Fundamental Diagram Adherent Interactive Crowd Simulation using Density-Dependent Filters

Sahil Narang[1], Andrew Best[1], Sean Curtis[1], Dinesh Manocha[1,*]
[1] **Department of Computer Science, University of North Carolina at Chapel Hill, North Carolina, USA**
* **E-mail: dm@cs.unc.edu**

Abstract

Pedestrian crowds often have been modeled as many-particle system including microscopic multi-agent simulators. One of the key challenges is to unearth governing principles that can model pedestrian movement, and use them to reproduce paths and behaviors that are frequently observed in human crowds. To that effect, we present a novel crowd simulation algorithm that aims to generate pedestrian trajectories that correspond to the speed/density relationships expressed using the Fundamental Diagram. Our approach is based on biomechanical principles and psychological factors. The overall formulation results in better utilization of free space by the pedestrians and can be easily combined with well-known multi-agent simulation techniques with little computational overhead. We are able to generate human-like dense crowd behaviors in large indoor and outdoor environments and validate the results with captured real-world crowd trajectories.

Introduction

The problem of simulating the movement and behaviors of human-like crowds is important in many applications, including architecture and urban design, pedestrian dynamics, computer animation, games, virtual reality, etc. In the absence of governing mathematical models, intuition and observations have driven much of the research and development in this field. A key observation in understanding how individual trajectories are formulated arises from studies in pedestrian dynamics and traffic management that highlight the relationship between crowd density and pedestrian movement; as density increases, speed decreases [1–7]. This phenomenon is called the *Fundamental Diagram* [8].

Although real-world pedestrians tend to slow down in areas of high crowd densities, some of the well known crowd simulation models do not exhibit such behaviors. In this paper, we address the problem of modeling crowd behaviors governed by these densities: which we term as *density-dependent behaviors*. We present a novel approach that combines physiological and psychological principles and effectively generates pedestrian trajectories that adhere to the Fundamental Diagram.

Crowd simulation has been extensively studied and a variety of techniques and models have been proposed to generate plausible, human-like crowd behaviors. While some methods try to model the macroscopic or overall motion of the crowd [9,10], our goal is to accurately simulate microscopic agent-agent and agent-obstacle interactions. Therefore, we propose a multi-agent simulation algorithm that specifically tries to model the trajectory and velocity of each individual agent (or particle) over time.

Many prior multi-agent simulation algorithms decompose the trajectory computation problem into two phases: global planning and local navigation. The global planner computes a path through the environment towards the current goal position of each agent while avoiding collisions with the static obstacles. The local navigation techniques take into account the path computed via the global planner and locally modify the trajectory to avoid collisions with dynamic obstacles or other agents in the environment. Collectively, this class of simulators will be referred to as Global-Local Planners (GLP) and such combinations are frequently used to simulate crowd movement and other emergent behaviors [11, 11, 12, 12–18].

Global trajectory planning includes a number of methods including navigation meshes [19], roadmaps and potential fields [20], etc. These data structures are computed off-line and then used at run-time to generate collision-free paths around static obstacles in the environment. Some work has been done to adapt these structures to dynamic obstacles [21–23]. Other techniques instead aim to model congestion at the global level and compute paths that simply to avoid these dense regions [24–26]. There is extensive literature on local navigation methods: that can be used as part of GLP simulators. They include cellular automata [27], force-based [11–13], rules-based [14], vision-based [15], and velocity-space-based [16–18] techniques. Recent models handle congestion by using global data structures for path computation [28] or physical forces [29].

Some prior crowd simulation algorithms sought to adhere to the fundamental diagram. Lemercier et al. [30] focus on generating realistic following behaviors based on varying densities. Other algorithms tend to explore the density response at a "meso-scale" level between the global and local planners [31, 32] . In each case, the authors apply local planning techniques to adapt an agent's path to the one computed to avoid collisions with distant obstacles. The notion of crowd densities and pedestrian motion has also been studied in other disciplines. Biomechanists have studied both personal space and the human gait at an individual level [1, 2]. There is extensive work in pedestrian dynamics on exploring the relationship between speed and density [4–7, 28]. Our approach builds on many of these ideas.

Main Results

We a novel approach for density modeling that can generate realistic density-dependent behaviors and result in good space utilization. Overall, our approach offers the following benefits:

- The speed and density relationship in the crowd movements computed by our algorithm adheres to the Fundamental Diagram.
- Our approach uses physiological and psychological techniques to model the underlying factors that affect pedestrian movement.
- The density-dependent filters result in smoother trajectories and also reduce the number of collisions between the agents.
- Our approach is general and doesn't make any assumptions about any global planning algorithm or local navigation technique. We have combined our approach with local navigation algorithms based on reciprocal velocity obstacles and social forces.
- The computational overhead of our density-dependent filterw is low and we can simulate hundreds of agents at interactive rates on a single core.

We validate our approach by comparing with captured crowd trajectories and also demonstrate its performance in indoor and outdoor environments.

A preliminary version of this paper appeared in [33]: this is the full version. The rest of the paper is organized as follows: We describe the density-dependent filters and the overall crowd simulation algorithm in Section 2. In Section 3, we compare simulation results with the empirical data. We highlight the performance of our algorithm in Section 4. Finally, in Section 5 we analyze the model's strengths and weaknesses.

Density-dependent Behavior

In this section, we present density-dependent filters and combine them with well-known local navigation algorithms. First, we introduce some of the notation and terminology used in the rest of the paper. In our crowd simulation algorithm, agents are modeled as two-dimensional disks with the following common *state*: $[\ r\ p\ v_c\ v^0]^T \in \mathbb{R}^7$, where r is the radius of the disk, p, v_c , and v^0 are two-dimensional vectors representing the current position, current velocity and the input velocity generated by the global planner, respectively. By convention, v^c and v^0 are the current speed and the input speed (i.e. the magnitudes of the corresponding vectors) generated by the global planner. We use subscripts to denote a particular agent i, such as r_i and v_i^0.

2.1 Fundamental Diagram

The Fundamental Diagram is the observed relationship between pedestrian speed and density; as density increases, speed decreases (Figure 1). Many prior crowd simulation algorithms tend to reduce the speed of the agents through bottlenecks [11,24]. However, this is largely a function of the reduced flow capacity of the environment, rather than the Fundamental Diagram. The Fundamental Diagram does not depend on specific environmental configurations; it corresponds to a relationship between the density and speed.

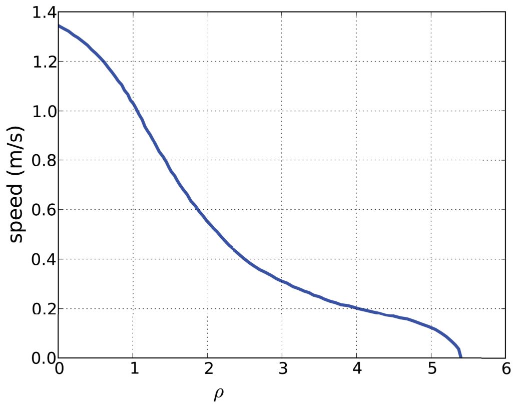

Figure 1. Fundamental Diagram. Empirical relation between pedestrian density and velocity according to Weidmann [8].

2.1.1 Physiological and Psychological Components

Our model incorporates two factors: a physiological and a psychological component. The physiological component is based on the biomechanical principle that stride length and walking speed are inextricably linked; a change in one implies a change in the other [1]. The relationship, as defined by biomechanists [2], is defined by the "stride factor" parameter (α). It relates a person's stride length (L) and walking speed (v) as:

$$L(v) = \frac{H}{\alpha}\sqrt{v}, \tag{1}$$

where $H = \textit{height}/1.72\,\text{m}$ is a height-normalizing constant.

The psychological component models the complex concept of personal space. Psychologists and biomechanists have shown that humans have a strong sense of personal space - a "buffer" that extends beyond mere physical requirements [3,5]. Our model includes a "stride buffer" parameter (β) that reflects this preference for personal space. As the stride buffer increases, an agent requires more space to walk comfortably at a given speed.

We use a linear combination of the physiological and psychological components to yield a function that determines the natural walking speed (V) for the available space (S) in front of an agent:

$$V(S) = \max\left(v^0, \left(\frac{S\alpha}{H(1+\beta)}\right)^2\right), \tag{2}$$

where α is the stride factor as defined by [2] and β is the stride buffer.

Space-estimation: The relationship in Eq. 2 is expressed in terms of the space on a line extending in a single direction. However, our agents move on a 2D plane, and we need to compute a similar mapping in a 2D region that comprises of other agents and obstacles. We base our method on some experimental observations performed by Seyfried et al. [4].

Seyfried et al. performed one dimensional experiments on the Fundamental Diagram by creating a narrow passage which constrained the subjects to walk in a line. The subjects' changes in speed are thus attributed to the available space in front and behind the agent, along the one-dimensional path. The density in this experiment was, likewise, a one-dimensional quantity (number of pedestrians/meter). The authors observed that the 1D density in the experiment was related to the 2D expectations defined by Weidmann [8] by a scale factor equal to the inverse of the individual's width. Given the average human width of 0.48 m, a measured 1D density value of 1 people/m translates to a 2D density value of 2.08 people/m^2. We employ this observation in our model of space estimation by first computing the local 2D density and then using Seyfried et. al.'s method to map it to the scalar "space" value used in Eq. (2).

2.2 Density Dependent Filters

We introduce density-dependent filters that can be used to generate human-like density responses. These filters can be integrated with any Global-Local Planner (GLP) based model.

GLP based models use a global planner to plan a path through the static environment. The path is used to generate immediate goals which are communicated to the local planner as *preferred velocities* - velocities in the direction of an immediate goal, at a user-defined *preferred speed.* The local navigation scheme selects a viable velocity based on the preferred velocity and the local dynamic conditions (i.e. nearby obstacles and agents). In most crowd simulation algorithms based on the GLP paradigm, the preferred velocity serves as the interface between the global and local modules.

Implicit to this model is the assumption that the agent's global plan is independent of the local conditions. This assumption places a particular burden on the local navigation module; it must be able to compute an appropriate response under an arbitrarily large range of local conditions. This burden can lead to problematic results: noisy trajectories, increased agent-agent collisions and undesirable agent behavior.

Our approach seeks to reduce the mismatch between the two stages by seamlessly adapting the preferred velocities generated by the global planner to local dynamic conditions. Architecturally, the density-dependent filter serves as the interface between the global planner and local navigation module, as shown in Figure 2. This results in smoother trajectories, reduced agent-agent collisions and realistic agent behavior.

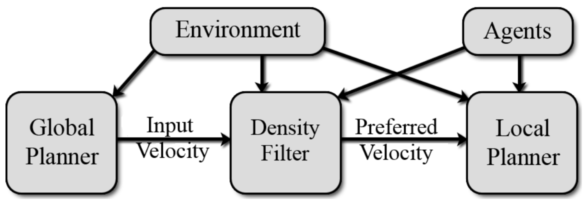

Figure 2. System Architecture. Our density filters act as an interface between the global and local planner.

The agents tend to optimize their progress towards the goal by exploring the space around them, while their motion is governed by the physiological and psychological factors. At each step, our algorithm computes an arc centered at the input velocity from the global planner. It chooses a velocity from this arc that minimizes the cost function (Eq. 9) while respecting the Fundamental Diagram constraints. Finally, the density filter communicates this velocity to the local planner as the new preferred velocity for collision avoidance.

2.2.1 Agent Density Computation

Given an agent i and direction vector $\hat{v}_i^\theta$ offset from the agent's input velocity v_i^0 by an angle θ, we define a point q_i^θ that lies one meter ahead along $\hat{v}_i^\theta$. We use a Gaussian density function to determine the contribution of each neighboring agent to the density at q_i^θ. It is well known that humans exhibit an elliptical personal space, with a preference for space in front [3]. To model the personal space of agent i, we transform the relative position of i's neighbors anisotropically, causing agents in front to have a greater contribution to density than those to the side (Eq. (5) and (6)). Thus, the density of neighboring agents at q_i^θ is:

$$\begin{aligned}
\rho_{Ai}^\theta &= \sum_j \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{\|d'_{ij_\theta}\|^2}{2\sigma^2}} \qquad j \neq i, & (3)\\
q_i^\theta &= p_i + \hat{v}_i^\theta, & (4)\\
d'_{ij_\theta} &= 2.5(d_{ij} - d_{ij_\theta}^y) + d_{ij_\theta}^y, & (5)\\
d_{ij_\theta}^y &= (d_{ij} \cdot \hat{v}_i^\theta)\hat{v}_i^\theta. & (6)
\end{aligned}$$

Effect of Obstacles on Density Computation

The presence of obstacles restricts the space available (S) for an agent, as given by Eq. 2. This implies that the effect of obstacles on density relates inversely to the amount of contiguous free space available to the agent. Given a point q, we represent this bias using a normalized 2D Gaussian kernel centered at q. The 2D kernel prioritizes space nearest q, increasing the agent's sensitivity to immediately proximate obstacles. The kernel is simply a weight function with compact support (w) and integrating it over the domain of the support (Ω) yields the contiguous free space (FS) available to an agent at q.

$$FS_q = \int_\Omega w(\vec{x}) \tag{7}$$

The effective density used in our algorithm for a point q_i^θ is the agent density ρ_{Ai}^θ scaled by the inverse of the free space available at q_i^θ.

$$\rho_i^\theta = \frac{1}{FS(q_i^\theta)} \rho_{Ai}^\theta \tag{8}$$

2.2.2 Agent's Response to Density

An agent i with width w_i chooses θ such that it minimizes the distance to the goal g_i with respect to time period τ (Figure 3)

$$\arg\min_\theta \| g_i - (p_i + v_i^{FD_\theta})\tau\|, \tag{9}$$

where $v_i^{FD_\theta}$ represents the Fundamental Diagram adherent velocity which can be computed as:

$$v_i^{FD_\theta} = \frac{v_i^\theta}{\|v_i^\theta\|} V(\rho_i^\theta / w_i). \tag{10}$$

Finally, $v_i^{FD_\theta}$ is set to be the new preferred velocity.

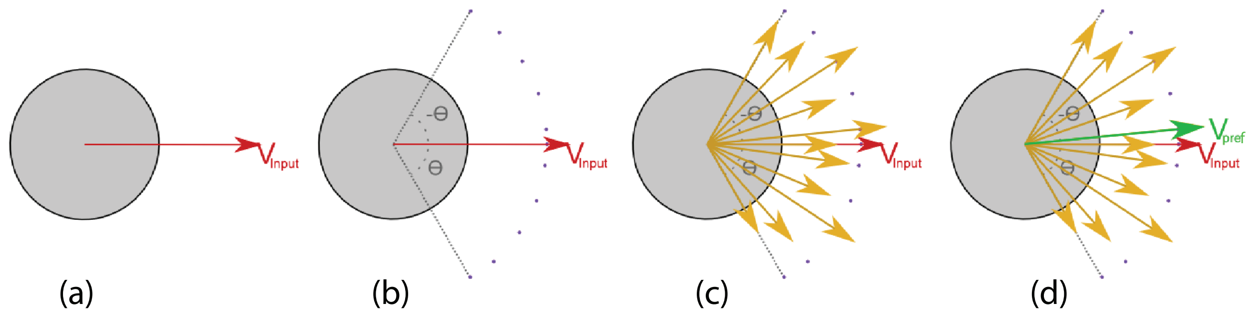

Figure 3. Sampling Mechanism. (a) The input velocity V_{input} given by the global planner. (b) The agent considers density at critical points along an arc defined by $[-\theta,\, \theta]$. (c) It computes the filtered velocities for each critical point along the arc. (c) The velocity which minimizes the cost function, V_{pref}, is chosen as the preferred velocity.

2.3 Implementation

We represent the obstacles over a grid of cell size c_b. The simulator identifies cells lying inside the obstacle and generates a level set. Next, it computes the free space available at each cell c_p that is not contained within any obstacle. It does so by splatting a precomputed 2D Gaussian kernel at point p, the center of cell c_p and then systematically walking around the neighborhood of c_p to determine contiguous free space with respect to p (Figure 4). Hence, for a scene with n obstacles, we can find the obstacles that contribute to "density computation" at a point p in $O(n)$ time. This can all be done offline thereby saving runtime costs.

During simulation, each agent computes the density at a critical point one meter ahead in the direction of the input velocity v_i using Eq. 8. The agent determines whether the density at that point is more than a preset threshold. If so, the agent samples the arc mentioned above at discrete angles θ in $[-\theta_{max}, \theta_{max}]$ (Figure 3). For each sample point, it computes the density and the corresponding Fundamental Diagram adherent velocity (Eq. 8 and 10). Finally, it optimizes the input velocity v_i using Eq. 9.

2.4 Models Augmented with Density Dependent Filters

Our use of density-dependent filters is orthogonal to the choice of global and local planners. To illustrate this generality, as well as our algorithm's efficacy, we have applied it to several local planners. The simulators all share the same global planner, which is a navigation mesh. The shared global planner means the differences between simulators will be dominated by the local planners. In particular, we demonstrate how our approach can be combined with two widely use local navigation algorithms: social forces [11] and optimal reciprocal collision avoidance [17].

For a given input velocity v_i^0 for agent i generated by the navigation mesh, our algorithm computes the optimal Fundamental Diagram adherent preferred velocity $v_i^{FD_\theta}$ using Eq. 10, where θ is found using the optimization problem given by Eq. 9. This velocity is then communicated to the local navigation module for collision avoidance.

2.4.1 Social Forces

The social force (SF) model treats the crowd as a collection of mass particles. Newtonian-like physics is applied to the system to compute the trajectories of the agents. At each time step, the superposition of various forces are computed for each agent, ultimately determining a feasible velocity by imparting

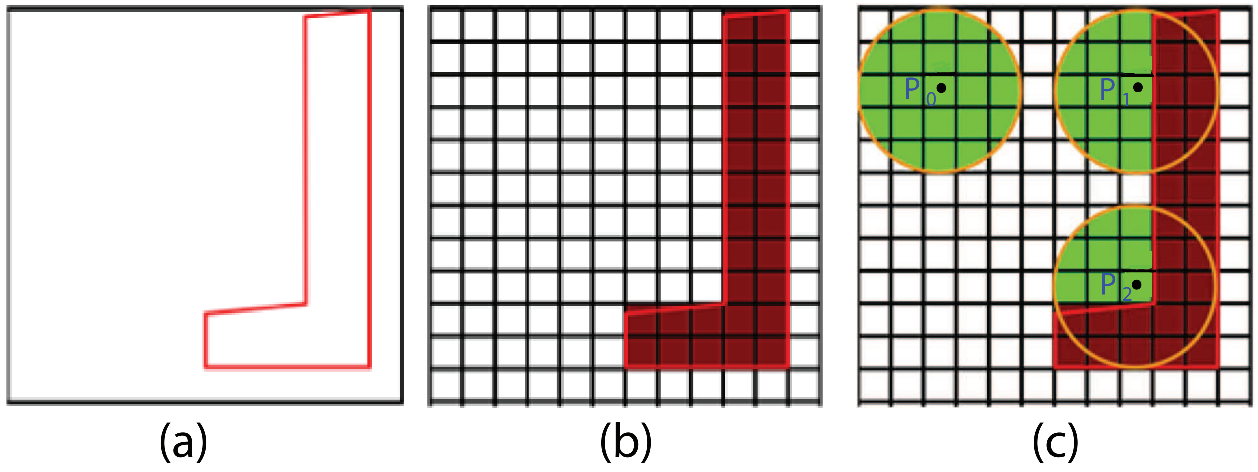

Figure 4. Computing free space at a point. (a) Scene with obstacle (b) Discretized scene (c) Contiguous free space (in green) available to agents centered at points p_0, p_1 and p_2.

acceleration on the agent. The preferred velocity is converted into a "driving force". The agent avoids collisions with other agents and obstacles through the application of repulsive forces. We have taken the exact formulation of these forces, including parameter values, from Helbing's work [11].

Social Forces with Density Dependent Filters (D-SF): An agent i under D-SF feels a driving force F_{drive} as:

$$F_{drive} = m_i \frac{v_i^{FD_\theta} - v_i^{new}}{\tau_i}, \tag{11}$$

where m_i is the mass of the agent, τ_i is the time step and v_i^{new} is the new velocity for agent i. v_i^{new} can be computed using the force equation:

$$m_i \frac{dv_i^{new}}{dt} = F_{drive} + \sum_{j \neq i} f_{ij} + \sum_W f_{iW}, \tag{12}$$

where f_{ij} represents the agent-agent repulsive force and f_{iW} the agent-obstacle repulsive force.

2.4.2 Optimal Reciprocal Collision Avoidance

Optimal Reciprocal Collision Avoidance (ORCA) [17] applies geometric optimization techniques in velocity space. It is based on the notion of velocity obstacles from robotics. Avoiding collision simply requires selecting a velocity which does not lie within the reciprocal velocity obstacle set for each agent. ORCA's unique formulation defines the velocity obstacles as half planes. The set of velocities that are permitted for agent i with respect to all other agents is the intersection of the half planes of permitted velocities induced by each other agent, and we denote this set $ORCA_i^\tau$.

ORCA with Density Dependent Filters (D-ORCA): An agent under D-ORCA selects a new velocity v_i^{new} for itself that is closest to its preferred velocity $v_i^{FD_\theta}$ amongst all velocities inside the region of permitted velocities:

$$v_i^{new} = \arg \min_{v \in ORCA_i^\tau} \| v - v_i^{FD_\theta} \| \tag{13}$$

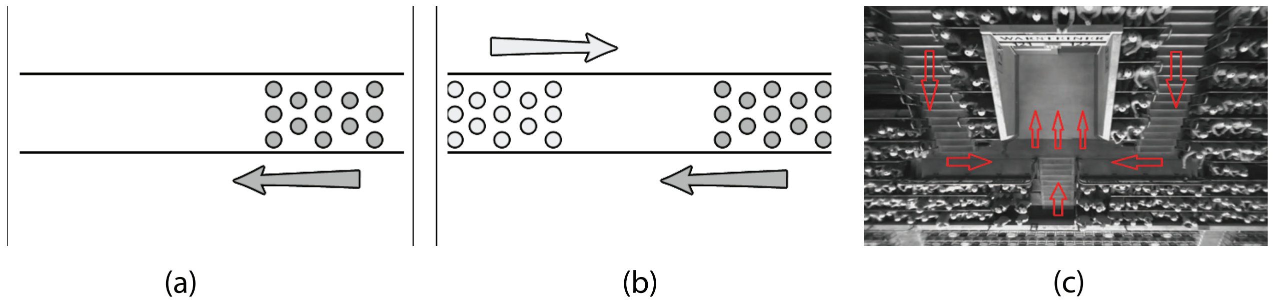

Figure 5. Benchmark scenarios used to validate our density dependent filters. (a) The uni-directional flow experiment in [6] (c) The bi-directional flow experiment in [7] and (c) The stadium experiment [34] with red arrowheads denoting the path to the exit tunnel.

Comparison to Real-World Data

We validate the performance of our model in three reproductions of live pedestrian experiments (Figure 5). These real world experiments examine the flow of pedestrians in various scenarios: a 2D uni-directional flow scenario, a bidirectional flow scenario, and a complex flow scenario of people exiting a stadium. For each experiment, we report adherence to the Fundamental Diagram.

We measure the simulator's ability to reproduce the Fundamental Diagram in the following manner. For each experiment, we define a region through which each agent must pass. We compute the time it takes for each agent to pass through the region and the average density of the agents in that region during that interval. This becomes a single density-speed data pair, which we present in a scatter plot (Figure 6). This analysis is similar to the ones use in pedestrian dynamics literature [4–7].

When the flow of agents is reduced because the capacity of the area is limited, congestion increases and the agents slow down. It is worth noting that both SF and ORCA agents exhibit such behavior. However, this is not a manifestation of the Fundamental Diagram. The Fundamental Diagram doesn't require changes in the capacity of a space. Simply increasing density should cause a reduction in speed.

3.1 Two Dimensional, Uni-Directional Flow

Zhang et al. [6] performed experiments of uni-directional flow in a corridor (Figure 5(a)). The authors performed multiple iterations of the experiment, varying the flow into the corridor and the flow out of the corridor to control the observed density in the measurement region inside the corridor. Reproducing this experiment tests the suitability of the space-estimation formulation described earlier.

Figure 6(a) shows the Fundamental Diagram results for ORCA, D-ORCA, SF and D-SF. Its easy to see that both ORCA and SF marginally reduce speeds at high densities. This is due to the constraints on the flow out of the corridor. The experiment created a bottleneck; the flow into the corridor was greater than the flow out of the corridor. However, D-ORCA and D-SF agents reduce speeds due to increase in density which is in accordance with the Fundamental Diagram and closely matches the captured human data.

3.2 Two-dimensional, Bi-directional Flow

Following the uni-directional flow experiment, Zhang et al. also examined bi-directional flow [7]. Using the same arrangement as in the previous uni-directional flow experiment, the authors placed subjects at both ends of the corridor. In our bi-directional experiment, we use an infinitely long corridor, of the same width in the pedestrian experiment. We vary the flow into the corridor by varying the initial agent

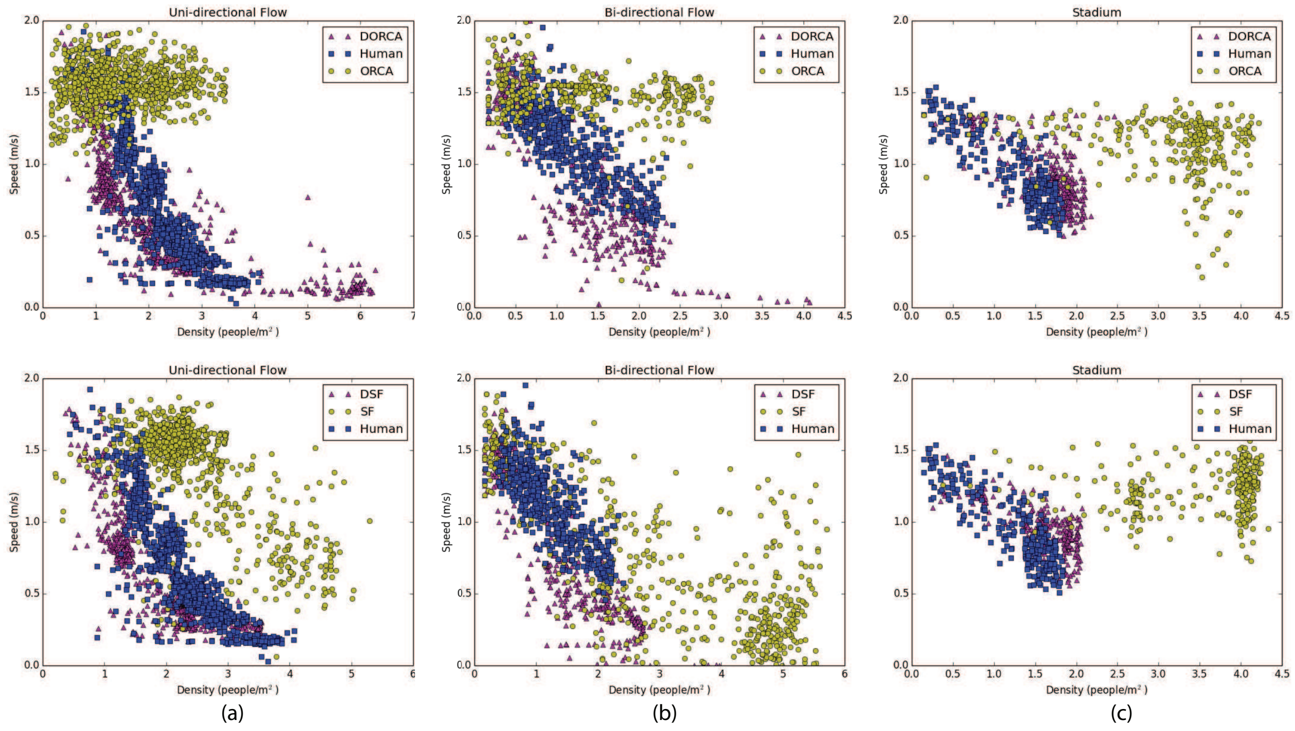

Figure 6. Fundamental Diagram for SF, D-SF, ORCA, D-ORCA for the three experiments. (a) The uni-directional 2D flow experiment in [6] (b) The bi-directional flow experiment in [7] and (c) The stadium experiment [34]. Agents in both D-ORCA and D-SF exhibit a human-like sensitivity to high densities and conform to the Fundamental Diagram.

density (ranging from 0.5–2.5 people/m^2). The Fundamental Diagram results can be seen in Figure 6(b). SF showed a consistent reduction in speed with respect to density. In contrast, ORCA exhibited relatively constant speeds irrespective of density. However, with the application of the density filters, agents were able to maintain a sparse distribution, which appears to be consistent with observed human-behaviors.

3.3 Stadium

Seyfried et. al. [34] performed experiments on the complex flow of a crowd exiting a soccer stadium. They recorded the trajectories of 300 spectators as they made their way toward a predetermined exit tunnel. The stadium is difficult to simulate because of its non planar layout; furthermore, there are three dense flows that meet almost orthogonally at the entrance of the tunnel which leads to a bottleneck. Figure 6(c) depicts the Fundamental Diagram, computed at the entrance of the tunnel, for this dataset. It is easy to see the success of the density filters at effectively modeling the highly dense and convoluted flows that make up the stadium dataset. In contrast, both ORCA and SF agents achieve significantly high densities and exhibit little or no response to density.

D-SF's response to density is further evident from the density field for captured pedestrian crowd (Figure 7(a)), D-SF (Figure 7(b)), and SF (Figure 7(c)) in the exit tunnel at the moment when 20% of the agents have entered the tunnel. SF agents achieve densities as high as 5.93 people/m^2 whereas D-SF agents achieve a maximum value of 3.96 people/m^2, more in keeping with captured human data.

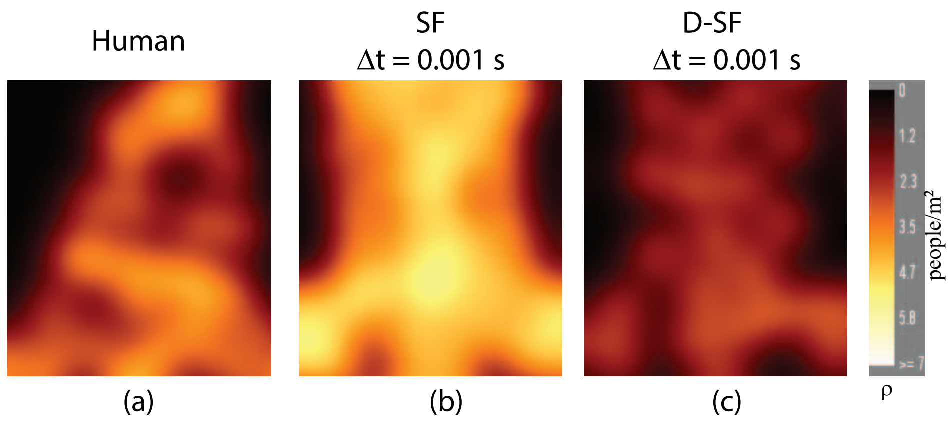

Figure 7. Density visualization for stadium experiment [34]. a) Captured pedestrian crowd b) SF and c) D-SF at the moment when 20% of the agents have entered the exit tunnel. D-SF agents maintain a sparse distribution, similar to captured pedestrian crowd while SF agents achieve densities as high as 5.93 people/m^2.

Results

In this section, we highlight the performance of our algorithm on different scenarios. In addition to generating density-dependent behaviors, our formulation also results in smoother trajectories and results in fewer agent-agent collisions as compared to prior methods.

4.1 Applications

We tested the capabilities of our model on a number of example scenarios, depicted in Figure 8. In one scenario, roughly depicting an hourglass, 200 agents stream in towards a narrow passage with concave windings (Figure 8(b)). Agents under D-SF and D-ORCA exhibit significantly fewer collisions than their SF and ORCA counterparts. In another scenario, agents move in concentric circles towards a ticket kiosk at the center (Figure 8(c)). They are programmed to wait for 10 seconds at the kiosk before heading outward again. This creates a high density situation since we have a large number of agents heading towards each other with opposing goals. Agents under D-SF and D-ORCA exhibit smoother trajectories and are able to reach their goals faster as compared to SF and ORCA respectively. Finally, we depict the scene outside a store on 'Black Friday' (Figure 8(d)). One thousand agents queue up outside the store waiting for the doors to open. Once opened, agents stream in and move towards the other end of the store. D-SF and D-ORCA agents exhibit a tendency to maintain personal space and to this end, utilize the free space around them. This translates to smoother trajectories and few collisions as compared to SF and ORCA respectively.

It is worth noting that we used the same sampling parameters for all our experiments. Furthermore, with the obstacle grid computed offline, the run time overhead of our algorithm was minimal, as can be in Table 1. In most cases, the introduction of the density filters increased the average frame computation time by a few milliseconds. However, in case of the Kiosk, D-SF actually reduced the computation time. This is due to the significant decrease in collisions (Table 2). The experiment created cross flows leading to regions of high density. SF agents attempted to achieve their preferred speed even in high densities, leading to collisions and thus increasing the time spent in collision detection. On the other hand, D-SF agents were able to maintain a sparse distribution and thus had fewer collisions.

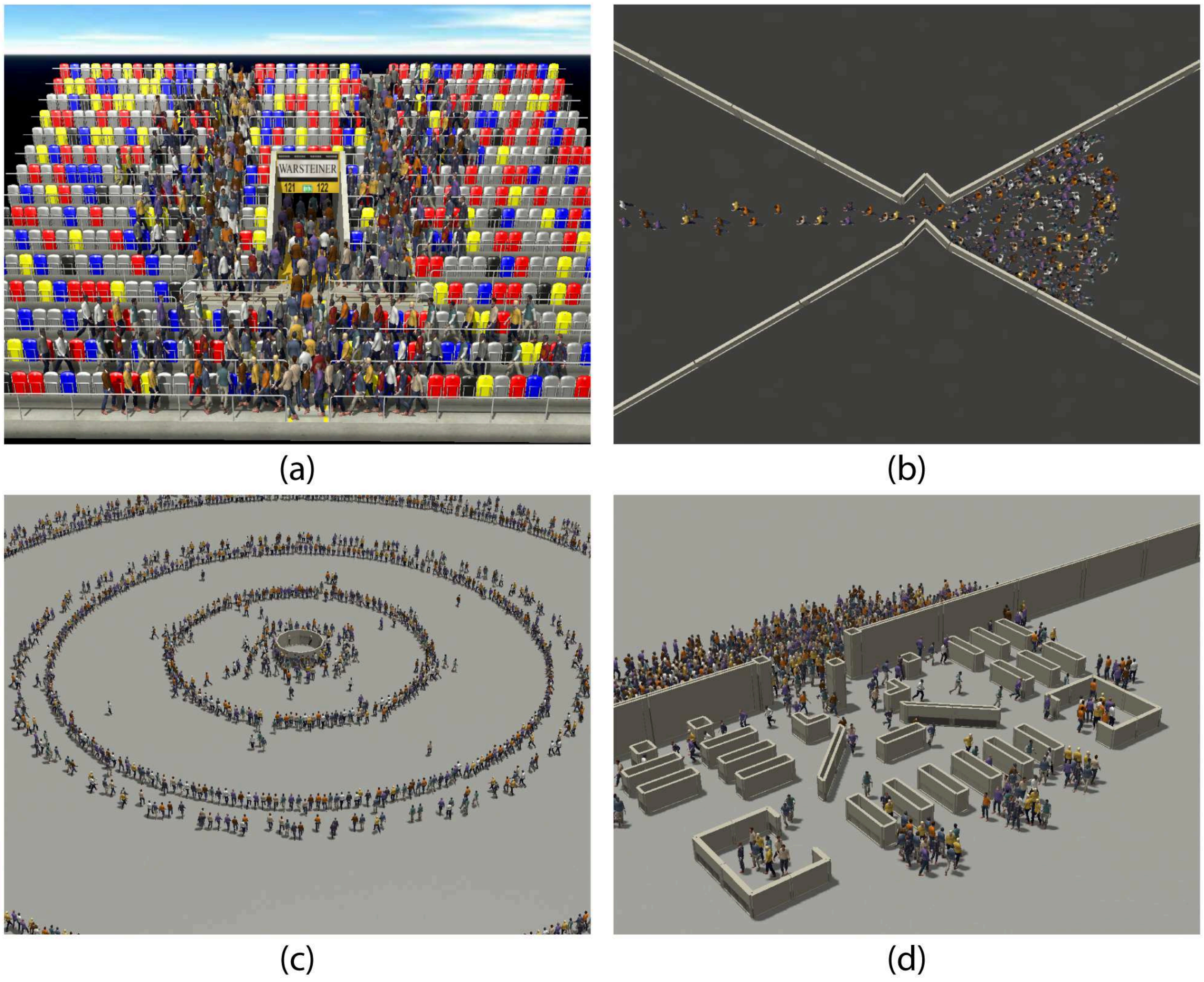

Figure 8. Application scenarios with D-ORCA agents. a) Stadium b) Hourglass c) Ticket Kiosk and d) Shopping center on Black Friday.

4.2 Trajectory Smoothness

Without the density filter, the trajectories exhibit a great deal of noise. The reason for this is that the unfiltered intention (preferred speed) might be inconsistent with the agents current condition; moving toward the goal at preferred speed may not be possible. But given that intention, the local navigation algorithm aggressively seeks to exploit every opportunity for increasing the agent's speed to match its preferred speed. However, such rapid accelerations can often leave the agent in an untenable situation which needs to be reversed in the very next time step. The introduction of the density filter causes the intended speed to scale down according to the congestion. Because of this, the acceleration taken is more likely to be valid over a longer window and the motion of the agent becomes smoother.

We illustrate our model's impact on trajectory smoothness by plotting trajectories of a small sample of agents (Figure 9). We performed a baseline simulation using SF with a time-step of 0.0625 s (middle).

Table 1. Average frame computation time.

Scenario	**Agents**	**ORCA (ms)**		**SF (ms)**	
		ORCA	D-ORCA	SF	D-SF
Stadium	302	2.10	13.27	2.32	15.02
Hourglass	200	1.15	12.31	1.01	12.07
Kiosk	1830	17.05	71.51	33.81	10.17
BlackFriday	980	11.55	111.49	11.29	139.70

Application of the density filters only marginally increases the frame computation time. In case of the Kiosk, it actually reduced the computation time due to a significant decrease in collisions.

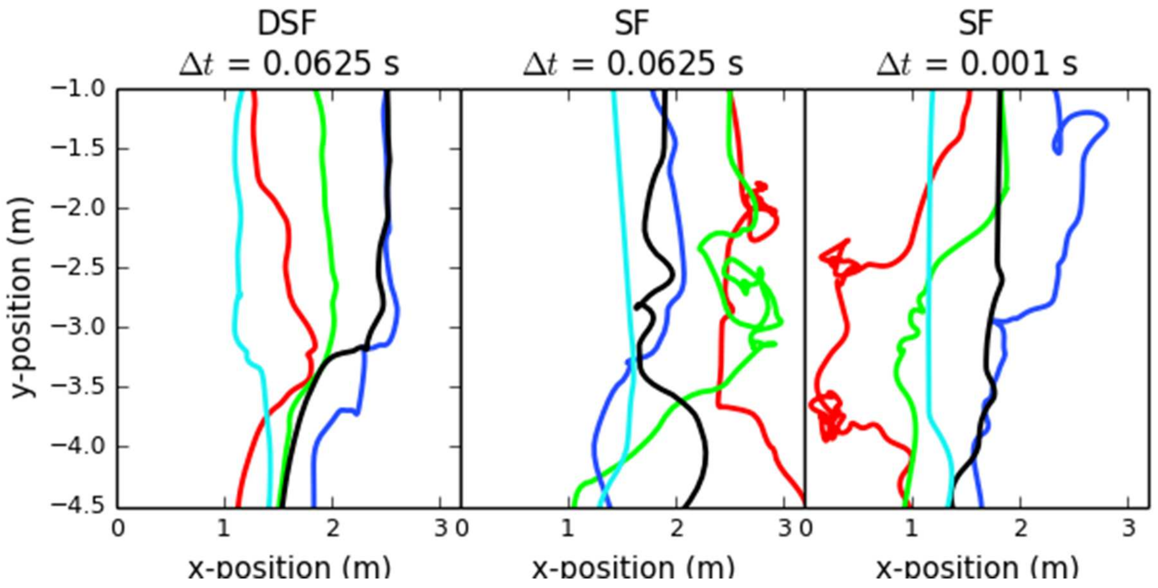

Figure 9. Impact of density filters on trajectory smoothness. Trajectories of five sampled SF agents under three simulation paradigms: with D-SF at 16 Hz (left), SF at 16 Hz (middle), and SF at 1000 Hz (right). D-SF exhibited the smoothest trajectories.

In addition, we simulated D-SF at the same time step (left) and SF at a 0.0001 s time step (right). The baseline trajectories are chaotic. Reducing the time step by a factor of 62.5 improved smoothness, but introducing the density filter produced the smoothest trajectories, even at the larger time step.

4.3 Inter-agent Collisions

Finally, models augmented with density-dependent filters reduce the number and degree of collisions in the simulation. By a collision, we mean the overlap or penetration of one agent into the geometric shape corresponding to another agent. We introduce a new metric for measuring the rate of collisions in a crowd simulation: *interval penetration depth.* It combines the concepts of penetration depth and continuous collision detection.

Penetration depth is the measure of how much two entities overlap; it is typically defined as the minimum displacement required to eliminate overlap. Continuous collision detection detects collisions, not just at discrete time steps, but at the intervals between them. For each pair of adjacent time steps, we compute the maximum penetration depth between two agents, i and j, over the interval bound by those time steps t and $t+1$. Penetration depth allows us to quantify the *severity* of collision and, by computing it in a continuous fashion, we can catch collisions that might otherwise be missed.

We assume the agents move linearly between time steps. The position of an agent over an interval is $p_i(t) = p_i + v_i t, 0 \leq t \leq \Delta t$. The maximum penetration depth can be computed by finding the minimum distance, or, equivalently, the minimum squared distance, between two agents, i and j, over the time interval $[\ t_i,\ t_{i+1}]$. Finally, we produce a collision score for a full simulation by computing the average penetration depth across all frames and agents: Table 2 reports the collision scores for the scenarios, with and without the density filters. The order of magnitude difference in logarithmic scale with the application of the density filters is high as ten for both ORCA and SF. The "Bidirectional flow" in case of ORCA and "Hourglass" scenario in case of SF witnessed the most improvement which is understandable since these scenarios were designed to have the most cross flow.

The fact that our density filter reduces collisions should not be surprising. Many of the collisions arise from the jostling due to non-smooth trajectories in dense scenarios. Smoother trajectories naturally lead to fewer collisions.

Table 2. The impact of the density filters on collisions.

Scenario	**ORCA**		**SF**	
	ORCA	D-ORCA	SF	D-SF
Uni-dir.	0.07	1.5E-03	0.181	0
Bi-dir.	0.526	0	0.52	3.07E-04
Stadium	0.112	9.6E-03	0.233	7.3E-05
Hourglass	0.841	4.19E-04	0.921	0
Kiosk	1.447	0.508	4.681	4.1E-03
BlackFriday	1.327	0.112	11.051	6.4E-03

Collision scores were computed using the interval penetration depth metric, described in 4.3. Application of the density filters considerably reduces the collision rate for both simulators.

Conclusion

We have introduced a novel crowd simulation algorithm based on density-dependent filters to generate human-like crowd flows. Our approach is applicable to a large number of GLP multi-agent algorithms that use a combination of local and global planners, and can generate realistic density-dependent behaviors. We have highlighted the performance on many complex scenarios and also validated the performance with captured human trajectories. The computational overhead of our approach is relatively small, and it is shown to generate smoother trajectories with fewer collisions. Furthermore, the introduction of the density filters requires no significant changes to either the global or local navigation components.

5.1 Limitations and Future Work

The potential impact of the density filters depends on how sensitive the local planner is to the preferred velocity. In both SF and ORCA, the preferred velocity plays a significant role and this results in improved performance of D-SF and D-ORCA. Also, the benefit of the density filters may be relatively less in scenarios where global density-dependent navigation techniques are used. The fact that the adaptation is done at a local level implies that the density filters may prove ineffective in scenarios where global density-dependent navigation techniques are more appropriate.

In future work, we will investigate the application of the density filters at the meso-scale so as to allow agents to account for sudden changes in density. Furthermore, we plan to augment our density filters with other psychological factors which may play a vital role in modeling dense heterogeneous crowds.

Acknowledgments

This work was supported by NSF awards 1000579, 1117127, 1305286, Intel, and a grant from the Boeing Company. We would like to thank the Julich Supercomputing Institute for their invaluable help in graciously providing their collected data and supplying a definition for the simulation domain. We would also like to thank Benjamin Chrtien for help with implementation.

References

1. Inman VT, Ralston HJ, Todd F, Lieberman JC (1981) Human Walking. Williams & Wilkins.
2. Dean GA (1965) An analysis of the energy expenditure in level and grade walking. Ergonomics 8: 31–47.
3. Gérin-Lajoie M, Richards CL, McFadyen BJ (2005) The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space. Motor Control 9: 242–269.
4. Seyfried A, Steffen B, Klingsch W, Boltes M (2005) The fundamental diagram of pedestrian movement revisited. J Stat Mech .
5. Chattaraj U, Seyfried A, Chakroborty P (2009) Comparison of pedestrian fundamental diagram across cultures. Advances in Complex Systems 12: 393–405.
6. Zhang J, Klingsch W, Schadschneider A, Seyfried A (2011) Transitions in pedestrian fundamental diagrams of straight corridors and t-junctions. J Stat Mech 2011: P06004.
7. Zhang J, Klingsch W, Schadschneider A, Seyfried A (2012) Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. J Stat Mech : P02002.
8. Weidmann U (1993) Transporttechnik der Fussgaenger. Schriftenreihe des IVT.
9. Narain R, Golas A, Curtis S, Lin MC (2009) Aggregate dynamics for dense crowd simulation. ACM Trans Graph 28: 122:1–122:8.
10. Treuille A, Cooper S, Popović Z (2006) Continuum crowds. In: ACM SIGGRAPH 2006. ACM, pp. 1160–1168.
11. Helbing D, Farkas I, Vicsek T (2000) Simulating dynamical features of escape panic. Nature 407: 487–490.
12. Chraibi M, Seyfried A, Schadschneider A (2010) Generalized centrifugal-force model for pedestrian dynamics. Phys Rev E 82: 046111.
13. Pelechano N, Allbeck JM, Badler NI (2007) Controlling individual agents in high-density crowd simulation. In: SCA '07. pp. 99–108.
14. Reynolds CW (1999) Steering behaviors for autonomous characters. Game Developers Conference .

15. Ondřej J, Pettré J, Olivier AH, Donikian S (2010) A synthetic-vision based steering approach for crowd simulation. In: Proc. SIGGRAPH. pp. 123:1–123:9.

16. Van den Berg J, Patil S, Sewall J, Manocha D, Lin M (2008) Interactive navigation of multiple agents in crowded environments. In: I3D '08. pp. 139–147.

17. Van den Berg J, Guy SJ, Lin M, Manocha D (2009) Reciprocal n-body collision avoidance. In: Inter. Symp. on Robotics Research.

18. Pettré J, Ondřej J, Olivier AH, Cretual A, Donikian S (2009) Experiment-based modeling, simulation and validation of interactions between virtual walkers. In: SCA '09. ACM, pp. 189–198.

19. Snook G (2000) Simplified 3D movement and pathfinding using navigation meshes. In: Game Programming Gems, Hingham, Mass.: Charles River, chapter 3. pp. 288–304.

20. LaValle S (2006) Planning Algorithms. Cambridge: Cambridge.

21. Jaillet L, Simeon T (2004) A PRM-based motion planning for dynamically changing environments. In: Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst. volume 2, pp. 1606–1611.

22. Sud A, Gayle R, Andersen E, Guy S, Lin M, et al. (2007) Real-time navigation of independent agents using adaptive roadmaps. In: Proc. ACM Symp. Virtual Real. Softw. Tech. pp. 99–106.

23. van Toll WG, Cook AF, Geraerts R (2012) A navigation mesh for dynamic environments. CASA .

24. Guy SJ, Chhugani J, Curtis S, Lin MC, Dubey P, et al. (2010) Pledestrians: A least-effort approach to crowd simulation. In: Symposium on Computer Animation. ACM.

25. Kretz T, Hengst S, Roca V, Pérez Arias A, Friedberger S, et al. (2011) Calibrating Dynamic Pedestrian Route Choice with an Extended Range Telepresence System. In: 2011 IEEE International Conference on Computer Vision Workshops. pp. 166–172.

26. van Toll WG, Cook AF, Geraerts R (2012) Real-time density-based crowd simulation. Computer Animation and Virtual Worlds 23: 59–69.

27. Schadschneider A (2001) Cellular automaton approach to pedestrian dynamics - theory. Pedestrian and Evacuation Dynamics .

28. Curtis S, Manocha D (2012) Pedestrian simulation using geometric reasoning in velocity space. In: Pedestrian and Evacuation Dynamics.

29. Kim S, Guy SJ, Manocha D (2013) Velocity-based modeling of physical interactions in multi-agent simulations. In: Symposium on Computer Animation. pp. 125-133.

30. Lemercier S, Jelic A, Kulpa R, Hua J, Fehrenbach J, et al. (2012) Realistic following behaviors for crowd simulation. Comput Graph Forum 31: 489-498.

31. He L, van den Berg J (2013) Meso-scale planning for multi-agent navigation. In: ICRA. pp. 2839-2844.

32. Golas A, Narain R, Lin MC (2013) Hybrid long-range collision avoidance for crowd simulation. In: ACM Symposium on Interactive 3D Graphics. pp. 29-36.

33. Best A, Narang S, Curtis S, Manocha D (2014) Densesense: Interactive crowd simulation using density-dependent filters. In: Symposium on Computer Animation. pp. 97-102.

34. Burghardt S, Klingsch W, Seyfried A (2012) Analysis of flow-influencing factors in mouths of grandstands. In: Pedestrian and Evacuation Dynamics PED.