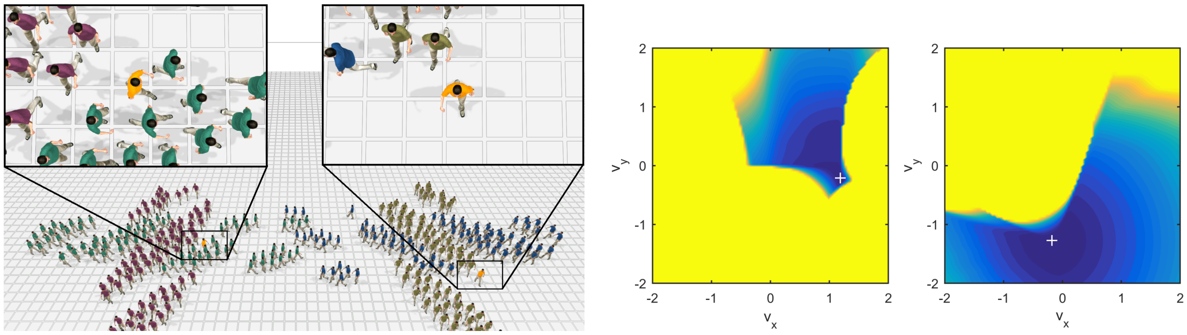
Implicit Crowds: Optimization Integrator for Robust Crowd Simulation

IOANNIS KARAMOUZAS, Clemson University
NICK SOHRE, University of Minnesota
RAHUL NARAIN, University of Minnesota
STEPHEN J. GUY, University of Minnesota



(a)

(b) Left inset

(c) Right Inset

Fig. 1. (a) A challenging crowd simulation scenario with multiple flows interacting simultaneously. By reformulating implicit integration to allow velocity-based energy functions, we can enable numerically stable crowd simulations even in these dense multi-directional interactions. Our method leads to smooth, collision-free motion for a wide range of time steps sizes. (b-c) We perform implicit integration by minimizing a global energy function, which is visualized here by taking slices with respect to the velocities of the two highlighted agents. White crosses indicate optimal new velocities for the highlighted agents.

Large multi-agent systems such as crowds involve inter-agent interactions that are typically anticipatory in nature, depending strongly on both the positions and the velocities of agents. We show how the nonlinear, anticipatory forces seen in multi-agent systems can be made compatible with recent work on energy-based formulations in physics-based animation, and propose a simple and effective optimization-based integration scheme for implicit integration of such systems. We apply this approach to crowd simulation by using a state-of-the-art model derived from a recent analysis of human crowd data, and adapting it to our framework. Our approach provides, for the first time, guaranteed collision-free motion while simultaneously maintaining high-quality collective behavior in a way that is insensitive to simulation parameters such as time step size and crowd density. These benefits are demonstrated through simulation results on various challenging scenarios and validation against real-world crowd data.

CCS Concepts: •**Computing methodologies** → *Physical simulation; Continuous simulation;*

Additional Key Words and Phrases: Crowd simulation, implicit integration, physics-based animation

ACM Reference format:
Ioannis Karamouzas, Nick Sohre, Rahul Narain, and Stephen J. Guy. 2017. Implicit Crowds: Optimization Integrator for Robust Crowd Simulation. *ACM Trans. Graph.* 36, 4, Article 136 (July 2017), 13 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073705

1 INTRODUCTION

Simulating the motion of multiple intelligent agents interacting with each other, such as in crowds, flocks, or traffic, is an important task in computer animation. In many different fields which study human motion, early methods for simulation used force-based models [Helbing and Molnár 1995; Reynolds 1987] inspired by physics, and could take advantage of improvements in physics-based animation. Over the past years, both multi-agent modeling and physics-based animation have seen many powerful advances, leading to highly sophisticated techniques in both fields. However, as the two fields have largely developed independently, they have diverged enough that recent numerical techniques from physics-based animation cannot directly be applied to modern multi-agent models. In this work, we seek to remove this barrier by building a new connection between the two fields, enabling robust and efficient simulation of intelligent, anticipatory agent behavior.

A key advancement in physics-based animation has been the development of numerically robust simulation techniques, which give consistent and stable results across a variety of scenarios, simulation conditions, and even time step sizes. In particular, implicit integration schemes such as backward Euler are exceptionally stable even

This work has been supported in part by the National Science Foundation, under grants CHS-1526693 and CNS-1544887.
Rahul Narain and Stephen J. Guy are joint last authors.
Author's addresses: I. Karamouzas, School of Computing, Clemson University, Clemson, SC; email: ioannis@clemson.edu; N. Sohre, R. Narain, and S.J. Guy, Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN; email: {sohre007,narain,sjguy}@umn.edu.

0730-0301/2017/7-ART136 $15.00
DOI: http://dx.doi.org/10.1145/3072959.3073705

for numerically challenging problems, while variational integrators offer impressive long-term conservation of energy and momenta. Recent advances in physics-based animation have made these time integration schemes faster and more robust by formulating them in terms of optimization, leading to performance speedups for implicit variational integrators [Kharevych et al. 2006] and highly robust and parallelizable solvers for backward Euler integration [Bouaziz et al. 2014; Gast et al. 2015]. These practical benefits are made possible by working directly with the underlying energy function that determines the dynamics of the system.

In a similar fashion, many modern techniques for procedural animation of intelligent systems model agents as individual entities each trying to minimize an energy or cost function, such as the distance to the goal or a cost based on proximity to other agents. However, in practice these techniques tend to use simple time integration schemes such as forward Euler, which require small time steps to maintain stability [Karamouzas et al. 2014; Pelechano et al. 2007; Reynolds 1999]. The necessity of small time steps brings with it several issues that widely affect fields such as crowd simulation. For example, it is often necessary to carefully tune simulation parameters such as the size of the time step for each new scenario. If the time step is too small, the simulation becomes computationally expensive; if it is too large, collisions, jittering, and discontinuous motion can be observed. The magnitude of inter-agent interaction forces can be adjusted to smooth out motion to an extent, but dense scenarios often still require impractically small time steps to be able to produce smooth motion. Even when these simulations are computationally feasible, very small time steps directly give rise to frequent changes in velocity, which can make it difficult to apply character animations to simulated trajectories. Consequently, many existing crowd simulations are fragile and require deep familiarity with the underlying methods to adapt to new scenarios.

In this paper, our goal is to allow the application of optimization-based integration schemes to enable robust simulation of large systems of intelligent agents, such as crowd simulation. A key challenge in applying these physics-based animation techniques to the simulation of intelligent systems is the non-physical nature of the dynamics. The entities in these systems do not follow force laws based on conservative position-dependent forces, but rather they represent agents that are trying to anticipate and react to the future trajectories of their neighbors. Such behavior cannot be modeled using traditional position-based potential energies, as it depends strongly (and nonlinearly) on the relative velocities of the agents. In this work, we show that it can instead be described in terms of an *anticipatory potential*, analogous to the Rayleigh dissipation function in classical mechanics. We present an approach to apply optimization-based time integration schemes to systems described by a general dissipation function, including those with highly nonlinear anticipatory forces that are characteristic of intelligent agents.

Given the above goals, our work makes two main contributions:

(1) We explicitly connect the damping formulations of Kharevych et al. [2006] and Gast et al. [2015] to Rayleigh dissipation functions, and provide a simple derivation of a method for incorporating general dissipation functions in optimization-based backward Euler.

(2) We combine this new integration approach with a potential function derived from a recent analysis of anticipation in human crowds, which we treat in the same way as a dissipation function. The resulting implicit crowd simulation technique allows robust simulation of anticipatory crowd dynamics.

Our implicit method yields crowd simulations with several valuable properties: stability across a very large range of time steps, scenarios, and densities; guaranteed collision-free motion; smooth, efficient trajectories for all agents; and anticipatory behavior between agents leading to the emergence of collective phenomena.

2 RELATED WORK

2.1 Crowd Simulation

Many approaches exist to simulate crowds, including techniques based on continuum dynamics [Hughes 2002; Treuille et al. 2006], data-driven approaches [Charalambous and Chrysanthou 2014; Ju et al. 2010; Lerner et al. 2007], as well as methods for interactive crowd authoring [Kim et al. 2014; Normoyle et al. 2014]. Below, we highlight some prior work on local navigation that is highly relevant to our paper. We also refer the reader to Sections 6 and 7 for comparisons of our approach with existing local collision avoidance schemes.

In general, local navigation methods can be classified into force-based approaches and velocity-based models. Force-based methods represent humans as particles and model their interactions using physical forces. Two of the most popular force-based methods are Reynolds's boids model [1987], which captures flocking behavior using separation, alignment, and cohesion forces, and the social force model of Helbing et al. [2000] which uses a mixture of sociological and physical forces to model pedestrian interactions. In both approaches, the forces depend only on the separation of the agents and, hence, can lead to simulation artifacts such as oscillations and backward movement. Since these two seminal works, many approaches have been proposed to address such issues, including techniques to control individual agents in dense crowds [Pelechano et al. 2007], and more recently anticipatory models that use the notion of time to collision to predict and resolve collisions [Karamouzas et al. 2009, 2014; Reynolds 1999; Singh et al. 2011a].

The main advantage of all the aforementioned force-based approaches is the simplicity in their formulation. However, they often require careful parameter tuning to generate desired simulation results. In addition, they suffer from numerical stability issues, since forces can assume large values and vary quickly. As such, velocity-based models have gained popularity over the past few years. These models plan directly in the velocity space by selecting at each simulation step a new velocity for each agent according to a given cost function. Numerous velocity-based formulations have been proposed in the past decade, including time-to-collision approaches [van den Berg et al. 2008], techniques based on the notion of minimal predicted distance [Moussaïd et al. 2011; Pettré et al. 2009], as well as geometrical methods based on linear programming [van den Berg et al. 2011]. Recently, vision-based approaches have also been introduced that can more closely match human behavior [Dutra et al. 2017; Hughes et al. 2015; Kapadia et al. 2012; Ondřej et al. 2010], as well as approaches that account for the holonomic nature of the human motion [Hughes et al. 2014; Singh et al. 2011b; Stuvel et al. 2016], non-linear motions [Wolinski et al. 2016], and behavioral realism [Kapadia et al. 2015].

2.2 Optimization-Based Integration

In physics-based animation, implicit time integration techniques such as the backward Euler method have long been popular due to their robustness and unconditional stability. These techniques lead to a nonlinear system of equations, whose solution is typically approximated by linearizing the problem about the current state [Baraff and Witkin 1998; Terzopoulos et al. 1987]. However, linearization works poorly in the presence of strong nonlinearities, for example as observed by Kaufman et al. [2014] in the case of hair contact simulation. Recent work has taken advantage of an alternative formulation of backward Euler for conservative systems, where the system of equations is replaced with an associated optimization problem [Gast et al. 2015; Martin et al. 2011]. This approach yields efficient and robust algorithms that can cope with strong nonlinearities, and has been applied successfully to simulation of elastic bodies [Bouaziz et al. 2014; Fratarcangeli et al. 2016; Liu et al. 2013, 2016; Martin et al. 2011; Narain et al. 2016; Wang and Yang 2016], snow [Gast et al. 2015], and fluids [Weiler et al. 2016]. However, as the interaction forces that arise when simulating multi-agent systems are both nonconservative and strongly nonlinear, none of the existing approaches for backward Euler can be applied effectively.

Another line of work has sought to formulate optimization solvers for *symplectic* time integration schemes, which tend to preserve physical invariants like energy and momenta. Most relevant to our work are Kane et al. [2000] and Kharevych et al. [2006], who also incorporate nonconservative forces such as dissipation into the optimization scheme. Kane et al. [2000] expressed the Newmark time integration scheme in an optimization form in a way that can also handle nonconservative forces arising from a Rayleigh dissipation function in an implicit manner. Kharevych et al. [2006] do not work explicitly with a dissipation function, but handle nonconservative forces by integrating them over velocities; the resulting integral can therefore be interpreted as a dissipation function. We discuss dissipation functions further in Section 3.2.

3 ENERGY-BASED FORMULATION FOR ANTICIPATORY SYSTEMS

In this section, we introduce our integration scheme for anticipatory systems, that is, systems in which interactions depend not only on the current state but also on the expected future state. We formulate these interactions in terms of a function we call the anticipatory potential, and derive an optimization-based implicit integration scheme that supports such forces. We defer the application of this scheme to crowd simulations to Section 4.

3.1 Background: Optimization-Based Integration

For background, we briefly describe some relevant techniques in physics-based animation.

For a physical system with d degrees of freedom, its state may be characterized by its generalized position and velocity $\mathbf{x}, \mathbf{v} \in \mathbb{R}^d$. If the system has a constant inertia matrix $\mathbf{M}$, its equations of motion can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x},\mathbf{v})\end{bmatrix} \tag{1}$$

where $\mathbf{f}$ is the total generalized force acting on the system. A time-stepping scheme approximates the solution of equation (1) over a time interval $[t_n, t_{n+1}]$, computing $(\mathbf{x}^{n+1}, \mathbf{v}^{n+1})$ given $(\mathbf{x}^n, \mathbf{v}^n)$. Several common time-stepping schemes are of the form

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\big((1-\alpha)\mathbf{v}^n + \alpha\mathbf{v}^{n+1}\big), \tag{2}$$

$$\mathbf{M}\mathbf{v}^{n+1} = \mathbf{M}\mathbf{v}^n + \Delta t\big((1-\alpha)\mathbf{f}^n + \alpha\mathbf{f}^{n+1}\big), \tag{3}$$

where $\alpha \in [0,1]$ is a parameter, and we denote $\Delta t = t_{n+1} - t_n$, $\mathbf{f}^n = \mathbf{f}(\mathbf{x}^n, \mathbf{v}^n)$, and $\mathbf{f}^{n+1} = \mathbf{f}(\mathbf{x}^{n+1}, \mathbf{v}^{n+1})$ for brevity. For example, $\alpha = 0$ and $\alpha = 1$ give forward and backward Euler respectively. The choice $\alpha = \frac{1}{2}$ gives the trapezoid rule, which is second-order accurate, and coincides with the Newmark method with $\beta = \frac{1}{4}$.

When $\alpha = 0$, the scheme is explicit, but its stability depends on the size of the time step Δt. If $\mathbf{f}$ varies rapidly with $\mathbf{x}$ or $\mathbf{v}$, the maximum stable time step can become extremely small, necessitating an onerously large number of time steps. For general α, we can plug equation (2) into equation (3) to reduce the problem to a system of equations in $\mathbf{v}^{n+1}$ alone. For simplicity, from here on we will only consider the $\alpha = 1$ case (backward Euler), although the technique can be readily extended to other values of α. We obtain that $\mathbf{v}^{n+1}$ satisfies the equation

$$\frac{1}{\Delta t}\mathbf{M}(\mathbf{v}^{n+1} - \mathbf{v}^n) = \mathbf{f}(\mathbf{x}^n + \mathbf{v}^{n+1}\Delta t, \mathbf{v}^{n+1}). \tag{4}$$

While the backward Euler method has much better stability properties than forward Euler, it requires solving the above d-dimensional nonlinear system, which can be numerically challenging for large problems.

Recent work in physics-based animation [Gast et al. 2015; Martin et al. 2011] has promoted an alternative approach when the forces in the system are *conservative*, that is, they are proportional to the gradient of a potential energy U independent of $\mathbf{v}$,

$$\mathbf{f}(\mathbf{x},\mathbf{v}) = -\nabla U(\mathbf{x}). \tag{5}$$

In this case, the time-stepping scheme can be expressed as the minimization of a single scalar-valued function,

$$\mathbf{v}^{n+1} = \arg\min_{\mathbf{v}}\left(\frac{1}{2}\|\mathbf{v} - \mathbf{v}^n\|_{\mathbf{M}}^2 + U(\mathbf{x}^n + \mathbf{v}\Delta t)\right), \tag{6}$$

where $\|\mathbf{v}\|_{\mathbf{M}} = (\mathbf{v}^T\mathbf{M}\mathbf{v})^{1/2}$ denotes the kinetic metric. (In existing work, this is usually expressed in an equivalent form with $\mathbf{x}^{n+1}$ as the optimization variable.) This works because the first-order optimality condition of equation (6) is precisely the time-stepping rule for $\mathbf{v}^{n+1}$. The objective itself can be interpreted as the sum of a "momentum potential" $\frac{1}{2}\|\mathbf{v}^{n+1} - \mathbf{v}^n\|_{\mathbf{M}}^2$ and the potential energy $U(\mathbf{x}^{n+1})$. This formulation enables strongly nonlinear forces to be handled without linearization, and permits the use of powerful numerical optimization techniques that are robust to nonsmoothness [Gast et al. 2015] and highly parallelizable [Bouaziz et al. 2014].

3.2 Velocity-Dependent Interactions and the Rayleigh Dissipation Function

Our goal in this work is to bring the advantages of optimization-based time integration to bear on the problem of multi-agent simulation. The essential challenge is that the approach of equation (6) is formulated for conservative forces that depend only on position, whereas we wish to model agents that are *anticipatory* and react to each other's current velocities as well. These are comparable to velocity-dependent dissipative forces in classical mechanics, and the integration method we use is closely related to existing techniques for handling such forces in symplectic integrators [Kane et al. 2000; Kharevych et al. 2006]. In this section, we provide a simpler, elementary derivation of an analogous technique for the backwards Euler method, which is attractive in graphics due to its excellent stability properties. We hope that our presentation will spur interest in this area in the computer animation community.

Many nonconservative forces in mechanics can be expressed as

$$\mathbf{f}(\mathbf{x},\mathbf{v}) = -\nabla_2 R(\mathbf{x},\mathbf{v}), \tag{7}$$

where R is known as the Rayleigh dissipation function [Goldstein 1980], and ∇_2 denotes the gradient with respect to the second argument. The dissipation function is usually assumed to be of the form $R(\mathbf{x},\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{C}(\mathbf{x})\mathbf{v}$ for some matrix $\mathbf{C}(\mathbf{x})$; this models linear dissipation forces, including generalized Rayleigh damping and Newtonian viscosity. However, a general nonlinear dissipation function can also be considered [Marsden and Ratiu 1999], allowing forces that depend nonlinearly on velocity.

In the context of multi-agent systems, many inter-agent interaction models can also be expressed as $-\nabla_2 R(\mathbf{x},\mathbf{v})$ for a nonlinear R, as we discuss in Section 4. Here R is thought of as the cost for agents to take velocities $\mathbf{v}$ given one another's positions and velocities, and the associated force drives them away from high-cost velocities. As the term "dissipation function" is not well suited to the role of R in multi-agent systems, we will refer to it as the *anticipatory potential* giving rise to anticipatory forces between agents, analogous to the conservative potential U which gives rise to conservative forces.

The force $\mathbf{f}(\mathbf{x},\mathbf{v}) = -\nabla_2 R(\mathbf{x},\mathbf{v})$ can be used in optimization-based backward Euler with only a small modification of the objective:

$$\mathbf{v}^{n+1} = \arg\min_{\mathbf{v}} \left(\frac{1}{2}\|\mathbf{v}-\mathbf{v}^n\|^2_{\mathbf{M}} + R(\mathbf{x}^n+\mathbf{v}\Delta t,\mathbf{v})\Delta t\right). \tag{8}$$

This minimization solves the condition

$$\frac{1}{\Delta t}\mathbf{M}(\mathbf{v}^{n+1}-\mathbf{v}^n) = \mathbf{f}(\mathbf{x}^{n+1},\mathbf{v}^{n+1}) + \nabla_1 R(\mathbf{x}^{n+1},\mathbf{v}^{n+1})\Delta t, \tag{9}$$

where the extraneous term is proportional to $\nabla_1 R\Delta t$ and vanishes as $\Delta t \longrightarrow 0$. Thus, the minimization problem (8) still provides a first-order accurate discretization of the equations of motion (although we would not retain the second-order accuracy of the trapezoid method if applied to $\alpha = \frac{1}{2}$). Gast et al. [2015]'s Rayleigh damping model is equivalent to using $R(\mathbf{x}^n,\mathbf{v})$ instead; this avoids the extra term but is not fully implicit in positions, and would not suffice for the collision avoidance guarantees we show in Section 5.1.

For completeness, we state the full time-stepping scheme for a system with both conservative and anticipatory forces, that is, $\mathbf{f}(\mathbf{x},\mathbf{v}) = -\nabla U(\mathbf{x}) - \nabla_2 R(\mathbf{x},\mathbf{v})$. The objective function is

$$f(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}-\mathbf{v}^n\|^2_{\mathbf{M}} + U(\mathbf{x}^n+\mathbf{v}\Delta t) + R(\mathbf{x}^n+\mathbf{v}\Delta t,\mathbf{v})\Delta t, \tag{10}$$

and time stepping is performed via

$$\mathbf{v}^{n+1} = \arg\min_{\mathbf{v}} f(\mathbf{v}), \tag{11}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1}\Delta t. \tag{12}$$

The minimization of f can be performed using standard optimization algorithms, as described in Section 5. We note that the terms $U(\mathbf{x}^n+\mathbf{v}\Delta t) + R(\mathbf{x}^n+\mathbf{v}\Delta t,\mathbf{v})\Delta t$ in equation (10) correspond to the "effective incremental potential" of Kane et al. [2000].

In the next section, we show how this optimization-based integration approach can be applied to inter-agent interactions in pedestrian crowds, which is the focus of our current work.

4 IMPLICIT CROWDS

Perhaps the most commonly encountered anticipatory system in real life is a crowd of pedestrians. Unlike simple physical systems, pedestrians interact with each other well before the moment of actual collision. As such, the interactions between a pair of pedestrians is strongly nonlinear in both their relative positions and velocities, and the implicit integration approach introduced in the previous section is well suited for simulating them. In this section, we define a generic crowd dynamics model using an energy-based formulation. We then describe a recently proposed interaction energy function derived from human motion data, which poses numerical difficulties for standard methods but fits well into our proposed method. Finally, we introduce two modifications to the system's energy to improve the behavior of numerical solvers. This leads to an optimization-based implicit integration formulation for simulating human crowds.

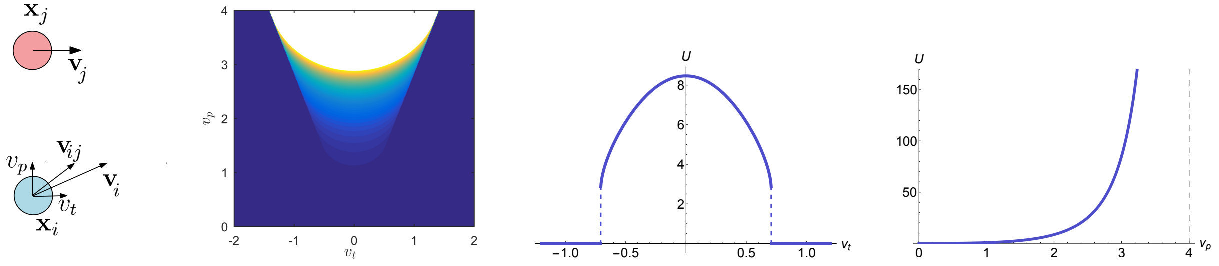
4.1 An Energy-Based Crowd Model

In our problem setting, we are given m agents that must navigate through a scene without colliding with each other and the environment. We assume that the agents move on a plane, and are modeled as discs with radii $r_1,\ldots,r_m \in \mathbb{R}^m$. The entire state of the system can be characterized by stacking the positions $\mathbf{x}_1,\ldots,\mathbf{x}_m$ and velocities $\mathbf{v}_1,\ldots,\mathbf{v}_m$ of the agents into the $2m$-dimensional vectors

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \vdots\\ \mathbf{x}_m\end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix}\mathbf{v}_1\\ \vdots\\ \mathbf{v}_m\end{bmatrix}. \tag{13}$$

All agents are assumed to have unit mass; therefore, the inertia matrix $\mathbf{M}$ is the identity, and $\|\cdot\|_{\mathbf{M}} = \|\cdot\|$.

We assume that a procedure external to our algorithm, such as a global planning routine, provides each agent with a goal velocity $\mathbf{v}_i^g$ at every time step. In its simplest form, the goal velocity of each agent is the vector pointing from the agent's current position to a user-specified goal position, with a magnitude equal to a user-specified preferred speed. Each agent's velocity is driven towards its goal velocity via a force $\mathbf{f}_G = \xi(\mathbf{v}_i^g - \mathbf{v}_i)$, similar to that in the social force model [Helbing et al. 2000]. Here ξ is a tunable parameter



(a) (b) (c) (d)

Fig. 2. (a) For two agents at positions $\mathbf{x}_i = (0, 0)$ and $\mathbf{x}_j = (0, 3)$, we define v_p and v_t as the components of their relative velocity $\mathbf{v}_{ij}$ parallel to and perpendicular to the line joining them. (b) For a time step of $\Delta t = 0.5$, we visualize the time-to-collision potential term $R_{\text{TTC}}(\mathbf{x}_{ij} + \mathbf{v}_{ij}\Delta t, \mathbf{v}_{ij})$ that appears in the objective function, equation (10), as a function of v_p and v_t. Brighter colors indicate higher energy. (c) Taking a slice at $v_p = 2$, the variation of R_{ij} with v_t shows discontinuities at the boundary of the VO cone. (d) Taking a slice along $v_t = 0$, the potential reaches infinity at $v_p = (\|\mathbf{x}_{ij}\| - r)/\Delta t$, that is, when the agents would collide in a single time step.

controlling the strength of the goal force. This force corresponds to a potential

$$R_{\text{G}}(\mathbf{v}_i) = \frac{1}{2}\xi\|\mathbf{v}_i - \mathbf{v}_i^{\text{g}}\|^2. \tag{14}$$

Furthermore, agents react to other agents' positions and velocities in order to avoid collisions. This behavior can be modeled by introducing a pairwise interaction potential $R(\mathbf{x}_i - \mathbf{x}_j, \mathbf{v}_i - \mathbf{v}_j)$ between all pairs of agents i, j. Different choices for this potential lead to different crowd dynamics models. Nevertheless, if the energy tends to infinity as the agents approach a collision, the model is guaranteed to prevent collisions, as we show in Section 5. We discuss two examples from prior work below, and turn to a recent energy-based interaction model in the next subsection.

- In Reynolds' boids model [Reynolds 1987], separation and cohesion are purely position-dependent. The alignment behavior can be modeled via an anticipatory interaction potential $R(\mathbf{x}_i - \mathbf{x}_j, \mathbf{v}_i - \mathbf{v}_j) = w(\|\mathbf{x}_i - \mathbf{x}_j\|)\|\mathbf{v}_i - \mathbf{v}_j\|^2$, where w is a weighting function that decreases with distance.
- In the velocity obstacle approach [Fiorini and Shiller 1998], velocities are disallowed if their extrapolated trajectories lead to a collision in the future. This model can be interpreted as an interaction potential that assigns infinite energy to colliding velocities, and zero energy to all others (see also Section 4.3). The time stepping problem (10)–(12) then is best understood as constrained optimization: minimize all the other terms in the objective, subject to the constraint that the interaction energy is finite, i.e. the velocities do not lead to a collision.

4.2 Time-To-Collision Potential

Karamouzas et al. [2014] have recently analyzed a wide range of publicly available crowd data [Lerner et al. 2007; Pellegrini et al. 2009; Seyfried et al. 2009], showing that the interaction energy between any given pair of pedestrians follows a *power law* as a function of their projected *time to collision* (TTC), denoted τ. Formally, the interaction potential between two pedestrians or agents, i and j, is given by:

$$R_{\text{TTC}}(\mathbf{x}_{ij}, \mathbf{v}_{ij}) = k\tau^{-p}e^{-\tau/\tau_0}, \tag{15}$$

where k is a scaling constant, and p is the exponent of the power law; the truncation time τ_0 models the fact that people tend to ignore collisions that take place in the far future [Olivier et al. 2012].

Here, the time to collision τ is understood to be a function of the relative displacement between the two agents, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, their relative velocity, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and their combined radius $r = r_i + r_j$, and denotes the first time in the future at which the corresponding discs of the agents intersect. Assuming linear motion, the discs will intersect if at some time $t \geq 0$ it holds $\|\mathbf{x}_{ij} + \mathbf{v}_{ij}t\| \leq r$. This leads to a quadratic equation in t, and τ is given by its smallest positive root (see supplementary material for more details):

$$\tau = \frac{-\mathbf{x}_{ij} \cdot \mathbf{v}_{ij} - \sqrt{(\mathbf{x}_{ij} \cdot \mathbf{v}_{ij})^2 - \|\mathbf{v}_{ij}\|^2(\|\mathbf{x}_{ij}\|^2 - r^2)}}{\|\mathbf{v}_{ij}\|^2} \tag{16}$$

If no such root exists, we take $\tau = \infty$, which subsequently leads to a zero interaction energy.

The TTC potential R_{TTC} has two features that make it unsuitable to be used directly in numerical optimization: a jump discontinuity along the boundary between colliding and non-colliding velocities, and arbitrarily steep behavior for agents on grazing trajectories. Therefore, we define a modified potential $R_{\text{TTC}'}$ by smoothing over the discontinuity, and add a position-dependent repulsion energy U_R to prevent agents from getting arbitrarily close. These modifications are described in the following subsections.

Our energy formulation thus consists of terms for repulsion (U_{R}), goal direction (R_{G}), and time to collision ($R_{\text{TTC}'}$). The total potential functions for the crowd are

$$U(\mathbf{x}, \mathbf{v}) = \sum_{i \neq j} U_{\text{R}}(\mathbf{x}_{ij}), \tag{17}$$

$$R(\mathbf{x}, \mathbf{v}) = \sum_{i} R_{\text{G}}(\mathbf{v}_i) + \sum_{i \neq j} R_{\text{TTC}'}(\mathbf{x}_{ij}, \mathbf{v}_{ij}), \tag{18}$$

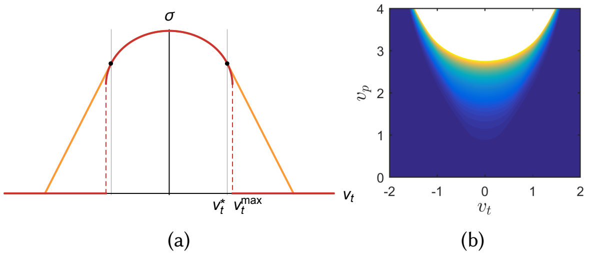


Fig. 3. (a) We smooth over the discontinuity in the inverse time to collision σ (red) by linear extrapolation from $v_t = \pm v_t^*$. The resulting function $\bar{\sigma}$ (orange) is continuous everywhere. (b) A visualization of the smoothed potential term $R_{\mathrm{TTC}'}(\mathbf{x}_{ij} + \mathbf{v}_{ij}\Delta t, \mathbf{v}_{ij})$; compare with Figure 2b.

and the total force is given by $\mathbf{f}(\mathbf{x}, \mathbf{v}) = -\nabla U(\mathbf{x}) - \nabla_2 R(\mathbf{x}, \mathbf{v})$ as usual. As described in Section 3.2, backward Euler integration for the system can now performed via equations (10)–(12).

Agents also experience interaction potentials arising from static polygonal obstacles in the environment. These agent-obstacle interactions are incorporated using the same time-to-collision and repulsion energies used in agent-agent interactions. However, these energies are evaluated between each agent and the union of all obstacles. As a result, only the energy contributed by the most threatening obstacle contributes to R (rather than, for example, adding the energy from every obstacle).

4.3 A Continuous Time-To-Collision Potential

Consider two agents as shown in Figure 2a. Their TTC potential, as defined in equation (15), depends on the time τ that it takes for the two agents to collide. Given the discs of the two agents located at $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, there is a range of relative velocities $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ that will result in a collision. In robotics, this set is commonly referred to as a velocity obstacle (VO), and can geometrically be interpreted as a cone in $\mathbf{v}_{ij}$ space with its apex at the origin and its legs tangent to the disc of radius $r = r_i + r_j$ centered at $-\mathbf{x}_{ij}$ [Fiorini and Shiller 1998].

Figure 2b depicts the TTC potential of our current example for different relative velocity values. Even in this simple 2-agent scenario, the interaction energy is discontinuous on the boundary of the VO cone, which leads to numerical difficulties in optimization. This discontinuity arises because τ is infinite outside the cone but has a finite limit approaching the boundary from the inside, and so R_{TTC} jumps from zero to a finite nonzero value.

To analyze the situation more closely, we consider a local coordinate system in velocity space, centered at the origin and oriented along $-\mathbf{x}_{ij}$. We can then decompose the relative velocity into two components: its projection, v_p, along the direction of $-\mathbf{x}_{ij}$, and its tangential counterpart, v_t, (see Figure 2a). For convenience, we will work with the reciprocal of the time to collision, $\sigma = \tau^{-1}$, which is zero when the agents are not on a colliding trajectory, and becomes larger for more imminent collisions. Figure 3a plots σ for different v_t values (solid red plot). We will define a continuous approximation to this function, $\bar{\sigma}$, by smoothing over the transitions between zero and positive values.

Assuming that $v_p \geq 0$, that is, the agents are approaching each other (otherwise $\tau = \infty$ and $\sigma = 0$), σ is given by the reciprocal of equation (16). As can be seen from Figure 3(a), σ is symmetric in v_t, and decreases from its maximum at $v_t = 0$ to a finite nonzero value at $v_t^{\max} = (r\, v_p)/(\|\mathbf{x}_{ij}\|^2 - r^2)$ before dropping discontinuously to zero (see supplementary material for details). We define $\bar{\sigma}$ by selecting an intermediate velocity between 0 and $v_t^{\max}$, namely $v_t^* = \sqrt{1 - \epsilon^2}\, v_t^{\max}$ for a parameter $\epsilon \in (0, 1)$. For any $|v_t| \leq v_t^*$, we keep the unchanged value $\bar{\sigma} = \sigma$, while the smoothed $\bar{\sigma}$ for $|v_t| > v_t^*$ is computed by linearly extrapolating from $\pm v_t^*$ and then clamping to nonnegative values.

This results in a reciprocal time-to-collision function that is continuous everywhere and differentiable away from 0, as shown in Figure 3a (dashed orange plot). Therefore, the corresponding interaction potential

$$R_{\mathrm{TTC}'} = k\bar{\sigma}^p e^{-1/(\bar{\sigma}\tau_0)} \tag{19}$$

visualized in Figure 3(b) is differentiable everywhere for $p > 1$. For a detailed explanation of the smoothing process see the supplementary material.

4.4 Maintaining inter-agent separation

The TTC potential $R_{\mathrm{TTC}'}$ depends on the projected time to collision, so two agents moving very narrowly past each other but not actually colliding will experience negligible force. As a result, agents tend to take grazing trajectories, passing at a separation of nearly zero in their closest approach. Since $R_{\mathrm{TTC}'}$ rises from zero to infinity over the interval $v_p \in \left[0, (\|\mathbf{x}_{ij}\| - r)/\Delta t\right]$, the objective becomes "L-shaped" with an infinite slope on one side as the distance between agents, $\|\mathbf{x}_{ij}\|$, approaches r (Figure 2d). This leads to difficulties for numerical optimization: while the use of a standard line search procedure permits the use of objectives that rise *smoothly* to infinity, progress can still stall in regions where the objective's gradient changes sharply in a small neighborhood.

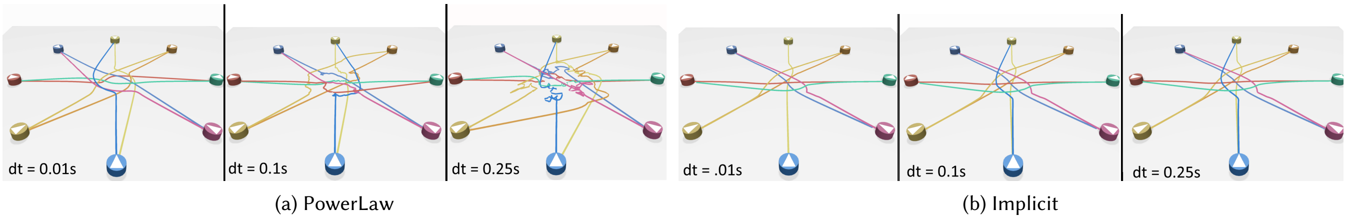
To avoid this issue, we add a distance-based repulsion energy U_{R}, defined as

$$U_{\mathrm{R}}(\mathbf{x}_{ij}) = \begin{cases} \dfrac{\eta}{\|\mathbf{x}_{ij}\| - r} & \text{if } \|\mathbf{x}_{ij}\| > r, \\ \infty & \text{otherwise,} \end{cases} \tag{20}$$

where η is a scaling constant. This energy acts as a barrier function that rises smoothly to infinity as the agents approach each other. Thus, the separation between agents remains bounded away from 0, and the numerical issues due to unbounded steepness in $R_{\mathrm{TTC}'}$ do not arise.

Continuous collision detection. The backward Euler integration scheme only considers the interaction energies at the end of the time step. Therefore, it is possible in principle that agents can "tunnel" through each other and appear on the other side, even though the trajectory between the initial and final state would have led to interpenetration at an intermediate time. This is essentially the well-known limitation of discrete collision detection.

A simple extension of the repulsion potential can eliminate tunneling as well. Instead of computing the repulsion based on the current distance $\|\mathbf{x}_{ij}\|$, we use the *minimum* distance over the current time step. For linear trajectories, this has the simple closed



(a) PowerLaw

(b) Implicit

Fig. 4. Comparison between (a) a standard PowerLaw simulation and (b) a simulation computed using our Implicit approach. Here, 8 agents must walk to antipodal points in a circle. The PowerLaw model leads to collisions and other discontinuities in motion with time steps much larger than 10ms. Our approach is smooth and collision free for time steps of any size.

form

$$d_{\min} = \min_{\alpha \in [0,1]} \|(1-\alpha)\mathbf{x}_{ij}^n + \alpha \mathbf{x}_{ij}\| \tag{21}$$

$$= \|(1-\alpha^*)\mathbf{x}_{ij}^n + \alpha^* \mathbf{x}_{ij}\| \tag{22}$$

where α^* is where the minimum is attained,

$$\alpha^* = \text{clamp}\left(\frac{\mathbf{x}_{ij}^n \cdot (\mathbf{x}_{ij}^n - \mathbf{x}_{ij})}{\|\mathbf{x}_{ij} - \mathbf{x}_{ij}^n\|^2}, [0,1]\right). \tag{23}$$

Using $d_{\min}$ instead of $\|\mathbf{x}_{ij}\|$ in (20) effectively performs continuous collision detection, sending the repulsion energy to infinity whenever the trajectories would intersect at some point within the time step. In our implementation, we added a small ϵ term to the denominator of α^* to avoid nondifferentiability when $\mathbf{x}_{ij} = \mathbf{x}_{ij}^n$.

5 ANALYSIS AND IMPLEMENTATION

5.1 Theoretical Properties

Our technique has the attractive theoretical properties of guaranteed collision-freeness and C^2-continuity, as we explain in this section.

First, through the use of implicit integration and smoothed TTC potentials, we can guarantee absence of collisions even for arbitrarily large time steps. Observe that if any two agents i and j are interpenetrating, their time to collision, τ, is zero and so their TTC potential as defined by equation (19) is infinite. Therefore, for any candidate velocity $\mathbf{v}^{n+1}$ that leads to an interpenetration at the next configuration $\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1}\Delta t$, the potential at the next time step $R(\mathbf{x}^{n+1}, \mathbf{v}^{n+1})$ would be infinite, and consequently so would our objective function $f(\mathbf{v}^{n+1})$ defined in equation (10). Such a velocity cannot be a minimum unless $f(\mathbf{v}^{n+1}) = \infty$ for *all* possible $\mathbf{v}^{n+1}$, that is, all possible velocities lead to interpenetrations. This is impossible in the absence of moving obstacles: in particular, when $\mathbf{v}^{n+1} = \mathbf{0}$, the TTC potential is 0 for all pairs of agents, so $f(\mathbf{0})$ is finite. Thus, the algorithm will never take a step which leads to collisions.

Furthermore, our formulation is smooth in the sense that the agent trajectories are provably C^2-continuous in the limit $\Delta t \to 0$. Our approach converges to the solution of a system of ordinary differential equations,

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t), \tag{24}$$

$$\dot{\mathbf{v}}(t) = -\nabla U(\mathbf{x}(t)) - \nabla_2 R(\mathbf{x}(t), \mathbf{v}(t)). \tag{25}$$

With the smoothed TTC potential, both U and R are differentiable, so the right-hand side of equation (25) is a continuous function of time. Integrating twice, we find that the trajectories $\mathbf{x}(t)$ are C^2-smooth with respect to time. In practice, with our implicit integration scheme the trajectories are also visually smooth for finite time steps, even moderately large ones (Fig. 4).

5.2 Implementation Details

We implemented in C++ the optimization-based model described in Section 4. The simulator advances through each time step from t^n to t^{n+1} by numerically solving the minimization problem defined in equations (10) and (11) and returning the new velocities $\mathbf{v}^{n+1}$ for the agents. Then, the positions of the agents are updated via equation (12).

The numerical minimization itself is carried out using our implementation of the L-BFGS method [Nocedal and Wright 2006], along with an inexact line search using the Armijo condition. L-BFGS is a quasi-Newton approach that maintains a compact approximation of the objective's Hessian to compute the descent direction. At each iteration, the solver requires the value of the objective $f(\mathbf{v})$ and its gradient $\nabla f(\mathbf{v})$ at the current iterate, updates its Hessian estimate accordingly, and takes a quasi-Newton step. The iterations are terminated when a fixed number of optimization iterations has been met, or the L^∞-norm of the difference between consecutive iterates is below a small user-defined threshold.

In our implementation, we compute the objective and its gradient analytically. To improve performance, when evaluating the interaction terms $R_{\text{TTC}'}$ and U_{R}, we only consider pairs of agents whose distance is within an interaction radius at the start of the time step. As we use L-BFGS rather than a direct Newton solver, we do not require an analytical Hessian, which could be onerous to calculate especially for the smoothed time-to-collision term. Furthermore, the full Hessian can become quite dense in a crowded scenario, when tens or hundreds of agents are within each agent's distance threshold. We did test Newton's method using automatic differentiation to compute the Hessian, but found that the memory-efficient L-BFGS to lead to better performance for our problem setting.

Another issue is the choice of initial guess for $\mathbf{v}^{n+1}$. Using the previous velocity $\mathbf{v}^n$ may not be advisable, because if $\mathbf{x}^n + \mathbf{v}^n\Delta t$ is a colliding state, the initial value of the objective will be infinite and the numerical method will fail. On the other hand, as long as the agents are not already colliding at the start of the step, choosing

$\mathbf{v}^{n+1} = \mathbf{0}$ is guaranteed to be a feasible state, so we use it as the initial guess. As our solver uses a line search that never accepts an infinite-energy iterate, the simulation is guaranteed to always be in a collision-free state, even if the solver is not run to convergence.

6 RESULTS

In this section, we detail quantitative results based on several simulations obtained with our Implicit method. We encourage readers to view the corresponding animations in the accompanying video. In all of our simulations, unless otherwise specified, we used $k = 2$, $\epsilon = 0.2$, $\tau_0 = 3$s, and $p = 2$ for the parameters of the TTC potential in equation (19), $\eta = 0.01$ for the repulsion energy defined in equation (20), and $\xi = 2$ for the scaling constant of the goal potential given in equation (14). Furthermore, we considered pairwise interactions only between agents that are less than 10 m away. Regarding the parameters of our L-BFGS solver, we used 100 optimization iterations, set the size of the L-BFGS memory window to 5, and employed early stopping whenever the difference between the new and previous solution was below 10^{-5}.

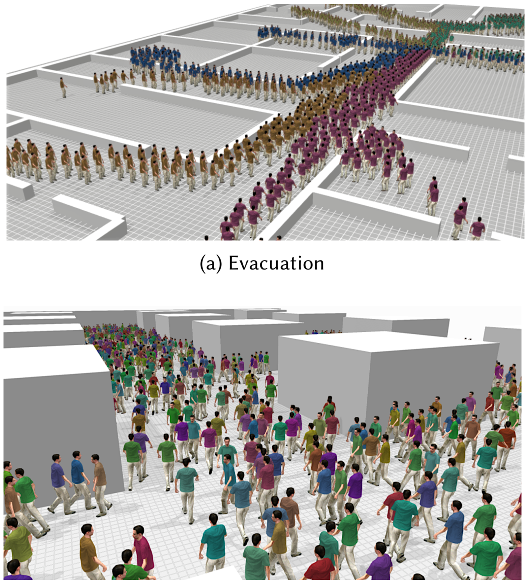
In analyzing our results, we considered the following scenarios:
Crossing: 400 agents are split into four groups and cross paths perpendicularly (Figure 1a).
Hallway: Two groups, each having 150 members, cross paths while coming from opposite directions of an open hallway.
Blocks: 2000 agents walk around a virtual city block set-up following a roadmap for global navigation (Figure 5b).
Evacuation: 1200 agents evacuate an office building following a visibility roadmap for global navigation and periodically replanning their global routes (Figure 5a).
Random: Agents are given multiple random goals on an empty plane. The number of agents is varied to test performance.

Below we show simulation results using our method, which we refer to as Implicit, and compare them to simulations obtained using the ORCA framework of van den Berg et al. [2011] and the time-to-collision force model (PowerLaw) of Karamouzas et al. [2014]. We chose ORCA as it is one of the few existing methods that provides some formal guarantees about collision-free motion, and the PowerLaw model as a state-of-the-art anticipatory force-based model that has been well validated against human behavior.

6.1 Time Step Stability

We start by comparing the robustness of PowerLaw, ORCA, and Implicit to different time step sizes Δt. Table 1 shows the range of collision-free time steps for each simulation method on each of the above scenarios. In ORCA and PowerLaw, we consider two agents as colliding only if their disks overlap by more than 5%. This allows agents to have small overlaps (as happens with PowerLaw) or glancing collisions (as happens with ORCA) without any penalties.

Because the PowerLaw approach relies on explicit integration, like other existing force-based models used in crowd simulation, it needs very small time steps to provide collision-free motion. For very complex dense scenarios, such as Evacuation, the method produces collisions even at very small time steps ($\Delta t < 5$ ms). As a result, obtaining collision-free simulations for arbitrary scenarios would require impractically small time steps or careful parameter tuning.


(a) Evacuation

(b) Blocks

Fig. 5. Our Implicit approach can be integrated with roadmaps to allow the simulation of complex scenarios. (a) 1200 agents evacuate a complex office setting. (b) 2000 agents walk through a virtual city block.

In contrast to the PowerLaw method, where the agents follow a statistically-derived energy function, the ORCA framework plans motions directly in velocity space. ORCA uses a constrained optimization approach, solving a linear program for each agent to bring it towards its goal while avoiding collisions. This approach leads to a substantially larger region of collision-free time steps than the PowerLaw approach. However, the limitation of the per-agent approach is that for some conditions there may be no feasible velocity which is provably collision-free. In such cases, ORCA relaxes the constraints, allowing for the possibility of collisions. As can be seen in Table 1 such conditions are more likely to occur in dense scenarios such as *Crossing* and *Evacuation*, leading to a substantial reduction in the allowable time step.

Unlike either ORCA or the PowerLaw, our proposed Implicit approach is unconditionally collision-free at all time steps in all scenarios tested. Here, we did *not* relax the collision criterion: two agents are considered as colliding whenever there is any amount of overlap between their disks. This empirical result matches the theoretical analysis provided in the previous section. Crucially, this means that simulations using our method may choose a time step appropriate given other considerations (e.g., available computation time, or desired animation update frequency), and be confident of simulation stability regardless of how the simulation evolves. This is in contrast to previous approaches, where careful per-scenario tuning was needed simply to avoid collisions.

6.2 Simulation Accuracy

It is, of course, not sufficient for a simulation to be smooth and collision-free; we must also capture the dynamics of the system

					Maximum collision-free Δt [ms]		
	# Agents	# Obstacles	Roadmap	Density	PowerLaw	ORCA	Implicit
Hallway	300	2	no	medium	40	100	> 1000
Crossing	400	0	no	high	20	35	> 1000
Random	500	0	no	low	30	140	> 1000
Evacuation	1200	178	yes	very high	< 5	25	> 1000
Blocks	2000	112	yes	low	30	90	> 1000

Table 1. The largest time step that leads to collision-free motion, evaluated by sampling Δt over the range of 5 ms to 1000 ms for each scenario and simulation method. For both PowerLaw and ORCA the range of collision-free time steps strongly depends on the given scenario, with dense scenarios necessitating very small time steps. For Implicit, all simulation time steps lead to collision-free motion.

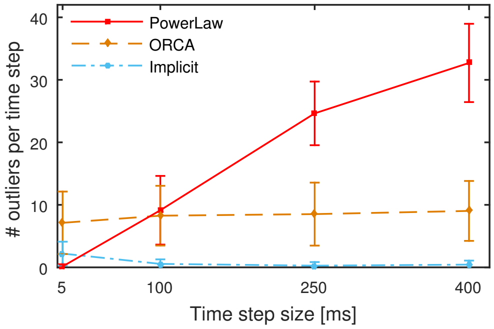


Fig. 6. Distance to nearest neighbor outliers for different time step sizes and simulation methods. The results were obtained by comparing a bottleneck simulation to real world data obtained by [Seyfried et al. 2009]. Error bars show standard deviation across simulation frames.

accurately. Here we investigate two forms of simulation accuracy. First, we compare simulated trajectories directly to observed human trajectories in similar scenarios. Second, we compare the statistical patterns in our agents' motion to those seen in real crowds.

Trajectory Comparison. Several researchers have detailed methods for evaluating the accuracy of crowd simulations based on either behavioral metrics [Kapadia et al. 2009] or reference human trajectories [Lerner et al. 2010]. Here, we follow the recent analysis approach outlined by Charalambous et al. [2014] which derives behavioral metrics from reference trajectories automatically using outlier detection.

For ground truth human motion, we use the experiments from Seyfried et al. [2009] which recorded human subjects walking out from a waiting room through a bottleneck created by a narrow corridor. We then create a custom simulation whose initial conditions closely match real-world evacuation experiments (50 agents placed in a 10 m x 5 m waiting area are directed to walk through a 2.5 m wide constriction). Localized p-value estimation (k-LPE) [Zhao and Saligrama 2009] is then used to find outliers in the simulation as compared to the ground truth data; the more outliers, the worse the simulation captures actual human behavior.

Figure 6 details the number of outliers found (as measured by local neighborhood distances) for different simulation methods over varying time steps. For small time steps, Implicit and PowerLaw both have almost no outliers, whereas ORCA is somewhat less accurate. As the time step grows, ORCA maintains the same level of accuracy, but the PowerLaw method gets significantly worse. At a fairly moderate time step of 0.1s both methods have nearly a quarter of agents producing incorrect behavior each time step. Our Implicit method produces no outliers in this scenario for a very large range of time steps (up to 400ms). We do note that for very small time steps, Implicit is slightly less accurate. This is due to the fact that the distance-based repulsion term, equation (20), is more likely to act with smaller time steps.

Statistical Anticipation Analysis. Karamouzas et al [2014] have shown that in many real-world settings, the interaction energy between humans is anticipatory in nature, following a power-law relationship with respect to time to collision. In much of the crowd data that they analyzed, this power law has an exponent of 2, that is, $p = 2$ in equation (15). Our simulations can reproduce the same anticipatory law. To account for the discrepancy between our force $\mathbf{f} = -\nabla_2 R_{\mathrm{TTC}'}$ and the force $-\nabla_1 R_{\mathrm{TTC}}$ used by Karamouzas et al., we use a larger exponent in the TTC potential of equation (19) as described in the supplementary material. We experimentally found an exponent of 3.25 to give a good match with the expected behavior after analyzing the pairwise interaction energies of our simulations.

Figure 7 shows the corresponding interaction energy graph in the Hallway scenario. As can be seen, the energy falls off quadratically with the time to collision even with very large time steps. We note that the PowerLaw model can of course reproduce the same power-law behavior, but only for sufficiently small time steps. As shown in [Karamouzas et al. 2014], ORCA can exhibit roughly similar behavior in some scenarios, but it does not match the empirically observed power law as closely as PowerLaw (and Implicit).

6.3 Performance

We evaluate the runtime performance of our approach using a 3.5GHz Intel Xeon E5-1650 with 6 cores and hyperthreading. Even for large scenarios with hundreds or even thousands of agents, our method runs at interactive rates (> 10 Hz) on all time steps tested. Our approach also maintains realtime performance (computation time < simulation time step) for all scenarios with time steps of 100ms or greater, running between 2.7x faster than realtime (on the Evacuation scenario) and 27x faster (on Hallway).

To isolate the effect of problem size and time step length on convergence, we varied these parameters while removing the 100

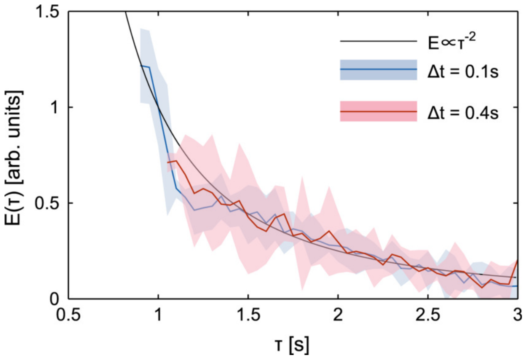


Fig. 7. The pairwise interaction energy U as a function of time-to-collision (τ) for Hallway simulations obtained using different time steps. Blue and red lines indicate the mean energy every 0.05 s and shaded regions denote the 95% confidence interval. In both simulations, the energy exhibits an inverse quadratic relationship with the time-to-collision, matching the behavior of real pedestrians as shown in [Karamouzas et al. 2014].

iteration cap in the solver. Figure 8 shows the corresponding results on the Random scenario. As can be seen in the figure, the runtime increases quite slowly as the time step increases. Varying the number of agents while keeping the time step fixed to $\Delta t = 0.25$ s results in a near-linear increase in runtime with the number of agents.

While our overall system is fast enough for many applications, our performance is significantly slower than either ORCA or the PowerLaw method, both of which can run 2-10 times faster than our method on similarly sized scenarios. We note though, that for many complex scenarios, such as Blocks and Evacuation, computing inter-agent interactions is not the computational bottleneck for the simulation; for example, global path planning may be significantly more expensive. In these scenarios, ORCA, PowerLaw, and Implicit all have somewhat similar effective runtimes.

7 DISCUSSION

Here, we supplement the above quantitative comparisons with a more qualitative analysis of the resulting behavior of our simulations, as well as a comparative analysis to other approaches.

7.1 Behavioral Analysis

An important aspect of crowd simulation is capturing the emergent phenomena in flow patterns seen in real human crowds. Our Implicit method succeeds in reproducing many of these important phenomena. For example, humans in crowds are known to emergently form lanes of travel when in bi-directional flows, swirl to avoid collisions in multi-directional encounters, and form diagonal clusters during perpendicular crossings (see, e.g., [Helbing et al. 2001] for a longer discussion of these phenomena). Our Implicit method reproduces all of these phenomena for both large and small time steps. For example, the vortex-based collision resolution can be seen in the antipodal circle scenario (Figure 4). Lane formation can be clearly seen in the Hallway scenario (see supplemental video), and diagonal clustering occurs in the crossing scenario (Figure 1). Likewise, Implicit agents clog in congestion and form "zipper" patterns when navigating through narrow bottlenecks.

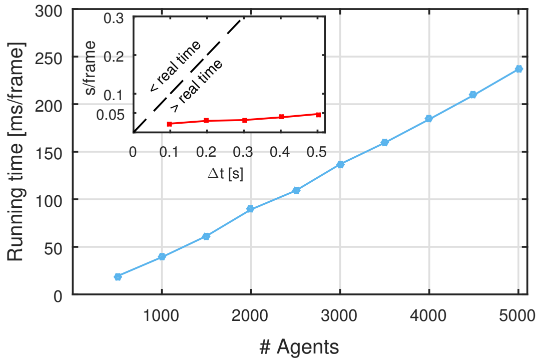


Fig. 8. Runtime performance of the Implicit method as a function of the number of agents. The experiment was run in the *Random* scenario with $\Delta t = 0.25$. The reported numbers are the averages over 5 runs, where each run computes 50 s of simulation time. The inset shows the performance for the same scenario with 500 agents while varying the size of the simulation time step.

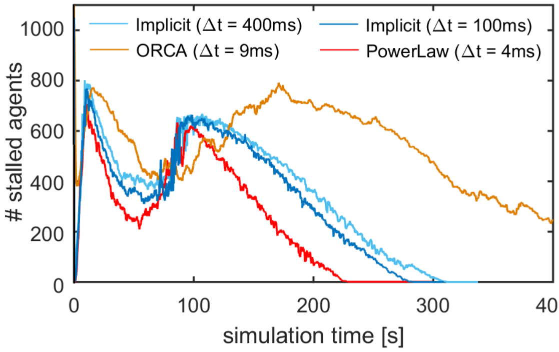


Fig. 9. The number of stalled agents (speed < 0.2 m/s) over the course of a simulation for the Evacuation scenario. PowerLaw and Implicit have similar, though not identical, behavior. ORCA agents get stuck in the congestion and fail to evacuate in a smooth, timely manner.

Overall, the qualitative behaviors produced by the PowerLaw method and Implicit are very similar, with PowerLaw simulations resulting in many of the same emergent crowd phenomena, though only at small time steps. Both methods also produce fairly similar overall flow predictions. For example, Figure 9 shows a graph of how many agents are stopped in congestion over the course of the Evacuation scenario. Both methods predict two peaks of congestion (one a few seconds in, the other at 100 s), though the PowerLaw agents complete the evacuation slightly faster.

Compared to PowerLaw and Implicit, ORCA produces very different behaviors regardless of simulation time step. Some of the behavioral differences are clearly visible in small-scale interactions. Consider, for example, the scenario in Figure 10, where three agents attempt to walk past each other. Using ORCA, the unpaired agent stops suddenly and does not move again until the other two agents have completely passed it. While this behavior is collision-free, the resulting motion is very unnatural. In contrast, using our Implicit

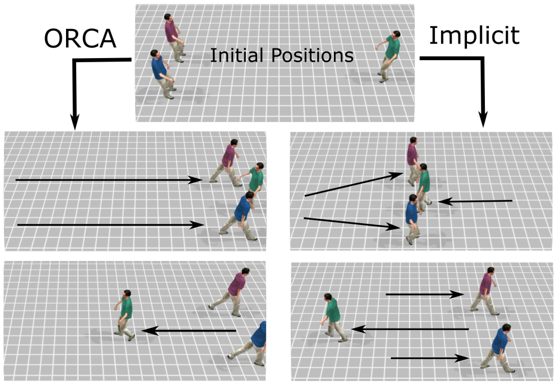


Fig. 10. In this three-agent scenario, ORCA is overly conservative and does not allow the rightmost agent to move forward until the two approaching agents have moved past. Implicit gives more realistic behavior, with all agents adapting plausibly to each other.

approach, the unpaired agent is willing to “take a step” towards a collision based on the assumption that more drastic avoidance maneuvers can be taken in future time steps if needed.

7.2 Relation to Existing Simulation Methods

Velocity-based approaches. Our Implicit crowd simulation approach shares many important aspects with ORCA. Both can be viewed as attempts to solve a constrained optimization problem over the space of agent velocities, and both provide some formal guarantees about robustness and collision-free behavior under certain conditions. However, there are some subtle, but important differences in the two approaches which lead to significant differences in visual behaviors. Consider a pair of agents interacting as shown in Figure 2a. Figure 11 shows the accelerations an agent will undergo for different goal velocities for both methods. For ORCA, any velocity inside the velocity obstacle (VO) is immediately moved outside it in a single time step; velocities outside the VO are unaffected. Due to our smoothing procedure, Implicit has a wider range of velocities which are affected, but the resulting acceleration is much less dramatic. This leads to smoother behaviors which allow collision resolution to happen over several time steps (Figure 10).

Many recent methods for crowd simulation can be considered velocity-based in their approach. Examples include anticipatory social forces [Karamouzas et al. 2009], synthetic vision approaches [Dutra et al. 2017; Hughes et al. 2015; Ondřej et al. 2010], and the recently proposed WarpDriver [Wolinski et al. 2016] which all have agents whose behavior depends primarily on the relative velocities of nearby agents. Unlike ORCA, but similar to PowerLaw, all these methods trade-off collision avoidance guarantees in favor for improved behaviors in situations like 3-agent and Crossing where ORCA does poorly. However, none of these methods provide any guarantees of robustness and collision-free motion, and are therefore liable to suffer from the same time step limitations in dense scenarios as seen with ORCA in Table 1. We conjecture that, subject to the limitations discussed below in Section 7.3, it may be possible to reformulate many of these techniques in terms of interaction energies. This would allow our implicit integration scheme outlined in Section 4.1 to be applied, automatically imparting robustness guarantees to these methods. We believe this is an exciting avenue for future work.

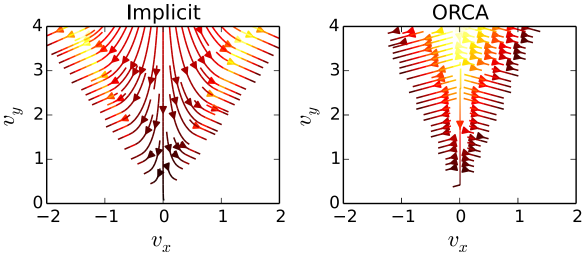


Fig. 11. Comparisons between Implicit and ORCA acceleration responses for the two agents shown in Figure 2a. (Left) The gradient of our proposed interaction potential, equation (19). (Right) The change in velocity induced by ORCA’s VO. ORCA instantaneously avoids the collision, whereas our method smoothly avoids collisions over several time steps.

Continuum Formulations. Continuum-based approaches to crowd simulation [Narain et al. 2009; Treuille et al. 2006] are similar to our work in several ways. They, too, seek to leverage the extensive literature of modern, robust physical simulation techniques to improve crowd simulation. In particular, they also rely on a global solve across the entire crowd to obtain collective behavior and avoid congestion. However, these methods cannot guarantee collision-free motion, as the agents do not interact directly with each other but rather are guided by a continuum approximation of the crowd flow that carries no information about specific nearby agents. Our approach works directly with individual agents and their pairwise interactions, yielding high-quality behavior along with theoretical guarantees.

7.3 Limitations

Backward Euler is a first-order integration scheme and effectively assumes that the agent trajectories are linear over a time step. This is particularly true for the continuous collision detection approach in our repulsion energy. We have seen that our method performs well for moderate time steps of up to 0.4 s, because the true solution does not deviate significantly from linearity over such time intervals. For much larger time steps, the simulation quality will eventually start to degrade as the linear-trajectory assumption becomes increasingly invalid. In the supplementary video we show such a result for the Hallway scenario; when a time step of 4 s is used, the flow of the crowd becomes slower as agents can no longer weave around each other easily.

Unlike many popular techniques for crowd simulation which use only per-agent computations, our method requires a global solve involving all the agents in the crowd. This has a significant computational cost, and limits our performance compared to other non-centralized methods such as ORCA and PowerLaw. Other limitations also arise from the use of a global approach: All agents must be updated simultaneously in our current formulation, so it is not easy to use techniques like adaptive time stepping, variable update rates, or even agents that are updated out of sync.

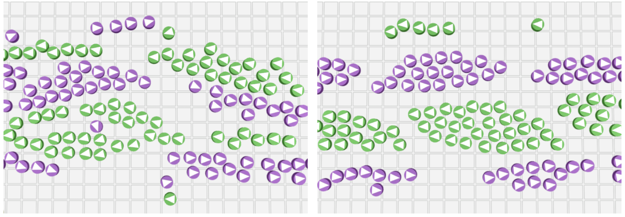

(a) Interaction radius = 2.5m (b) Interaction radius = 10m

Fig. 12. Varying solver parameters such as the maximum interaction radius allowed between agents can change the global behavior of agents. (a) Using a small radius produces less organized behavior, while (b) a larger radius results in more organization.

Despite the robustness of the overall simulation approach, per-scenario tuning may still be required to adjust the behavior of the overall simulation. In addition to the behavioral parameters which already existed in the original PowerLaw model, our model also has tunable parameters associated with the implicit solver that affect the agent behavior. Tuning, these parameters can expose a trade-off between simulation quality and runtime performance. For example, using too few L-BFGS steps per frame will not result in collisions between agents, but can result in velocities too close to their initial value of $\mathbf{0}$, unnaturally damping crowd flows. Likewise, changing the cut-off radius at which agent-agent interactions energies are ignored can improve runtime, but also changes the amount of global coordination between agents (see Figure 12).

Finally, our method requires all velocity-dependent forces in the system to be of the form $\mathbf{f}(\mathbf{x}, \mathbf{v}) = -\nabla_2 R(\mathbf{x}, \mathbf{v})$ for some scalar-valued function R, analogous to how many optimization-based integrators in previous work requires conservative forces $\mathbf{f}(\mathbf{x}) = -\nabla U(\mathbf{x})$. While many nonconservative forces of interest are of our proposed form, it certainly does not include all possible force functions. For example, any interaction energy between two agents i and j will induce forces on both of them, in fact equal and opposite forces if the energy depends on $\mathbf{v}_i - \mathbf{v}_j$. As a consequence of this, asymmetric interactions such as fleeing and leader-following cannot be updated through implicit integration. These interaction forces may be integrated explicitly by evaluating them at the beginning of the time step and treating them as if they were constant. While these explicit forces may be a source for non-smooth motion at large time steps, our approach still ensures collision-free behavior even in the presence of these explicit forces.

8 CONCLUSION

We have shown how a large class of velocity-dependent interactions in multi-agent systems such as crowds can be described using an anticipatory potential, analogous to the Rayleigh dissipation function in classical mechanics, and thus simulated effectively using optimization-based implicit integration schemes. Applying our formulation to the problem of local navigation of multiple virtual agents, we have proposed the Implicit Crowds model. Compared to previous techniques for local collision avoidance, we demonstrate that our approach can guarantee collision-free motions and generate robust simulations with high-quality behaviors at across a large range of densities, time steps, and settings.

Future work. In the future, we would like to address several of the limitations discussed above. Ultimately, it would be desirable to find a local scheme for independently updating each agent's velocity according to equation (10), that would allow us to exploit the use of local-global alternating minimization techniques, such as projective dynamics [Bouaziz et al. 2014], ADMM [Narain et al. 2016], and descent methods [Wang and Yang 2016], which allow for massive parallelization. Another interesting direction for future work is applying our proposed anticipatory potential model to other multi-agent systems besides crowds, such as traffic, although this may require a different pairwise potential formulation than one based on the smoothed time to collision introduced in this paper.

Furthermore, we believe that our method opens up a wide range of new possibilities for how crowd simulations are integrated with larger systems. For example, since the frequency of a human step is about 1.5-2.5 Hz, an update rate of about 0.4 s for the motion of the agent's center of mass would be well suited as input for a footstep-based character animation system. Additionally, supporting large time steps suggests the possibility of adaptive time stepping for crowd simulation in games, where very large time steps could be used for agents far from the player and small time steps for nearby agents. Because our system provides robust and accurate behavior even for relatively large time steps, we believe that it can be naturally integrated into a level-of-detail system without compromising the quality of the simulated crowd's behavior. While the synchronized nature of the global solve currently precludes such applications, we hope that this limitation may be overcome with future work.

Finally, we plan to explore applications of nonlinear dissipation functions to physics-based animation as well. Previous optimization-based integrators in animation [Gast et al. 2015; Kharevych et al. 2006] have demonstrated linear damping forces, essentially by using quadratic dissipation functions. The use of more general dissipation functions would allow nonlinear dissipative forces such as Coulomb friction in solid contact and shear-dependent viscosity in foams, paints, and gels to be incorporated into an optimization-based time integrator. Therefore, the robustness and efficiency benefits of optimization-based integration could be extended to a much broader class of phenomena.

REFERENCES

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, 43–54.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Transaction on Graphics* (2014), 154:1–154:11.

Panayiotis Charalambous and Yiorgos Chrysanthou. 2014. The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation. *Computer Graphics Forum* 33, 8 (2014), 95–108.

Panayiotis Charalambous, Ioannis Karamouzas, Stephen J. Guy, and Yiorgos Chrysanthou. 2014. A Data-Driven Framework for Visual Crowd Analysis. *Computer Graphics Forum* 33, 7 (2014), 41–50.

Teófilo Dutra, Ricardo Marques, Joaquim B. Cavalcante-Neto, Creto Augusto Vidal, and Julien Pettré. 2017. Gradient-based steering for vision-based crowd simulation algorithms. *Computer Graphics Forum* 36, 2 (2017).

Paolo Fiorini and Zvi Shiller. 1998. Motion Planning in Dynamic Environments using Velocity Obstacles. *The International Journal of Robotics Research* 17, 7 (1998), 760–772.

Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-seidel Method for Stable Soft Body Dynamics. *ACM Transaction on Graphics* 35, 6 (2016), 214:1–214:9.

Theodore F. Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M. Teran. 2015. Optimization Integrator for Large Time Steps. *IEEE Transactions on Visualization and Computer Graphics* 21, 10 (2015), 1103–1115.

Herbert Goldstein. 1980. *Classical Mechanics*. Addison-Wesley.

Dirk Helbing, Illés Farkas, and Tamas Vicsek. 2000. Simulating dynamical features of escape panic. *Nature* 407, 6803 (2000), 487–490.

Dirk Helbing and Péter Molnár. 1995. Social Force Model for Pedestrian Dynamics. *Physical Review E* 51 (1995), 4282–4286.

Dirk Helbing, Péter Molnár, Illés J. Farkas, and Kai Bolay. 2001. Self-Organizing Pedestrian Movement. *Environment and Planning B: Planning and Design* 28, 3 (2001), 361–383.

Rowan Hughes, Jan Ondřej, and John Dingliana. 2014. Holonomic Collision Avoidance for Virtual Crowds. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 103–111.

Rowan Hughes, Jan Ondřej, and John Dingliana. 2015. DAVIS: density-adaptive synthetic-vision based steering for virtual crowds. In *Motion in Games*. ACM, 79–84.

Roger L. Hughes. 2002. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological* 36, 6 (2002), 507–535.

Eunjung Ju, Myung Geol Choi, Minji Park, Jehee Lee, Kang Hoon Lee, and Shigeo Takahashi. 2010. Morphable crowds. *ACM Transactions on Graphics* 29 (2010), 140:1–140:10. Issue 6.

Couro Kane, Jerrold E. Marsden, Michael Ortiz, and Matt West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (2000).

Mubbasir Kapadia, Nuria Pelechano, Jan Allbeck, and Norm Badler. 2015. Virtual crowds: Steps toward behavioral realism. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging* 7, 4 (2015), 1–270.

Mubbasir Kapadia, Shawn Singh, Brian Allen, Glenn Reinman, and Petros Faloutsos. 2009. Steerbug: an interactive framework for specifying and detecting steering behaviors. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 209–216.

Mubbasir Kapadia, Shawn Singh, William Hewlett, Glenn Reinman, and Petros Faloutsos. 2012. Parallelized egocentric fields for autonomous navigation. *The Visual Computer* 28, 12 (2012), 1209–1227.

Ioannis Karamouzas, Peter Heil, Pascal van Beek, and Mark H. Overmars. 2009. A Predictive Collision Avoidance Model for Pedestrian Simulation. In *Motion in Games*. 41–52.

Ioannis Karamouzas, Brian Skinner, and Stephen J. Guy. 2014. Universal Power Law Governing Pedestrian Interactions. *Physical Review Letters* 113 (2014), 238701. Issue 23.

Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Transactions on Graphics* 33, 4 (2014), 123:1–123:12.

Liliya Kharevych, Weiwei Yang, Yiying Tong, Eva Kanso, Jerrold E. Marsden, Peter Schröder, and Matthieu Desbrun. 2006. Geometric, Variational Integrators for Computer Animation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 43–51.

Jongmin Kim, Yeongho Seol, Taesoo Kwon, and Jehee Lee. 2014. Interactive Manipulation of Large-scale Crowd Animation. *ACM Transactions on Graphics* 33, 4 (2014), 83:1–83:10.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by example. *Computer Graphics Forum* 26 (2007), 655–664.

Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or. 2010. Context-Dependent Crowd Evaluation. *Computer Graphics Forum* 29, 7 (2010), 2197–2206.

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-spring Systems. *ACM Transactions on Graphics* 32, 6 (2013), 214:1–214:7.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2016. Towards Real-time Simulation of Hyperelastic Materials. *arXiv preprint arXiv:1604.07378* (2016).

Jerrold E. Marsden and Tudor Ratiu. 1999. *Introduction to Mechanics and Symmetry*. Springer.

Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based Elastic Materials. *ACM Transactions on Graphics* 30, 4 (2011), 72:1–72:8.

Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. 2011. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108, 17 (2011), 6884–6888.

Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C. Lin. 2009. Aggregate Dynamics for Dense Crowd Simulation. *ACM Transaction on Graphics* 28, 5 (2009), 122:1–122:8.

Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM $\supseteq$ Projective Dynamics: Fast Simulation of General Constitutive Models. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 21–28.

Jorge Nocedal and Steve J. Wright. 2006. *Numerical optimization*. Springer.

Aline Normoyle, Maxim Likhachev, and Alla Safonova. 2014. Stochastic Activity Authoring with Direct User Control. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 31–38.

Anne-Hélène Olivier, Antoine Marin, Armel Crétual, and Julien Pettré. 2012. Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. *Gait Posture* 36, 3 (2012), 399–404.

Jan Ondřej, Julien Pettré, Anne-Hélène Olivier, and Stéphane Donikian. 2010. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics* 29, 4 (2010), 1–9.

Nuria Pelechano, Jan M. Allbeck, and Norman I. Badler. 2007. Controlling individual agents in high-density crowd simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 99–108.

Stefano Pellegrini, Andrea Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *IEEE International Conference on Computer Vision*. 261–268.

Julien Pettré, Jan Ondřej, Anne-Hélène Olivier, Armel Crétual, and Stéphane Donikian. 2009. Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 189–198.

Craig W. Reynolds. 1987. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21, 4 (1987), 24–34.

Craig W. Reynolds. 1999. Steering Behaviors For Autonomous Characters. In *Game Developers Conference*. 763–782.

Armin Seyfried, Oliver Passon, Bernhard Steffen, Maik Boltes, Tobias Rupprecht, and Wolfram Klingsch. 2009. New insights into pedestrian flow through bottlenecks. *Transportation Science* 43, 3 (2009), 395–406.

Shawn Singh, Mubbasir Kapadia, Billy Hewlett, Glenn Reinman, and Petros Faloutsos. 2011a. A Modular Framework for Adaptive Agent-based Steering. In *ACM Symposium on Interactive 3D Graphics and Games*. 141–150.

Shawn Singh, Mubbasir Kapadia, Glenn Reinman, and Petros Faloutsos. 2011b. Footstep navigation for dynamic crowds. *Computer Animation and Virtual Worlds* 22, 2-3 (2011), 151–158.

Sybren Stuvel, Nadia Magnenat-Thalmann, Daniel Thalmann, A Frank van der Stappen, and Arjan Egges. 2016. Torso Crowds. *IEEE Transactions on Visualization and Computer Graphics* (2016).

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. ACM, 205–214.

Adrien Treuille, Seth Cooper, and Zoran Popović. 2006. Continuum crowds. *ACM Transactions on Graphics* 25, 3 (2006), 1160–1168.

Jur van den Berg, Stephen J. Guy, Ming C. Lin, and Dinesh Manocha. 2011. Reciprocal n-body Collision Avoidance. In *International Symposium of Robotics Research*. 3–19.

Jur van den Berg, Ming C. Lin, and Dinesh Manocha. 2008. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation*. 1928–1935.

Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Transaction on Graphics* 35, 6 (2016), 212:1–212:10.

Marcel Weiler, Dan Koschier, and Jan Bender. 2016. Projective Fluids. In *ACM Motion in Games*. 79–84.

David Wolinski, Ming C. Lin, and Julien Pettré. 2016. WarpDriver: context-aware probabilistic motion prediction for crowd simulation. *ACM Transactions on Graphics* 35, 6 (2016), 164.

Manqi Zhao and Venkatesh Saligrama. 2009. Anomaly detection with score functions based on nearest neighbor graphs. In *Advances in neural information processing systems*. 2250–2258.