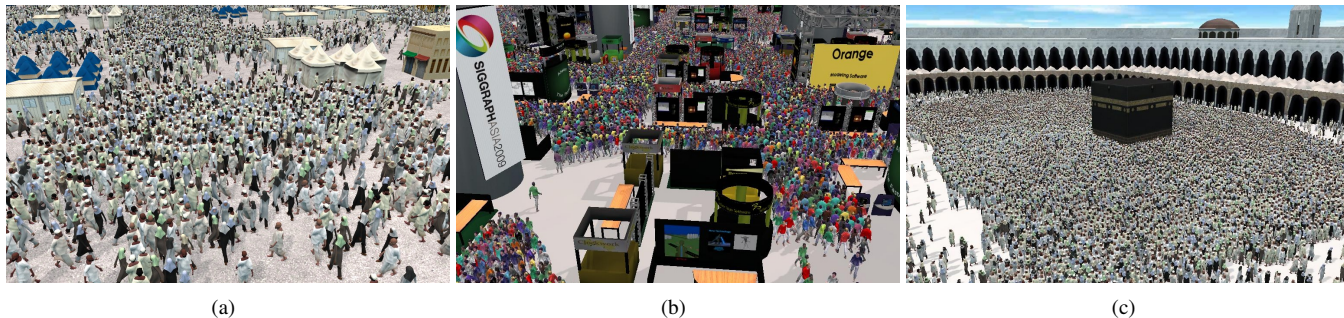
Rahul Narain   Abhinav Golas   Sean Curtis   Ming C. Lin

University of North Carolina at Chapel Hill*

(a)   (b)   (c)

Figure 1: *Some examples of large, dense crowds simulated with our technique. (a) 100,000 pilgrims moving through a campsite. (b) 80,000 people on a trade show floor. (c) 25,000 pilgrims with heterogeneous goals in a mosque.*

Abstract

Large dense crowds show aggregate behavior with reduced individual freedom of movement. We present a novel, scalable approach for simulating such crowds, using a dual representation both as discrete agents and as a single continuous system. In the continuous setting, we introduce a novel variational constraint called *unilateral incompressibility*, to model the large-scale behavior of the crowd, and accelerate inter-agent collision avoidance in dense scenarios. This approach makes it possible to simulate very large, dense crowds composed of up to a hundred thousand agents at near-interactive rates on desktop computers.

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

Keywords: crowds, planning, continuum, incompressibility

1 Introduction

Human crowds exhibit highly complex behavior driven by individual decisions of agents with respect to their goals, environmental obstacles, and other nearby agents. The problem of simulating virtual crowds has attracted increasing attention recently, due to its use for education and entertainment, emergency training, architectural design, urban planning, policy making, traffic engineering, and numerous other applications. Existing approaches often simplify this

---
*E-mail: {narain,golas,seanc,lin}@cs.unc.edu

problem by separating global planning from local collision avoidance. The local collision avoidance module requires each agent to take into account the motion of its nearby neighbors; this step quickly turns into a major computational bottleneck for very dense crowds.

In this paper, we focus specifically on the problem of simulating the inter-agent dynamics of large, dense crowds in real time. Such crowds exhibit a low interpersonal distance and a corresponding loss of individual freedom of motion. This observation suggests that the behavior of such crowds may be modeled efficiently on a coarser level, treating its motion as the flow of a single aggregate system. Based on such an abstraction, we develop a novel inter-agent avoidance model which decouples the computational cost of local planning from the number of agents, allowing very large-scale crowds consisting of hundreds of thousands of agents to be simulated scalably at interactive rates.

Our method combines a Lagrangian representation of individuals with a coarser Eulerian crowd model, thus capturing both the discrete motion of each agent and the macroscopic flow of the crowd. In dense crowds, the finite spatial extent occupied by humans becomes a significant factor. This effect introduces new challenges, as the flow varies from freely compressible when the density is low to incompressible when the agents are close together. This characteristic is shared by many other dynamical systems consisting of numerous objects of finite size, including granular materials, hair, and dense traffic. We propose a new mathematical formulation to model the dynamics of such aggregate systems in a principled way.

The key results of this work can be summarized as follows:

- A novel hybrid representation for large crowds with discrete agents using both Lagrangian and Eulerian methods (§3);
- A new continuum projection method that enforces density-dependent incompressibility to model the varying behavior of human crowds (§4);
- A scalable crowd simulation that can model hundreds of thousands of agents at interactive rates on current desktops (§5).

Fig. 1 shows some of our results for dense crowds of up to 100,000 agents closely packed in complex scenes simulated at interactive rates with our techniques. More analysis of our system's performance can be found in §5.

2 Related Work

In this section, we give a brief overview of prior work related to crowd modeling and multi-agent navigation. Crowd simulation consists of many different components, including global planning, local avoidance, behavioral modeling, and motion synthesis. In the interest of space, we only touch upon the former two topics here, which are most directly related to our work. We refer the interested readers to the excellent surveys [Schreckenberg and Sharma 2001; Thalmann et al. 2006; Pelechano et al. 2008; Pettré et al. 2008] for more details.

Many different techniques have been proposed for modeling the motion of multiple human agents in a crowd. The primary task in this problem is to compute each agent's path towards its goal, while avoiding collisions with obstacles and other agents and reproducing natural human behavior. On a broad level, the problem can be separated into *global* planning and *local* behavior.

Local models for agent behavior can be traced back to the seminal work of Reynolds [1987; 1999] who demonstrated emergent flocking and other behaviors from simple local rules. A large body of further work has accounted for sociological factors [Musse and Thalmann 1997], psychological effects [Pelechano et al. 2005], directional preferences [Sung et al. 2004], social forces [Helbing and Molnár 1995; Cordeiro et al. 2005; Gayle et al. 2009; Sud et al. 2007], and other models of pedestrian behavior, including [Shao and Terzopoulos 2007; Paris et al. 2007; Yeh et al. 2008] and many others. Many methods have also been proposed for collision avoidance between nearby agents. These include geometrically-based algorithms [Fiorini and Shiller 1998; Feurtey 2000; Guy et al. 2009; Sud et al. 2008; van den Berg et al. 2008a; van den Berg et al. 2008b; van den Berg et al. 2009], grid-based methods [Loscos et al. 2003; Lin et al. 2009], force-based methods [Heigeas et al. 2003; Lakoba et al. 2005; Sugiyama et al. 2001; Sud et al. 2007], Bayesian decision processes [Metoyer and Hodgins 2003], and divergence-free flow tiles [Chenney 2004].

Local control of agents cannot properly model the behavior of agents that aim to move towards specified goals, because of the possibility of getting stuck behind an obstacle. Therefore, local models are often combined with global planning techniques for practical crowd simulation. These methods typically represent the collision-free space as a graph, and perform search to compute the path for each agent [Funge et al. 1999; Bayazit et al. 2002; Kamphuis and Overmars 2004; Lamarche and Donikian 2004; Pettré et al. 2005; Sung et al. 2004; Sud et al. 2007].

Our approach is closest to some previous work on using continuum-based models for crowd dynamics. A continuum theory for the flow of pedestrians was proposed by Hughes [2003] and then extended by Treuille et al. [2006]. This approach combines global and local planning in a single optimization-based framework, and gives compelling results for many kinds of crowds. However, Treuille et al. state that this approach is not suited for dense crowds where people are very closely packed and tend to come into contact.

In other areas, some approaches have used models inspired by fluid dynamics to control large groups of robots [Shimizu et al. 2003; Kerr and Spears 2005; Pimenta et al. 2008] and to handle collision in cloth simulation [Sifakis et al. 2008]. Our work shares some common themes with the concurrent work of McAdams et al. [2009], who accelerate hair simulation by coupling the motion of individual hair strands to an incompressible Eulerian flow. Note that our unilateral incompressibility constraint could also be applied in their method to naturally handle hair separation.
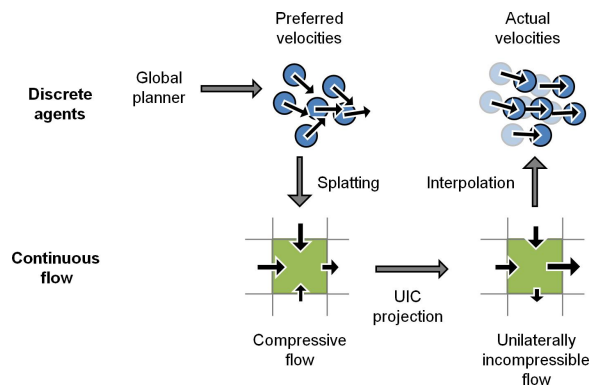

Figure 2: *The system overview of our algorithm.*

3 Discrete and Continuous Crowds

Our approach is motivated by the observation that in a dense crowd, individual agents have a reduced freedom of movement, as they are tightly constrained by nearby agents. This observation has been exploited by previous work on continuum models for medium-density crowds [Hughes 2003; Treuille et al. 2006], and it has been noted that crowds at high density show behavior similar to granular flows [Helbing et al. 2005]. We are motivated by this observation to develop a crowd model that directly describes the aggregate motion of the crowd as a whole.

Specifically, in our approach, we augment the standard representation of a crowd as a set of agents with a continuous representation, which characterizes the crowd as a continuum fluid described by a density and flow velocity. We map the idea of local collision avoidance into the continuous domain to obtain a variational constraint on the crowd flow, which we introduce as the *unilateral incompressibility constraint* (UIC). This constraint acts as a large-scale collision avoidance step to accelerate the simulation.

Our approach can be used as a local planning module in conjunction with a global planner, such as a roadmap-based algorithm or the continuum-based optimization [Treuille et al. 2006] on a coarse grid, for the simulation of large dense crowds. The main simulation loop is structured as follows (Fig. 2): At every timestep, after each agent i determines its preferred velocities $\tilde{\mathbf{v}}_i$ using a global planner (§3.1), the discrete set of agents is converted to a continuous representation, yielding density ρ and velocity $\tilde{\mathbf{v}}$ (§3.2). The UIC is solved in this continuous setting to give the corrected velocity field $\mathbf{v}$ (§4). Finally, agents determine their actual motion based on the velocities sampled from $\mathbf{v}$ (§3.3).

In the remainder of this section, we set the background by describing the familiar discrete representation of a crowd and how it relates to the continuous model. The projection operation to solve the UIC in the continuous setting will then be described in §4.

3.1 Agent-Level Planning

To determine the preferred velocities for each agent, a high-level model of agent behavior is necessary, such as a global planning step. Typically, global planning ignores the presence of other agents and determines a path that avoids the large static obstacles in the scene. We use the global planner to determine the preferred velocities of agents, by taking the initial segment of the globally planned path as the preferred direction of motion. The task of reconciling these preferred velocities with collision avoidance among agents is then performed by the UIC solve.

In our implementation, we used a roadmap-based approach (see *e.g.* [Bayazit et al. 2002; Sud et al. 2007; Sud et al. 2008] for details). However, our approach is entirely agnostic to the particular global planner that is used; more complex behavioral models, a continuum planning approach, or user-scripted goal directions could be used instead.

3.2 Agents and the Grid

We store the continuous fields ρ and $\mathbf{v}$ on a staggered grid, as is common in fluid simulation. The density ρ is stored at both cell centers and edges, to facilitate the numerical method described in §4. The x-component of velocity is stored on the cell edges normal to x, and similarly for the y component.

The transfer of information from discrete agents to the grid closely follows the particle-in-cell method of fluid simulation [Harlow 1963; Zhu and Bridson 2005] (also used for crowd planning by [Treuille et al. 2006]). To obtain the values of density ρ and velocity $\tilde{\mathbf{v}}$ on the grid, we accumulate the values carried by nearby agents, weighted by the bilinear interpolation weights $w_{\mathbf{x}}(\mathbf{x}_i)$ associated with the agent position $\mathbf{x}_i$. That is,

$$\rho(\mathbf{x}) = \sum_i w_{\mathbf{x}}(\mathbf{x}_i)m_i, \qquad (1)$$

$$\tilde{\mathbf{v}}(\mathbf{x}) = \frac{\sum_i w_{\mathbf{x}}(\mathbf{x}_i)\tilde{\mathbf{v}}_i}{\sum_i w_{\mathbf{x}}(\mathbf{x}_i)} \qquad (2)$$

We take the mass m_i of each agent to be unity. With this procedure, the crowd is converted to a continuous representation, on which the UIC projection can be performed.

3.3 Agent Motion

The UIC projection, which will be described in detail in §4, defines a flow field through ρ and $\mathbf{v}$ which prevents inter-agent collision on a macroscopic level. Here, we first describe how to use the computed flow field to determine the actual velocity of each agent and thus to execute its motion.

In dense regions, the agents are constrained by the flow of the crowd, and we directly use the interpolated grid velocity at the agent position $\mathbf{v}(\mathbf{x}_i)$ as the velocity of agent i. However, in regions of low density, following the flow velocity completely would over-constrain the agents' motion. Accordingly, we interpolate between the continuum velocity $\mathbf{v}(\mathbf{x}_i)$ and the agent's own preferred velocity $\tilde{\mathbf{v}}_i$, depending on the crowd density $\rho(\mathbf{x}_i)$ at its location:

$$\mathbf{v}_i = \tilde{\mathbf{v}}_i + \frac{\rho(\mathbf{x}_i)}{\rho_{max}}\left(\mathbf{v}(\mathbf{x}_i) - \tilde{\mathbf{v}}_i\right). \qquad (3)$$

Finally, the UIC projection can only act towards enforcing separation between agents on a gross level. It is still necessary to introduce an additional step to enforce minimum distances for each pair of individual agents. We perform this by simply moving agents apart if they are too close to each other, following Treuille et al. [2006]. While this simple pairwise collision resolution procedure is not guaranteed to separate all pairs exactly to the preferred distance, it gives good results because agent overcrowding is already prevented by the UIC. Fig. 3 illustrates that both this step and the UIC solve are necessary to prevent inter-agent collisions.

4 The Unilateral Incompressibility Constraint

This section describes the operation that enforces volumetric constraints on the crowd, the primary contribution of our work. The
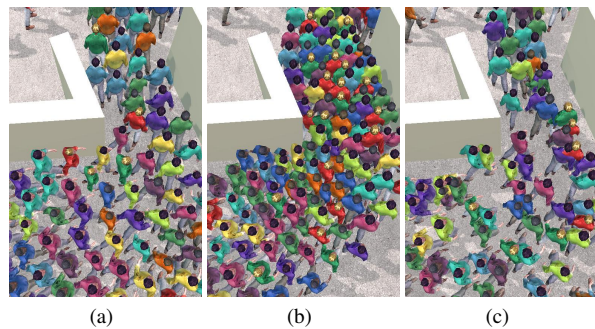
Figure 3: *Agents attempting to exit through a narrow doorway, (a) with both the UIC projection and pairwise collision resolution, (b) without UIC, and (c) without collision resolution.*

role of this operation is a macroscopic counterpart to inter-agent collision avoidance. First, we will develop this in the mathematically continuous setting, then derive the numerical method that implements it on the simulation grid.

We have represented the crowd's state and preferred motion as continuous fields of density ρ and preferred velocity $\tilde{\mathbf{v}}$. Their evolution over time can be seen as a fluid-like system. However, it does not correspond directly to a physical fluid, as it is neither purely incompressible (a crowd can easily converge or disperse, changing its density), nor purely compressible (it is not possible to compress it indefinitely). Instead, we treat a crowd as a new kind of "unilaterally incompressible" fluid, which is a hybrid of the two. We directly impose an inequality constraint on ρ, preventing it from increasing beyond a maximum value $\rho_{\max}$ which corresponds to agents being packed as closely together as possible.

We make the assumption that agents do not come closer together than a specified distance, say $d_{\min}$. This need not represent actual person-to-person contact, but simply a minimal comfortable distance to be maintained. From a macroscopic perspective, this constraint implies an upper bound on the number density of agents,

$$\rho \leq \rho_{\max} = 2\alpha/(\sqrt{3}d_{\min}^2), \qquad (4)$$

with $\rho_{\max}$ denoting the maximum allowed density. We call this the *unilateral incompressibility constraint* (UIC), following the terminology from mechanics of unilateral (inequality) vs. bilateral (equality) constraints. The constant factor $\alpha \leq 1$ is present to allow for the fact that perfect close packing is rarely achieved. This kind of constraint can in fact be applied not just to crowds, but to many other aggregate systems consisting of objects of a finite size, including granular materials and dense traffic flows.

Now, we can choose the actual flow, defined by $\mathbf{v}$, to be one which is close to the preferred flow $\tilde{\mathbf{v}}$, while maintaining the UIC (4). This idea is formalized in the following subsection.

4.1 Unilateral Incompressibility

Formally, we may define the problem as follows. Consider the continuous model of the crowd as a constrained dynamical system, with an inequality-constrained state variable ρ which evolves under the action of the field $\tilde{\mathbf{v}}$. The relationship between ρ and $\mathbf{v}$ follows from the fact that the number of agents is conserved:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = 0. \qquad (5)$$

To ensure that the UIC (4) is maintained at all times, a correction must be applied to $\tilde{\mathbf{v}}$ so that ρ remains within the feasible space.

The issue thus is to determine the corrected velocity $\mathbf{v}$ which is in some sense "close" to $\tilde{\mathbf{v}}$ but maintains UIC.

We make the assumption that agents will try to make as much progress in their desired direction as possible, subject to collision and walking speed constraints. Then, the choice of $\mathbf{v}$ is determined by maximizing $\int \rho \mathbf{v} \cdot \tilde{\mathbf{v}}$, subject to some maximum walking speed, $||\mathbf{v}|| \leq v_{\max}$. In our implementation, $v_{\max}$ is a constant, but for additional realism it can be made variable for each grid cell, depending on the density or the characteristics of nearby agents.

For the above condition, it can be shown via variational calculus that the optimal solution is of the form

$$\mathbf{v} = v_{\max} \frac{\tilde{\mathbf{v}} - \nabla p}{||\tilde{\mathbf{v}} - \nabla p||} \tag{6}$$

for some scalar "pressure" $p \geq 0$ satisfying

$$\rho < \rho_{\max} \quad \Rightarrow \quad p = 0, \tag{7}$$
$$p > 0 \quad \Rightarrow \quad \nabla \cdot \mathbf{v} = 0. \tag{8}$$

The interpretation of (6) is as follows. The crowd flow experiences a nonnegative pressure p which prevents the flow from converging to a density higher than $\rho_{\max}$. After the effect of pressure, agents amplify their adjusted velocity to attain the maximum allowed speed $v_{\max}$. Note that since (7) is also equivalent to

$$p > 0 \Rightarrow \rho = \rho_{\max}, \tag{9}$$

there is a *complementarity* between the unilaterally constrained terms ρ and p, in that either $\rho = \rho_{\max}$ or $p = 0$ must hold at any point. Intuitively, this means that pressure only acts in regions when the density ρ is at the maximum. Where $\rho < \rho_{\max}$, agents are not densely packed and exert little influence on each other, and the crowd flows freely.

4.2 Numerical Method

Here, we develop the numerical method for performing the UIC projection on the simulation grid.

Due to the nonlinearity of 6, we do not solve it directly. Instead, we split it into the composition of two operations,

$$v_{\max} \frac{\tilde{\mathbf{v}} - \nabla p}{||\tilde{\mathbf{v}} - \nabla p||} = \mathsf{renorm}(\mathsf{psolve}(\tilde{\mathbf{v}})), \tag{10}$$

where $\mathsf{psolve}(\tilde{\mathbf{v}}) = \tilde{\mathbf{v}} - \nabla p$, and $\mathsf{renorm}(\hat{\mathbf{v}}) = v_{\max}\hat{\mathbf{v}}/||\hat{\mathbf{v}}||$.

We note that collision avoidance is performed by psolve, while the role of renorm is merely to speed up the flow. To make analysis possible, we neglect the effect of renormalizing velocities when solving for the pressure; this will be accounted for later.

To obtain the value of p, we start by substituting $\mathsf{psolve}(\tilde{\mathbf{v}})$ into the evolution equation, and find

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \tilde{\mathbf{v}}) + \nabla \cdot (\rho \nabla p). \tag{11}$$

In the discrete setting with cell size Δx, the gradient of pressure ∇p is defined at cell edges by finite differencing the pressure at adjacent cells. The divergence $\nabla \cdot \mathbf{u}$ of some vector field $\mathbf{u}$ on a cell is given by summing the flux over the cell edges,

$$\nabla \cdot \mathbf{u} \Delta x^2 \approx \sum_{\mathrm{edges}} \mathbf{u} \cdot \mathbf{n} \Delta x. \tag{12}$$

At the nth timestep, the density values ρ^n and pre-projection velocities $\tilde{\mathbf{v}}^n$ are known. To the first order, the density at the $(n + 1)$th timestep is then given by

$$\rho^{n+1} = \rho^n - \nabla \cdot (\rho^n \tilde{\mathbf{v}}^n)\Delta t + \nabla \cdot (\rho^n \nabla p^n)\Delta t. \tag{13}$$

With discrete timesteps, it is preferable to apply UIC to the value at the next timestep, rather than instantaneously in differential form as in the previous subsection. Applying the constraint (7) to ρ^{n+1} itself, we obtain a linear complementarity problem (LCP). A general LCP is of the form

$$\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b},$$
$$\mathbf{z} \geq 0, \quad \mathbf{x} \geq 0, \quad \mathbf{z}^T \mathbf{x} = 0. \tag{14}$$

With the substitution $\sigma = \rho_{\max} - \rho$, it can be seen that the problem is equivalent to an LCP with

$$\mathbf{z} = \sigma^{n+1}, \tag{15}$$
$$\mathbf{b} = \sigma^n + \nabla \cdot (\rho^n \tilde{\mathbf{v}}^n)\Delta t, \tag{16}$$
$$\mathbf{A}\mathbf{x} = -\nabla \cdot (\rho^n \nabla \mathbf{x})\Delta t, \tag{17}$$
$$\mathbf{x} = p^n. \tag{18}$$

Also, because the matrix $\mathbf{A}$ is symmetric and positive semidefinite, the LCP is equivalent to a bound-constrained quadratic programming (QP) problem.

This problem has certain useful structure which can be exploited in order to solve it more efficiently than general-purpose QP or LCP solvers. In particular: it is symmetric, positive semidefinite and sparse; an efficient preconditioner, the MIC(0) preconditioner [Bridson and Müller-Fischer 2007], is known; and only nonnegativity constraints exist on the variables. We used the algorithm of Dostál and Schöberl [2005] which exploits these properties to solve the UIC projection efficiently. This is an iterative algorithm which can be warm-started with the pressure values from the previous timestep for improved performance.

After solving the LCP, we obtain the pressure-corrected velocity, say $\hat{\mathbf{v}}$. However, because the effect of velocity renormalization was neglected in the pressure solve, $\mathsf{renorm}(\hat{\mathbf{v}})$ may no longer satisfy the pressure constraint (7). As this constraint is essential for collision avoidance, we solve the pressure projection on the renormalized velocities again. The final velocity, thus, is

$$\mathbf{v} = \mathsf{psolve}(\mathsf{renorm}(\mathsf{psolve}(\tilde{\mathbf{v}}))). \tag{19}$$

In practice, numerical error can cause ρ^n to exceed $\rho_{\max}$ for one timestep. To prevent renorm from amplifying the transient corrective motion it causes, the inner psolve projection is performed with $\mathbf{b}$ clamped to nonnegative values.

4.3 Obstacles

If there are obstacles in the simulation domain, each grid cell that is partially or totally covered by an obstacle has a smaller area available for agents to be present. Therefore, it is simply the density constraint on that cell that has to be modified. For each grid cell, we determine the fraction f of area *not* covered by obstacles. (This can of course be precomputed for static obstacles.) Then, in the projection step, the maximum density is scaled by f to reflect the reduced available area. That is, we simply use $\sigma = f\rho_{\max} - \rho$ and solve the LCP as usual. This simple procedure, inspired by Batty et al. [2007], allows obstacles to be handled without grid stairstepping.
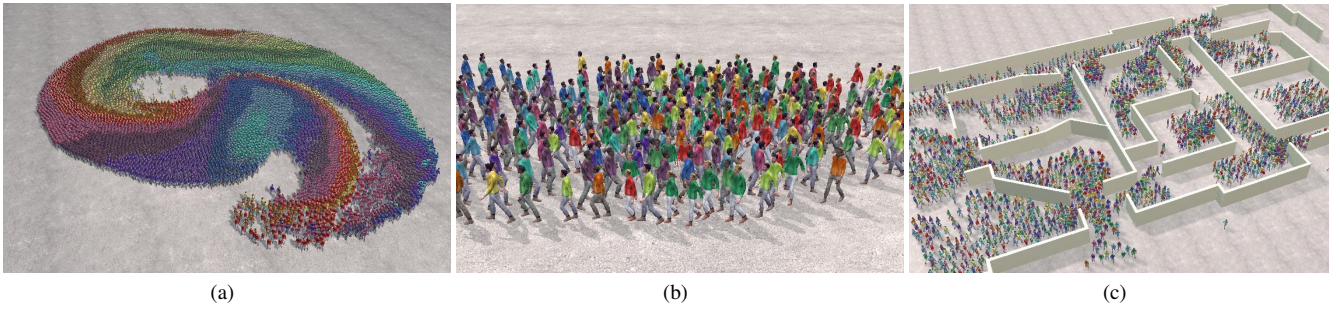
Figure 4: *(a) 10,000 agents in a circle moving to diametrically opposite points. (b) UIC-driven agents (left) interacting with RVO-driven agents (right) in the same simulation. (c) 2,000 agents evacuating a building.*

With this approach, we can also mix our method with other methods for agent-based planning [Helbing and Molnár 1995; Reynolds 1999; van den Berg et al. 2008b] as well as scripted agents in the same simulation. For each time step, we first advance the individually controlled agents, then treat them as obstacles in the continuum model so that the crowd motion will flow around them. In Fig. 4(b) we demonstrate this with a group of agents from our continuum model interacting with agents directed by the reciprocal velocity obstacle (RVO) method of van den Berg et al. [2008b].

4.4 Algorithm Summary

For clarity, here we summarize the full simulation loop.

1. At the beginning of each timestep, we know the position $\mathbf{x}_i$ of each agent.

2. Global planning is performed to determine the preferred velocity $\tilde{\mathbf{v}}_i$ of each agent, taking into account environmental obstacles but not neighboring agents. (For efficiency, the global planner may cache results from earlier timesteps, so that the full planning cost is not paid at each timestep.)

3. The agent positions $\mathbf{x}_i$ and preferred velocities $\tilde{\mathbf{v}}_i$ are transferred to the simulation grid by (1) and (2).

4. If there are moving obstacles in the environment, the free area f of each grid cell is recomputed.

5. The UIC solve is performed, giving the corrected velocity field $\mathbf{v} = \mathsf{psolve}(\mathsf{renorm}(\mathsf{psolve}(\tilde{\mathbf{v}})))$.

6. Each agent determines its actual velocity $\mathbf{v}_i$ taking the corrected flow into account via (3), and updates its position for the next timestep as $\mathbf{x}_i := \mathbf{x}_i + \mathbf{v}_i \Delta t$.

7. Finally, pairwise collision resolution is performed on the new $\mathbf{x}_i$ to handle inter-agent collisions.

5 Results and Discussion

We have tested our algorithm on several challenging scenes with large, dense crowds. Our simulations show that the method can efficiently handle crowds of hundreds of thousands of agents without a notable increase in computation time.

Figs. 4(a) and 5 show two examples that demonstrate the behavior of our method in some idealized scenarios. In the former, 10,000 agents attempt to cross over to the opposite points of a circle and meet in the middle, before forming a vortex to resolve the deadlock. The latter example shows a four-way crossing of pedestrians under sparse, medium and dense flow. With sparse flow, streams of agents

are able to pass through easily, but as the density increases, a complex pattern of lanes and vortices forms to resolve the congestion.

In Figs. 1(b), 6(a), 6(b), we show a real-world scenario of 80,000 individual agents being evacuated from a trade show. Since there are many obstacles in the scene, evacuation causes congestion and a gradual outflow.

We also took the Hajj pilgrimage as another real example of very large crowds. Fig. 1(a), 6(c) show a scenario based on the Plain of Arafat campsite on the pilgrimage route. Through this large environment, 100,000 pilgrims travel in different directions and form lanes and vortices as they pass each other. We also simulated the motion of pilgrims in al-Masjid al-Haram (the Grand Mosque) in Mecca. 25,000 agents with different goal directions form a dense, complex flow with no collisions, as shown in Fig. 1(c).

5.1 Performance

We measured the performance of our algorithm on an Intel Core i7-965 machine at 3.2 GHz with 5.99 GB of RAM. The timing results for all our examples are shown in Table 1. Even with very large numbers of agents, we can achieve close to interactive performance. Our method supports general scenarios with independent, heterogeneous agents; the number of unique goals has no effect on the performance.

Fig. 7 shows a profile of the time spent in different stages of the algorithm, for the circle sequence of Fig. 4(a). The computational cost of the UIC solve is approximately linear with the number of actively constrained cells, *i.e.* those at maximum density, but is independent of the actual number of agents in the scene. The other expensive step is the pairwise collision resolution, which is an unavoidable per-agent cost.

One of the benefits of our UIC-based approach is that it allows a much simpler per-agent scheme to be employed. This is evident in Fig. 8, which shows how our algorithm scales with the number of agents. For very large numbers of agents, the per-agent processing cost begins to dominate the computation time, but this cost is nevertheless much lower than that of existing methods. In particular, our method makes it possible to simulate a crowd of *1 million agents* at 3 seconds per frame on a desktop computer. For comparison, we used the most recent publicly-available implementation of the RVO method [van den Berg et al. 2008b], a geometrically-based collision avoidance method, whose performance is also shown. It failed to run on scenes containing more than 70,000 agents.

We also tested the effectiveness of our approach in performing pairwise collision avoidance. In our experiments, we found that moving colliding agents to a separation somewhat larger than the minimum intersection-free distance $d_{\min}$ after enforcing UIC gives smoother,
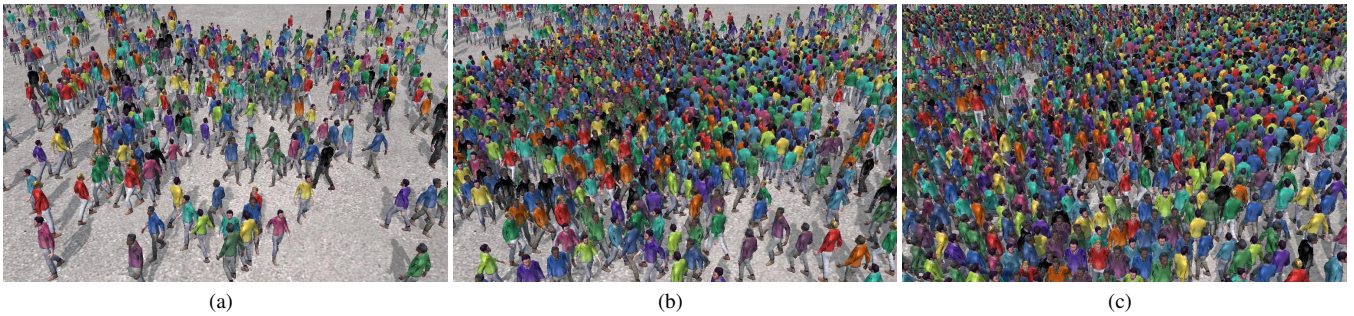
Figure 5: *A four-way crossing of pedestrians entering the scene at rates of (a) 40, (b), 80, and (c) 120 agents per second.*
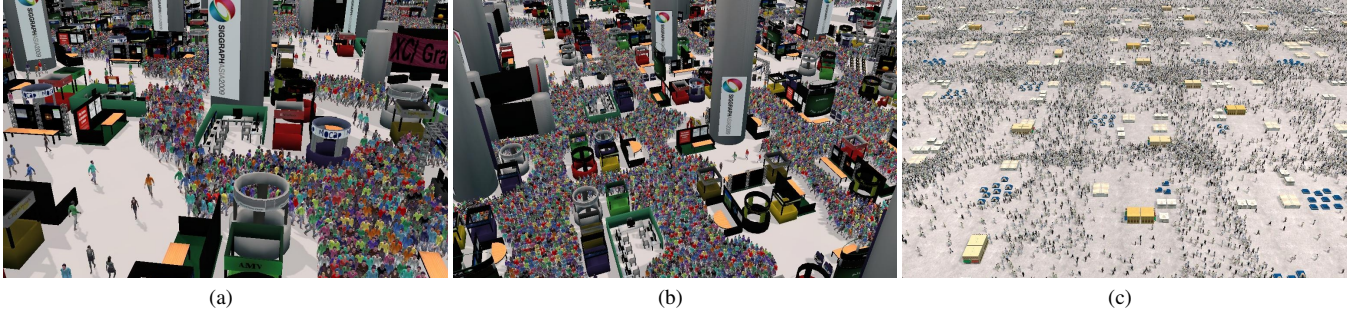
Figure 6: *(a, b) Two additional views of the evacuation in Fig. 1(b). (c) A zoomed out view of the campsite in Fig. 1(a).*

jitter-free motion and a better spatial distribution of agents. For experimental validation, we measured the distance of each agent to its nearest neighbor over all frames of the circle scenario. The nearest distances have a mean of $1.25d_{\min}$ in the densest regions, and only less than 0.12% of agents approach slightly closer than $d_{\min}$ over the entire sequence.

Over the different scenes shown in Table 1, the performance varies depending on the complexity of the scene, including static obstacles and the global planning roadmap. Also, when scaling up to a million agents, memory issues begin to be a significant factor. Overcoming these new bottlenecks is the next challenge that has to be addressed.

5.2 Limitations and Future Work

Since our method is based on some abstractions and approximations to the true individual-driven behavior of a crowd, it has some limitations when these approximations are not applicable.

Firstly, the pressure projection looks only at local information, and cannot anticipate future collisions from distant agents. Thus, two groups of agents approaching each other will not react until they are adjacent to each other. Since this requires non-local information, perhaps it should be a part of a global planning step instead. For example, our method could be used in conjunction with the potential fields approach of continuum crowds [Treuille et al. 2006], which has global lookahead but does not handle the constraints of a dense crowd. (In their original work, agent overlaps are prevented by pairwise collision resolution, which Fig. 3 shows to be inadequate for dense crowds, or by using a roughly agent-sized grid, which is expensive.)

When defining the continuum constraint handling step, we chose to optimize only progress in the desired direction, in order to have a simple mathematical formulation. However, in real life human crowds display other behavioral features; for example, there is often

| Scenario	Agents	Grid size	Time/frame
Circle (Fig. 4(a))	10 k	40×40	34.2 ms
Crossing (Fig. 5)	6 k	40×40	16.5 ms
Building (Fig. 4(c))	2 k	90×60	16.1 ms
Trade show (Fig. 1(b))	80 k	120×108	806 ms
Campsite (Fig. 1(a))	100 k	120×90	447 ms
Mosque (Fig. 1(c))	25 k	80×80	88.1 ms

Table 1: *The performance of our method on several scenarios.*

an informal cultural convention to walk on one side of the path to avoid oncoming pedestrian traffic. It would be worthwhile to investigate the addition of such cultural and social behaviors to our method to enhance the realism of the results.

We anticipate that our approach can be combined with other techniques to enable a level-of-detail approach for simulating extremely large crowds. In specific regions where detailed individual motion is needed, such as close to the viewpoint, a more expensive agent-based scheme could be used, while distant agents would be efficiently simulated using our model on a coarser grid. Adaptively refined grids can also be used for further control over the level of detail. This would achieve high quality behavior where it is desired, without sacrificing performance even for extremely large crowds.

Performing motion synthesis to map animated characters to the simulated paths is another challenge. In dense crowds, velocities fluctuate and often drop near zero, which we found leaves most simple approaches inadequate and vulnerable to foot skating artifacts. Generating more plausible character motion while remaining scalable to very large crowds is an open problem that we hope can be addressed by future work in motion synthesis.
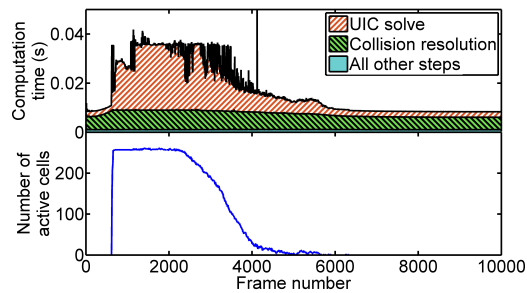
(a)
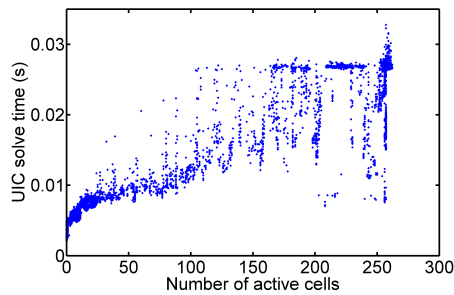
(b)

Figure 7: *(a) The simulation time per frame (top) for the circle scenario, compared with the number of grid cells where the UIC constraint is active (bottom). The scenario contained 10,000 agents on a 40×40 grid. The majority of the simulation cost is in the collision resolution, which is largely constant, and the UIC solve, whose cost increases approximately linearly with the number of active cells (b).*

6 Conclusion

Dense crowds of humans present a challenging problem for crowd simulation techniques, because of the large number of agents and the highly constrained nature of their motion. We have presented a hybrid continuum-based method for solving the behavior of such crowds in a scalable fashion. By employing a new mathematical constraint to enforce inter-agent separation in the continuous domain, we are able to decouple the computational cost of local collision avoidance from the actual number of agents in the scene. Our method makes it possible for the first time to simulate very large crowds of up to a hundred thousand agents at near-interactive rates on desktop computers. Finally, the unilateral incompressibility constraint is a very general formulation for modeling large aggregates of objects, and we anticipate its applicability to many other domains such as animation of granular materials and traffic simulation.

Acknowledgments

References

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3, 100.
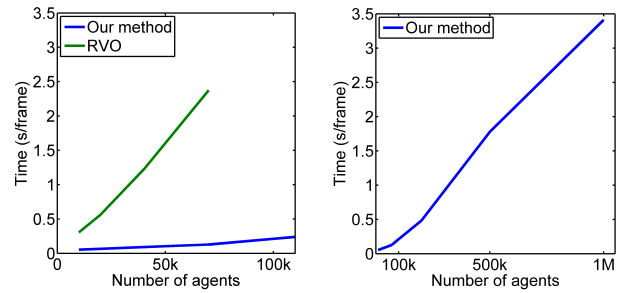
Figure 8: *Our algorithm scales well with the number of agents up to 1 million, while the RVO method is more expensive and fails to run on simulations with more than 70,000 agents.*

BAYAZIT, O. B., LIEN, J.-M., AND AMATO, N. M. 2002. Better group behaviors in complex environments with global roadmaps. *Proc. 8th Intl. Conf. Artificial Life*, 362–370.

BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation. In *ACM SIGGRAPH 2007 courses*, 1–81.

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

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

DOSTÁL, Z., AND SCHÖBERL, J. 2005. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl. 30*, 1, 23–43.

FEURTEY, F. 2000. *Simulating the Collision Avoidance Behavior of Pedestrians*. Master's thesis, University of Tokyo, School of Engineering.

FIORINI, P., AND SHILLER, Z. 1998. Motion planning in dynamic environments using velocity obstacles. *Intl. J. on Robotics Research 17*, 7, 760–772.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. *Proc. of ACM SIGGRAPH*, 29–38.

GAYLE, R., MOSS, W., LIN, M. C., AND MANOCHA, D. 2009. Multi-robot coordination using generalized social potential fields. *Proc. IEEE Conf. Robotics and Automation*.

GUY, S., CHHUGANI, J., KIM, C., SATISH, N., LIN, M. C., MANOCHA, D., AND DUBEY, P. 2009. Clearpath: Highly parallel collision avoidance for multi-agent simulation. *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

HARLOW, F. H. 1963. The particle-in-cell method for numerical simulation of problems in fluid dynamics. In *Proc. Symp. Appl. Math.*, vol. 15.

HEIGEAS, L., LUCIANI, A., THOLLOT, J., AND CASTAGNÉ, N. 2003. A physically-based particle model of emergent crowd behaviors. *Proc. Graphikon '03 2*.

HELBING, D., AND MOLNÁR, P. 1995. Social force model for pedestrian dynamics. *Physical Review E 51* (May), 4282.

HELBING, D., BUZNA, L., JOHANSSON, A., AND WERNER, T. 2005. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Sci. 39*, 1–24.

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

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

KERR, W., AND SPEARS, D. 2005. Robotic simulation of gases for a surveillance task. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (Aug.), 2905–2910.

LAKOBA, T. I., KAUP, D. J., AND FINKELSTEIN, N. M. 2005. Modifications of the Helbing-Molnar-Farkas-Vicsek social force model for pedestrian evolution. *SIMULATION 81*, 339.

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, 509–518.

LIN, M. C., GUY, S., NARAIN, R., SEWALL, J., PATIL, S., CHHUGANI, J., GOLAS, A., DEN BERG, J. V., CURTIS, S., WILKIE, D., MERRELL, P., KIM, C., SATISH, N., DUBEY, P., AND MANOCHA, D. 2009. Interactive modeling, simulation and control of large-scale crowds and traffic. *Proc. Workshop on Motion in Games (Springer-Verlag Lecture Notes in Computer Science Series)*.

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*, 122–129.

MCADAMS, A., SELLE, A., WARD, K., SIFAKIS, E., AND TERAN, J. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph. 28*, 3, 62.

METOYER, R. A., AND HODGINS, J. K. 2003. Reactive pedestrian path following from examples. In *Proc. 16th Int. Conf. Computer Animation and Social Agents*, 149.

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

PARIS, S., PETTRE, J., AND DONIKIAN, S. 2007. Pedestrian reactive navigation for crowd simulation: a predictive approach. *Computer Graphics Forum 26*, 3 (September), 665–674.

PELECHANO, N., O'BRIEN, K., SILVERMAN, B., AND BADLER, N. 2005. Crowd simulation incorporating agent psychological models, roles and communication. *First Intl. Workshop on Crowd Simulation*.

PELECHANO, N., ALLBECK, J. M., AND BADLER, N. I. 2008. *Virtual Crowds: Methods, Simulation and Control.* Morgan and Claypool Publishers.

PETTRÉ, J., LAUMOND, J.-P., AND THALMANN, D. 2005. A navigation graph for real-time crowd animation on multilayered and uneven terrain. *First Intl. Workshop on Crowd Simulation*, 81–90.

PETTRÉ, J., KALLMANN, M., AND LIN, M. C. 2008. Motion planning and autonomy for virtual humans. In *ACM SIGGRAPH 2008 classes*, 1–31.

PIMENTA, K., MICHAEL, N., MESQUITA, R., PEREIRA, G., AND KUMAR, V. 2008. Control of swarms based on hydrodynamic models. *Proc. IEEE Int. Conf. Robotics and Automation* (May), 1948–1953.

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH 21*, 25–34.

REYNOLDS, C. W. 1999. Steering behaviors for autonomous characters. *Game Developers Conference 1999*.

SCHRECKENBERG, M., AND SHARMA, S. D. 2001. *Pedestrian and Evacuation Dynamics*. Springer.

SHAO, W., AND TERZOPOULOS, D. 2007. Autonomous pedestrians. *Graph. Models 69*, 5-6, 246–274.

SHIMIZU, M., ISHIGURO, A., KAWAKATSU, T., MASUBUCHI, Y., AND DOI, M. 2003. Coherent swarming from local interaction by exploiting molecular dynamics and Stokesian dynamics methods. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems 2* (Oct.), 1614–1619 vol.2.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2008. Globally coupled impulse-based collision handling for cloth simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

SUD, A., GAYLE, R., ANDERSEN, E., GUY, S., LIN, M., AND MANOCHA, D. 2007. Real-time navigation of independent agents using adaptive roadmaps. In *Proc. ACM Symp. Virtual Reality Software and Technology*, 99–106.

SUD, A., ANDERSEN, E., CURTIS, S., LIN, M., AND MANOCHA, D. 2008. Real-time path planning in dynamic virtual environments using multi-agent navigation graphs. *IEEE Trans. Visualization and Computer Graphics 14*, 3, 526–538.

SUGIYAMA, Y., NAKAYAMA, A., AND HASEBE, K. 2001. 2-dimensional optimal velocity models for granular flows. In *Pedestrian and Evacuation Dynamics*, 155–160.

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

THALMANN, D., O'SULLIVAN, C., CIECHOMSKI, P., AND DOBBYN, S. 2006. *Populating Virtual Environments with Crowds*. Eurographics 2006 Tutorial Notes.

TREUILLE, A., COOPER, S., AND POPOVIC, Z. 2006. Continuum crowds. *ACM Trans. Graph. 25*, 3, 1160–1168.

VAN DEN BERG, J., LIN, M. C., AND MANOCHA, D. 2008. Reciprocal velocity obstacles for realtime multi-agent navigation. *Proc. IEEE Conf. Robotics and Automation*, 1928–1935.

VAN DEN BERG, J., PATIL, S., SEAWALL, J., MANOCHA, D., AND LIN, M. C. 2008. Interactive navigation of individual agents in crowded environments. *Proc. ACM Symposium on Interactive 3D Graphics and Games*, 139–147.

VAN DEN BERG, J., GUY, S. J., LIN, M. C., AND MANOCHA, D. 2009. Reciprocal n-body collision avoidance. *Proc. Intl. Symposium on Robotics Research*.

YEH, H., CURTIS, S., PATIL, S., VAN DEN BERG, J., MANOCHA, D., AND LIN, M. C. 2008. Composite agents. *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. In *Proc. ACM SIGGRAPH*, 965–972.