

Parameter Estimation and Comparative Evaluation of Crowd Simulations

D. Wolinski¹ S. J. Guy² A.-H. Olivier¹ M. Lin³ D. Manocha³ J. Pettré¹

¹INRIA Rennes - Bretagne Atlantique, France

²University of Minnesota, USA

³University of North Carolina at Chapel Hill, USA

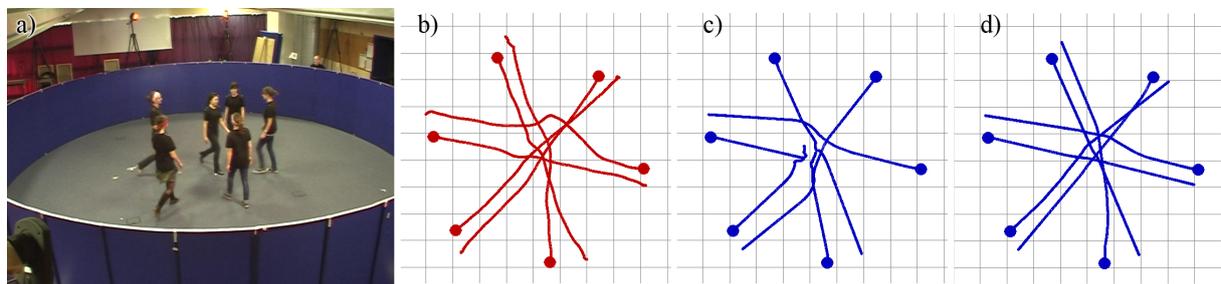


Figure 1: Parameter Optimization Applied to Crowd Data (a) motion capture session for recording reference trajectories for six human agents (b) reference data plot (circles are initial positions) (c) paths taken by simulated agents with default parameters (d) paths taken by simulated agents with optimized parameters. The stock parameters of a simulation model often do not match closely with actual paths humans take in the same situation. Using our parameter optimization technique, the resulting simulation can be made to better match the human trajectories.

Abstract

We present a novel framework to evaluate multi-agent crowd simulation algorithms based on real-world observations of crowd movements. A key aspect of our approach is to enable fair comparisons by automatically estimating the parameters that enable the simulation algorithms to best fit the given data. We formulate parameter estimation as an optimization problem, and propose a general framework to solve the combinatorial optimization problem for all parameterized crowd simulation algorithms. Our framework supports a variety of metrics to compare reference data and simulation outputs. The reference data may correspond to recorded trajectories, macroscopic parameters, or artist-driven sketches. We demonstrate the benefits of our framework for example-based simulation, modeling of cultural variations, artist-driven crowd animation, and relative comparison of some widely-used multi-agent simulation algorithms.

1. Introduction

Creating simulation models of crowds has recently received considerable attention in computer animation, pedestrian dynamics, and virtual reality. Many approaches have been investigated that suggest different techniques to simulate crowds, and a variety of simulation algorithms are known in the literature. These include multi-agent simulation algo-

gorithms that are widely used in computer games, virtual reality, animation, and pedestrian dynamics.

A key research issue in this area is to perform a formal or rigorous evaluation of these algorithms. One widely used criterion is to perform comparative evaluation of simulation algorithms against some real-world reference datasets. However, a major challenge is to estimate the best set of param-

eters for a given algorithm that would result in the *optimal match* with the reference data.

The issue of optimal parameter selection is critical, because most of existing crowd simulation algorithms depend on various parameters and the resulting trajectories or behaviors can vary noticeably based on the choices of parameters. There is no standard way to make comparative evaluation of simulation algorithms. At the same time, data capture of real-world human crowd motion is becoming increasingly ubiquitous. Such datasets can in fact help in describing and analyzing specific crowd phenomena, as well as in calibrating and evaluating crowd simulation models. Given the increase in the number of crowd simulation algorithms and real-world datasets, we need rigorous and automatic techniques to evaluate them.

In this paper, we present a novel framework that can be used to evaluate different crowd simulation algorithms against reference datasets. In this context, we address the problem of computing optimal parameters for a crowd simulation algorithm and present a general scheme that is applicable to a broad class of algorithms and reference datasets. We formulate the evaluation of a simulation algorithm as an optimization problem. First, we find a set of parameters that enables the best match between each simulation algorithm and the reference data. Second, we compare the objective function scores (i.e., distance to reference data) for the given set of algorithms. Our framework is general and capable of supporting a wide range of comparison metrics and simulation techniques.

We illustrate the benefits of our evaluation framework over several existing multi-agent crowd simulation algorithms. Moreover, we consider heterogeneous types of reference datasets: recorded individual trajectories, macroscopic quantities, or even animation sketches. We gather a set of relevant metrics to compare simulated crowds with reference data. We highlight the benefits of parameter estimation by demonstrating its application to example-based simulation and behavior modeling with cultural variation. Our framework is available as an open-source package and can be used by others to evaluate different simulation algorithms and metrics. We demonstrate its performance on many widely-used multi-agent simulations and consider different scenarios with a varying number of agents.

The rest of the paper is organized as follows. In Section 2, we give an overview on related work in crowd simulation, parameter calibration, and algorithm evaluation. Section 3 describes our parameter estimation framework and its key components: algorithms, metrics, reference data, and optimization techniques. Finally, a wide range of concrete examples and applications are presented in section 4 to demonstrate the benefits of our solution.

2. Related Work

Crowd simulation has received significant attention in various disciplines, such as graphics, robotics, and fire safety, which has led to the emergence of numerous simulation algorithms. Specifically, a large variety of microscopic simulation models have been proposed, all of which are based on representing each individual in a crowd (usually as particles or agents) and modeling the interactions between them. Reynolds's seminal Boids model [Rey87] is representative of microscopic approaches: local interactions, which match an individual's speed and orientation to its neighbors', determine the individual's motion and result in large scale emergent behaviors that are visible at the crowd level. Reynolds [Rey99] later extended this work to various kinds of interactions, such as herding and path following. Many physically-inspired models formulate collision-avoidance interactions as repulsive forces between agents [HM95], and the models can be combined with rules to improve behavior [LD04, PAB07]. More recently, velocity-based models have been developed [vPS*08, POO*09, KHvBO09], which reason about an agent's motion in velocity space to compute collision-free trajectories over a short future time window. Other related recent approaches are based on cognitive models [YT07], affordance [KSHF09], and short-term planning using a discrete approach [SKRF11]. Finally, Ondrej et al [OPOD10] use perceptual variables (derived from the virtual optic flow of agents) to compute collision-free motions.

This is only a representative sample, for there are many more algorithms than those listed here, for instance algorithms that operate at the macroscopic level such as [TCP06]. One major problem is developing metrics to compare these dissimilar models; they are based on very different principles, and are qualitatively different as a result. For example, some, like velocity- or vision-based models, enable anticipated avoidance, while others, like force-based models, do not. It is thus very difficult to quantitatively compare these qualitatively different simulation algorithms in order to improve their accuracy.

Another solution is to perform evaluation based on experimental data. In this case, a significant issue is parameter determination for a given model, which is typically computed independently for each agent in these microscopic algorithms. For example, Pettre et al. [POO*09] set parameters of a collision-avoidance model from experimental data capturing avoidance motion based on Maximum Likelihood Estimation. Lerner et al. compare local decision taken from local context between simulations and real data in [LCSCO09]. More complex situations were considered in [LJK*12]: experimental data that captured both the microscopic and macroscopic features of pedestrian motion was used to calibrate and compare various approaches.

Learning methods were also used to calibrate models from computer vision data [PESVG09], or to learn model parameters from real-world data [CC10, JCP*10]. Some ap-

proaches also use real data as a basis to drive agents' motion [LCHL07, LCL07, KHHL12]. These techniques are intrinsically realistic in a sense, given that they are based on real data. However, these approaches are limited to rather specific scenarios. Also, realism of portions of trajectories does not mean that the composition of data is realistic at larger scales. In comparison, we present a technique to perform parameter estimation in a general, model-independent manner. We also consider various types of reference datasets, which take into account measurements of macroscopic parameters and artist-driven sketches, as well as recorded trajectories.

The most direct precursors to our work are previous efforts to evaluate crowd simulations or multi-agent steering methods. Kapadia and colleagues proposed a framework that evaluates the steering approaches' ability to perform navigation tasks in various scenarios; the framework evaluates performance in terms of path smoothness or the number of collisions [SKFR09, KWS*11]. Our approach also shares some resemblance with another set of approaches that tend to compare generated trajectories with real-world data. The entropy-based metric proposed by Guy et al. [GvdBL*12] enables such comparisons with real-world trajectories and is robust with respect to the chaotic nature of individual motion. Such metrics can be used to compare the results of simulation methods to either reference data or user-defined goals. However, these methods do not explicitly address the issue of finding a global optimum solution over all possible simulation parameters. We propose to unify these tasks by posing crowd evaluation as a parameter optimization problem, and we demonstrate its benefits via improved behavior or performance optimization in crowd simulation tests.

3. Optimization Framework

Our framework brings together three components: crowd simulation algorithms, reference data, and a set of metrics for measuring how well the simulation algorithm matches the data. This section provides the details of our parameter optimization algorithm and gives an overview of different simulation models, data sources, and comparison metrics used in our evaluations. We also give a brief summary of the global optimization techniques used in our work. Additional information, including a more detailed description of our metrics and optimization algorithms as well as a comparison of these optimization algorithms can be found in the appendix at gamma.cs.unc.edu/CParameter.

3.1. Overview of Approach

We define a crowd simulator as an algorithm which takes a collection of agent states (i.e., positions and velocities of agents in the crowd) and produces a new set of agent states representing the movement of the crowd over a timestep Δt . We introduce the following notation for specifying a crowd

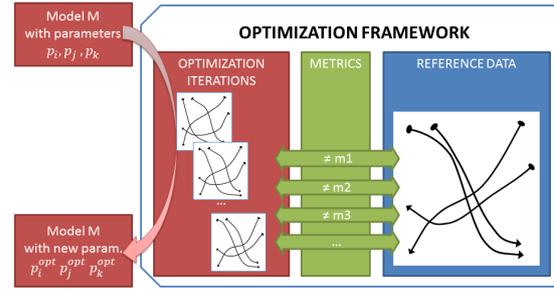


Figure 2: System Overview Our approach optimizes simulation parameters to match target data. Our framework has 3 components: an optimization technique, metrics, and reference data.

simulation algorithm: let k be a given timestep, then \mathbf{x}_k will represent the positions of all agents at timestep k , \mathbf{v}_k the velocity of all agents, and \mathbf{g} the goals of all agents. We can then formally define a crowd simulation algorithm f as follows:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}). \quad (1)$$

In general, a crowd simulation algorithm may have several tunable parameters that affect the behavior of an agent computed by the simulator. Common examples of parameters include an agent's preferred speed or some notion of personal space. While the exact nature of the agent parameters are specific to each algorithm, our framework assumes that these parameters can be defined separately for each agent. Given an agent i , we use \mathbf{p}_i to denote the current parameter set for that agent, and $\mathbf{p} = \{\mathbf{p}_1 \dots \mathbf{p}_n\}$ to denote the vector of parameters over all n agents.

We can now introduce the notion of a *Parameterized Crowd Simulation* as an algorithm f where crowd parameters are part of the input. Formally, for each timestep k :

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}, \mathbf{p}). \quad (2)$$

Simulation Models Several common crowd simulation models fit the form described by Equation (2). We focus on five widely used agent-based simulation algorithms:

1. In the *Boids* model [Rey99], f is a function of the agents' position at some specified future time (current time plus constant). When the predicted distance between agents gets too low, a separation force is computed and added to an attraction force which is pulling towards the agent's goal. Parameters are: radius (size of 2D circle agents) and comfort speed (i.e., speed when no interactions occur).
2. In the *Helbing Social Force* model [HFV], f is a function of the agents' positions. Repulsive forces are computed between agents and combined with attraction forces toward goals. Parameters are: radius and comfort speed.
3. In the *RVO2 model* [VDBGLM11], which computes an

agent's admissible velocity space (space which remains collision-free in a future time window), f returns the optimal admissible velocity. Parameters are: comfort speed, neighbor distance (only agents within this distance are considered for local interactions), radius, and time horizon (only future collisions within this horizon are considered for local interactions).

4. In the *Synthetic Vision* model [OPOD10], which is based on principles from human cognition and visual navigation, f is a function of perceptual variables derived from synthetic optic flow. Parameters are: comfort speed and (a, b, c) , which define a threshold function; perceived values under this threshold are considered for local interactions.
5. In the *Tangent* model [POO*09], which works in the velocity space and considers possible perception errors, f returns the optimal admissible velocity. Parameters are: comfort speed, radius and two error-quantifying parameters.

The parameter set \mathbf{p} for each of these models can be found in Appendix E.

Given a parameterized crowd simulation (Eqn. (2)), our goal is to find a parameter set \mathbf{p}^{opt} which leads to the closest match between a model and some user-defined reference data, which can vary per timestep \mathbf{z}_k . Over all timesteps m , we can define the reference data as follows:

$$\mathbf{z} = \bigcup_{k=1}^m \mathbf{z}_k. \quad (3)$$

In the same way, we can define a complete simulation as all states of a simulator initialized with the reference data:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} = f(\mathbf{z}, \mathbf{p}) = \bigcup_{k=1}^m f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}, \mathbf{p}), \quad (4)$$

initialized with $\mathbf{x}_1 = \mathbf{z}_1$, $\mathbf{v}_1 = \text{speed}(\mathbf{z}_1)$ and $\mathbf{g} = \mathbf{z}_m$.

Given this complete simulation and reference data, plus some user defined distance metric, $\text{dist}()$, we can formally define our framework as computing

$$\mathbf{p}^{\text{opt},f} = \underset{\mathbf{p}}{\text{argmin}} \text{dist}(f(\mathbf{z}, \mathbf{p}), \mathbf{z}), \quad (5)$$

where $\mathbf{p}^{\text{opt},f}$ is the parameter set which matches the reference data closest for a given simulation method f .

In general, the optimization problem that we propose in Equation (5) is very high-dimensional ($\text{dim}(\mathbf{p}) = \sum_{i=1}^n \text{dim}(\mathbf{p}_i)$), making it difficult to optimize consistently across a wide range of similarity metrics, reference data, and simulation methods. We describe our optimization algorithm in Section 3.3, which is designed to deal with such high-dimensional problems.

Once an optimal parameter set \mathbf{p}^{opt} has been computed, we can fairly compare two different simulation methods, f_1 and f_2 , by examining their optimal distance from the reference data. Formally, we declare simulation method f_1 better

than simulation method f_2 if and only if:

$$\text{dist}(f_1(\mathbf{z}, \mathbf{p}^{\text{opt},f_1}), \mathbf{z}) < \text{dist}(f_2(\mathbf{z}, \mathbf{p}^{\text{opt},f_2}), \mathbf{z}). \quad (6)$$

3.2. Optimization Metrics

The role of reference data is to provide a description of the desired behavior or motion trajectories that the simulation should generate. This data can either come from measurements of real motion (e.g., from an overhead camera or motion-capture devices) or can be generated. Generated data can come from artists (flow fields) or some other high-level simulation algorithms. The function $\text{dist}()$ in Equation (5) should capture how close a simulation state comes to matching the reference data. The exact representation of $\text{dist}()$ depends on the nature of the reference data and on the features of the data which the user considers most salient for his or her application.

At a high level, there are two fundamentally different types of reference data that can be used in Equation (5):

- *microscopic data*, which specifies the exact trajectory of each agent in the data, and
- *macroscopic data*, which describes aggregate measures of the overall crowd motion.

Below we briefly describe various metrics that can be easily used in our framework, a more detailed description as well as their mathematical representation can be found in Appendix B

Microscopic Data Metrics

- *absolute difference metric (D)* computes the total distance in position over all agents over all timesteps,
- *path length metric (L)* compares the difference in total length traveled between agents in the reference data and the simulated agents,
- *inter-pedestrian distance metric (I)* compares the difference in average distance (as a 2-norm) between every pair of agents,
- *progressive difference metric (P)* measures the absolute difference between the simulated agents and the reference data *when the simulation is reinitialized at each timestep*.

Macroscopic Data Metrics

- *vorticity metric (V)* measures the vorticity (as defined in fluid mechanics) of the crowd flow,
- *fundamental diagram metric (F)* compares the speed of an agent to the density of agents in its location. This metric is inspired by the field of pedestrian dynamics, where it is commonly used to measure pedestrian flow rates (e.g., [CM12]).

3.3. Optimization Techniques

Once a user has chosen the reference data and an appropriate optimization metric, Equation (5) can be optimized

using different combinatorial optimization algorithms. Because several parameters need to be chosen for every agent in the simulation, the result is a very high-dimensional search problem (hundreds of dimensions) and the complexity of finding an optimal solution is very high. We have analyzed three different widely-used optimization techniques that can be applied to these high-dimensional search spaces.

Each optimization technique uses a different strategy to sample plausible parameters for each agent, maximizing the match of the simulation algorithm to the reference data. All three methods proceed by choosing perspective values for the per-agent simulation parameters from a user-specified domain of reasonable values. A list of the parameter distributions used in our experiments is given in Appendix E.

Greedy approach (G): this approach works by replacing one parameter from \mathbf{p} at a time for each agent. If this replacement lowers the optimization function, the new parameter value is chosen; if not, the previous value is restored. This method can get stuck in local minima.

Simulated annealing (SA): a variant of the greedy approach, this approach attempts to avoid local minima by occasionally using new parameter sets that are “worse” (have higher value of the optimization function) than the old ones [KJV83]. The likelihood of accepting a worse parameter set decreases over time. Given an unlimited amount of time, SA will compute the global minimum.

Genetic algorithm (GA): methods based on genetic algorithms also seek to avoid local minima, and do so by maintaining a pool of parameters that can lead to different local minima [Hol92]. New pools of parameters are computed by combining and modifying previously successful candidates.

Covariance Matrix Adaptation (CMA) [HHOO96]: a solution-pool based method similar to GA, which generates new solutions from distributions defined by a covariance matrix that is adapted at each iteration.

After a comparison in terms of convergence and time of convergence, a combination of genetic and greedy algorithms (GA+G) has been chosen as offering the best compromise between score optimization and runtime performance. While there is no estimation of how close they come to the real optima, the simulated annealing method should provide a good indication as it can theoretically find a global optimum given an infinite computation time. More results on this comparison can be found in Appendix D; pseudocode for the above methods can be found in Appendix C.

Other global optimization techniques, such as Particle Swarm Optimizations (PSO) [PKB07] or the adjoint method [MTPS04], can be applied to optimize Equation (5). Multiple methods can be combined or applied sequentially.

4. Results

The primary benefit of this framework is its generality: it can automatically find the best parameters for any simulation algorithm, based on any metric, for any reference data. In this section, we highlight some advantages and benefits of our parameter-optimization and the framework. First, we present different types of ground-truth data, which are a basis for model comparison, along with examples of model comparisons for each type. Second, we present scores obtained by our implementations of the Boids-like, Social-force and RVO2 models, in the form of a benchmark chart and its analysis.

4.1. Data Categories

Microscopic data, 2-5 agents This data category regroups various cases of 2-5 pedestrians crossing ways. The following are two visual examples showing model comparisons based on this data using the Difference metric:

- a simple crossing scenario between two pedestrians. With default parameters, none of the algorithms correctly replicate the avoidance strategy of the pedestrians; after calibration, only the Vision-based algorithm can reproduce it (Figure 3(a)). This is consistent with its goal, which is to reproduce human-like reactions to impending obstacles.
- a scenario where trajectories are correct but the agents are ill-synchronized with the real pedestrians, i.e. their speeds along the trajectories are incorrect. After calibration, the Tangent simulation model [POO*09] gives a better speed profile than the other algorithm, because its agents accelerate and decelerate when needed and are better synchronized with pedestrians (Figure 3(b)).

Microscopic data, 6-24 agents This data category is similar to the previous, except that more pedestrians are present (6-24) and they are organized into circles, with their goal being to get to the antipodal positions. Here are two examples of comparisons:

- six pedestrians case: before calibration all trajectories are far from the real ones; after calibration, RVO2 trajectories are near-identical to real-world data (Figure 3(c)).
- twenty-four pedestrians case: after calibration, Social-force agents agglomerate in the center without anticipation, while RVO2 agents anticipate future collisions and are spread in a pattern more similar to that of the real pedestrians (Figure 4).

Microscopic data, ~150 agents This data [PCBS11] contains ~150 people in two groups crossing ways. Due to the nature of this data (pedestrians constantly appearing and disappearing from the cameras’ field of view), a time-window is extracted and only a subset of all simulated pedestrians are evaluated (their positions and velocities are then known during the entirety of the time-window; the more numerous

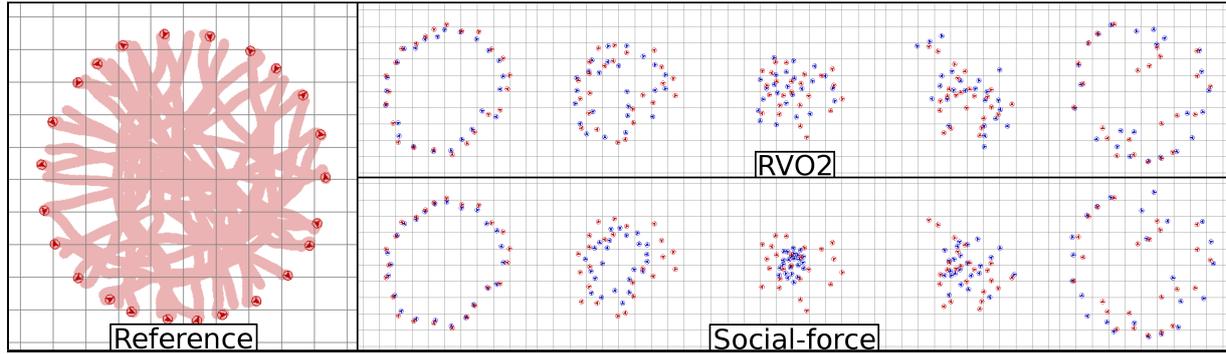
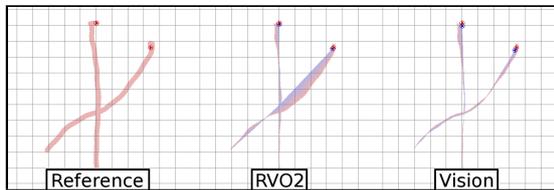
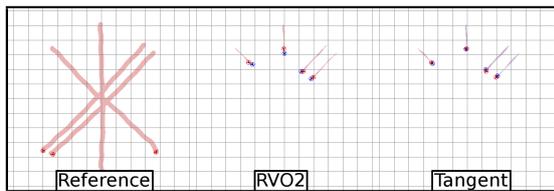


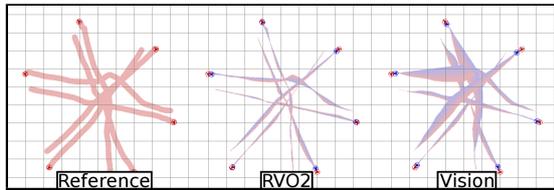
Figure 4: Example of a result of the calibration of two models with the Difference metric in the case of 24 pedestrians in a circle with the goal to cross said circle. Left: reference data trajectories (ending positions in bright). Top row: five consecutive positions of real (red) and simulated (blue) agents in the case of the RVO2 model. Bottom row: same for the Social-force model. With the lack of anticipation, the Social-force model's agents tend to agglomerate in the center before solving their interactions. On the contrary, the RVO2 model's agents anticipate interactions and spread in a way more similar to the original data.



(a) Pair-wise crossing.

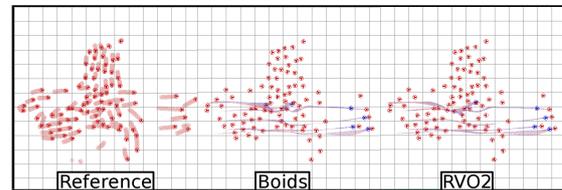


(b) Four-agent crossing.

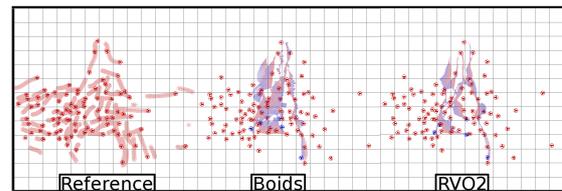


(c) Six agents in a circle going for antipodal positions.

Figure 3: Examples of calibration results of algorithms with the Difference metric in different scenarios, which enables their fair comparison. Colored area represents error between real (red) and simulated (blue) trajectories identifiable through the color gradient. Left: reference data trajectories (ending positions in bright). (a) Middle: RVO2, incorrect trajectories. Right: Vision-based model, correct trajectories. (b) Middle: the RVO2 model's agents are ill-synchronized with real pedestrians. Right: Tangent correctly synchronizes agents with pedestrians. (c) Middle: RVO2; copes well with these more complex interactions. Right: Vision-based model; incorrect paths.



(a) Five controlled agents (blue) amidst a flow of 111 agents.



(b) Five controlled agents (blue) amidst a flow of 152 agents.

Figure 5: Example of a result of the calibration of two models with the Difference metric in the case of 150 agents; 5 of them are controlled by simulation algorithms. Colored area represents error between real (red) and simulated (blue) trajectories identifiable through the color gradient. Left: reference data; only recent trajectories are shown for clarity (ending positions in bright). Middle: Boids-like. Right: RVO2. (a) RVO2 leads to a slightly increased error compared to the Boids-like model. (b) RVO2 leads to a decreased error compared to Boids-like.

the controlled pedestrians are, the shorter the time-window is). Here are two examples:

- 111 agents total, 5 controlled, traveling from left to right: RVO2 leads to slightly higher errors than the Boids-like model (Figure 5(a)).
- 152 agents total, 5 controlled, traveling from left to right: RVO2 leads to lower errors than the Boids-like model (Figure 5(b)).

Macroscopic data In many scenarios, the goal is not to match a specific motion, but to produce an overall characteristic flow. Our framework can be used in these scenarios. For example, there are well-documented cultural differences between the flow rates of Germans and Indians [CSC09]. These cultural flow differences can be described by two different fundamental diagrams.

We can match these fundamental diagrams in different situations with different simulation methods. To this end, we consider the fundamental diagrams of German and Indian people as described in [CSC09] and, in the case of 30-pedestrian crowds, constrain the Boids-like, Social-Force and RVO2 models to follow them. To constrain these models to the fundamental diagrams, we set them to 6 consecutive density-velocity points on each diagram (at 0.75, 1.0, 1.25, 1.50, 1.75 and 2.0 pedestrians per square meter).

The diagrams obtained after merging the data for these 6 points are represented on Figure 6, along with the original data found in [CSC09].

These diagrams are very useful in setting up evacuation scenarios and adapting them to different cultures. Additionally, the framework can also help decide which model is best suited to a task. For example, here the Boids-like algorithm gets easily stuck in the corridor, and the resulting diagrams are far from the original data. However, the Social-force and RVO2 algorithms fit the data well; RVO2 ultimately matches the fundamental diagrams better at higher densities. This is largely due to the Social-force agents displaying instabilities near walls at high densities (as seen in Figure 7).

Sketch-like data Our framework also has a broader application as a metric-driven animation tool for artists animating crowd scenes. If an animator provides a rough idea of a motion (in the form of a metric), our framework can be used to indicate which is the best algorithm to generate the animation, and can also provide the best parameters for the task. This spares the animator the tedious task of setting each agent’s trajectory individually or building new mechanics into the model he is using. Additionally, this process is independent of the simulation algorithm. We provide three examples below. The first example is a group of pedestrians that are made to walk close to each other, then separate, and finally regroup. In order to simulate such behavior, we define three zones based on waypoints, where a distance metric is applied to determine pedestrian distance from one another. This metric is then used to maintain a low inter-agent distance in the first zone, a higher distance in the second zone, and a lower distance in the last zone. Figure 9 shows the resulting animations at various stages for the RVO2 algorithm.

The second example involves the vorticity metric used to create vortex-like patterns. Figure 8 shows results obtained with Boids and RVO2. Here, the Boids-like model lacks anticipation and fails to completely recreate the wanted behavior; RVO2 is more successful, thanks to anticipation.

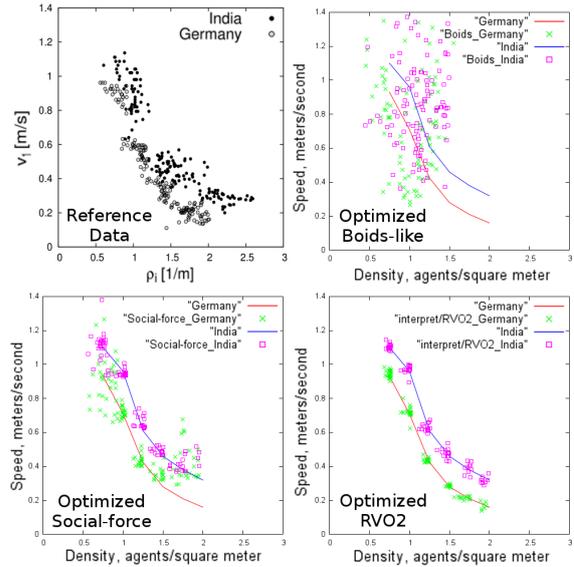


Figure 6: Cultural variation in fundamental diagrams [Chattaraj et al. 2009]. Top left: original data; top right: Boids-like model fit to the fundamental diagrams; bottom left: Social-force; bottom right: RVO2. Boids poorly fits the data while Social-force and RVO2 models offer better matches. Social-force agents travel too fast at high densities compared to RVO2.

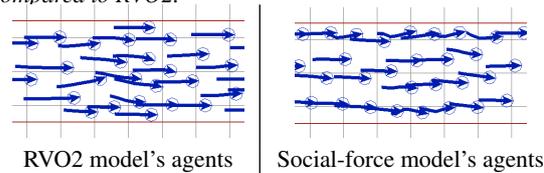


Figure 7: High density (2 agents per square meter) illustration (“tails” indicating recent movement) after calibrating to a fundamental diagram. RVO2 agents on the left go straight. Social-force agents on right become unstable and bounce off of walls causing them to accelerate (explains the excess speed in the fundamental diagram in Figure 6).

Figure 10 shows another example with three corridors. Two groups enter through two corridors and exit as one group through a third corridor. They are made to be sparse when entering and dense when exiting.

4.2. Benchmarks

Figure 11 summarizes scores obtained by the Boids-like, Social-force and RVO2 algorithms for the various data categories in a benchmark fashion. These three algorithms were chosen for this benchmarking analysis as they are broadly applicable, make few assumptions about the scenarios being used in, and are representative of the types of simulation strategies commonly found in games and VR.

Examining the results, some trends can be seen to emerge

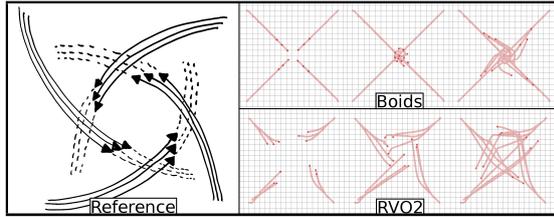


Figure 8: Sketch-based simulation. Left: desired trajectories, four groups forming a vortex-like pattern. Top row: optimized solution for the Boids-like model. Bottom row: same for RVO2. Boids only offers radius and speed parameters, and doesn't include anticipation, it is unable to maintain significant distances between groups of agents without breaking them apart. RVO2 anticipates over a parameterized time horizon, allowing it to keep groups distant while maintaining members of a same group close.

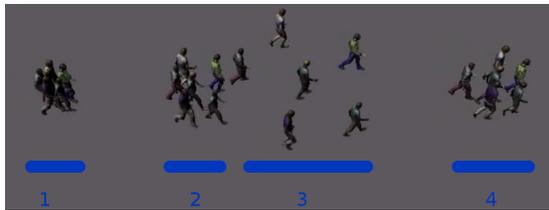


Figure 9: Sketch-based simulation. RVO2 agents are made to travel grouped, then disperse and then regroup again. Four consecutive states are shown.

across various dataset and metrics. For example, across many datasets and metrics the RVO2 algorithm tends to lead to better scores than our simple Boids-style simulation and Social-force models. This is likely due to RVO2 being the only one of the three methods to incorporate predictive collision avoidance. In fact, as the number of agents (and complexity of the scenario) increased, the advantage of RVO2 decreased significantly. This trend can be seen most clearly in the difference metric. The convergence of the performance is expected, in part, because there is little room to anticipate trajectories in dense scenarios with 100s of individuals in close quarters. RVO2 and the Social-force model score similarly in the Fundamental diagram metric for similar reasons.

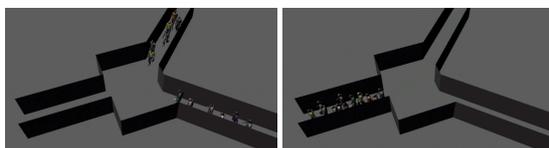


Figure 10: Sketch-based simulation where two sparse groups enter through two corridors and merge into a dense group when exiting through a third corridor. Two consecutive states are shown.

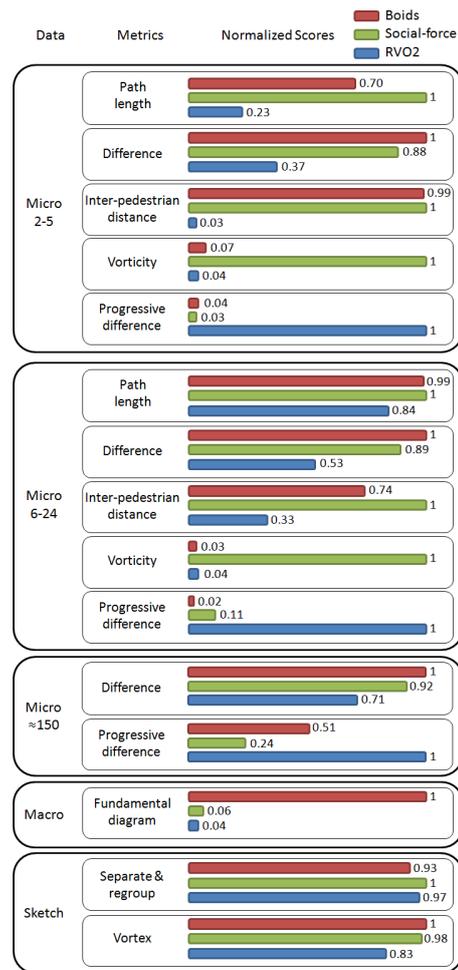


Figure 11: Benchmarks. Comparative scores (lower is better) of the Boids-like (red), Social-force (green) and RVO2 (blue) models in five data categories: microscopic data with 2-5 agents, microscopic data with 6-24 agents, microscopic data with ~150 agents, macroscopic, and sketch-based.

Applying our framework to sketch-like data can also reveal interesting aspects of the simulation techniques. For example, the smooth group behaviors produced by the Boids-like simulation work well in capturing the desired behavior in separate & regroup benchmark. In contrast, the vortex scenario shows how RVO2's anticipation can be used to create novel behaviors difficult for the other methods such as keeping high distances between groups of agents while keeping agents belonging to a same group close to each other.

5. Analysis and Conclusions

In this paper, we have addressed the problem of comparing various crowd simulation algorithms. We have formulated the estimation problem in a generalized way as an optimization

tion problem. We have optimized the set of simulation parameters to provide the closest match between the simulation results and target reference data provided by users. We have implemented several comparison metrics to evaluate various aspects of simulation results. The resulting framework can be widely applied and supports different optimization metrics to match a user's target application.

Our approach has several important properties. First, our technique is very generally applicable. In this paper we have shown our ability to use the framework with a variety of simulation models (force-based, rule-based, velocity-based, etc.); a variety of types of metrics (microscopic, macroscopic); a variety of reference data (real data, example trajectories, macroscopic measures, sketch-based, etc.). Second, we have explored the problem in its full dimensionality; each agent given individualized parameter values while maintaining reasonable computing requirements. Finally, we have demonstrated the applications of our framework in various contexts.

Our results demonstrate the importance of parameter estimation: the same model can show very different behaviors depending on the parameters of the simulation. It is therefore crucial that researchers account for the effects of parameters when evaluating and comparing simulation models. Our framework has sought to address this question in a generalized way, and we hope this contribution opens various perspectives for future work. We would first like to add more models and more metrics to the framework (and try combining our approach with [SKFR09, KWS*11]). This would facilitate the use of the framework by the research community at large and facilitate efforts for more exhaustive comparisons for different simulation methods. We have also demonstrated the applicability of this framework to the creation of fully animated simulations from high-level specifications of desired behavior. Such cases were previously handled by means of scripts or tedious waypoint sequences that had to be defined by animators. We have shown that our parameter-estimation approach can alleviate some of this tedious work to assist in crowd animation. Further exploration of this area, perhaps with human subject studies, is a very promising research direction.

Limitations Along with the above contributions, our method has some limitations still to be addressed. Most importantly, because a simulation's parameter-space is so high-dimensional, large scenarios with hundreds of agents are still time-consuming. It is not easy to estimate the complexity of our framework because of all the components involved. One metric evaluation step is usually equivalent to running a whole simulation based on the reference data and then comparing the results with the data. Our framework is thus very dependent on the complexities of the metrics and simulation algorithms as well as reference data. For instance, a longer time-window in the reference data or the lack of an accelerating structure (e.g. kd-tree) in the simulation algorithm's im-

plementation can greatly impact the time an evaluation call takes. As for optimization algorithms, only the complexity (in terms of evaluation calls) of an iteration can be theoretically estimated: $O(nm)$ for the greedy algorithm and simulated annealing (n number of agents, m number of parameters per agent); constant (solution pool size) for the genetic algorithm. Some indicative times can be found in Figure 14 from Appendix D. It is possible to address the question of performance and scalability by using parallel versions of optimization algorithms and simulators (this is less of a problem when evaluating multiple simulation algorithms on multiple scenarios which is easily parallelizable - e.g. one thread per scenario).

We would also like to study how parameters can be generalized beyond a given scenario or metric: can a certain set of parameters work well across various scenarios? Can a "style" be exported and kept for much larger crowds? In situations where there is not a single metric or a single piece of reference data to fit, it would be desirable to find parameter sets that are "good enough" in some high-level sense, or a set of scenarios that are semantically related and provide a good coverage of the problem domain. Such a goal will likely involve user studies and be greatly affected by perceptual factors. Finally, a key point in future studies will have to focus on the metrics. This would include determining which metrics (or combinations of metrics) are best adapted to various scenarios. But also defining metrics that are immune to the variability of agent behavior to similar conditions and that are able to capture high-level aspects that remain consistent across different data.

We hope this framework will prove to be an important step in the direction of standardized assessment of crowd-simulation models. By providing fair comparisons between methods, a standardized means of analysis will help forward the field of parameterized crowd simulation as a whole.

Acknowledgements

This project is funded by the French Research Agency ANR, grant JCJC PERCOLATION ANR-13-JS02-0008. It is further supported by NSF awards 100057, 1117127, 124124456, 1305282, and a grant from the Boeing Company. The authors would also like to thank the Golaem company for their help preparing parts of the companion video.

References

- [CC10] CHARALAMBOUS P., CHRYSANTHOU Y.: Learning crowd steering behaviors from examples. In *Proceedings of the Third international conference on Motion in games* (Berlin, Heidelberg, 2010), MIG'10, Springer-Verlag, pp. 35–35. 2
- [CM12] CURTIS S., MANOCHA D.: Pedestrian simulation using geometric reasoning in velocity space. In *Pedestrian and Evacuation Dynamics (PEDS)* (2012). 4, 11
- [CSC09] CHATTARAJ U., SEYFRIED A., CHAKROBORTY P.: Comparison of pedestrian fundamental diagram across cultures. In *Advances in Complex Systems* (2009), pp. 393–405. 7, 11

- [Fri37] FRIEDMAN M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32, 200 (1937), 675–701. 13
- [GvdBL*12] GUY S. J., VAN DEN BERG J., LIU W., LAU R., LIN M. C., MANOCHA D.: A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 190. 3
- [HFV] HELBING D., FARKAS I., VICSEK T.: Simulating dynamical features of escape panic. *Nature* 407, 487–490. 3
- [HHOO96] HANSEN N., HANSEN N., OSTERMEIER A., OSTERMEIER A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. Morgan Kaufmann, pp. 312–317. 5, 12
- [HM95] HELBING D., MOLNAR P.: Social force model for pedestrian dynamics. *Physical Review E* 51 (1995), 4282. 2
- [Hol92] HOLLAND J. H.: Genetic algorithms. *Scientific american* 267, 1 (1992), 66–72. 5
- [JCP*10] JU E., CHOI M. G., PARK M., LEE J., LEE K. H., TAKAHASHI S.: Morphable crowds. *ACM Trans. Graph.* 29, 6 (Dec. 2010), 140:1–140:10. 2
- [KHHL12] KIM M., HWANG Y., HYUN K., LEE J.: Tiling motion patches. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics conference on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2012), EUROSCA'12, Eurographics Association, pp. 117–126. 3
- [KHvBO09] KARAMOUZAS I., HEIL P., VAN BEEK P., OVERMARS M. H.: A predictive collision avoidance model for pedestrian simulation. In *Motion in Games* (2009), pp. 41–52. 2
- [KJV83] KIRKPATRICK S., JR. D. G., VECCHI M. P.: Optimization by simulated annealing. *science* 220 (1983), 671–680. 5
- [KSHF09] KAPADIA M., SINGH S., HEWLETT W., FALOUTSOS P.: Egocentric affordance fields in pedestrian steering. In *Proc. of the 2009 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2009), I3D '09, ACM, pp. 215–223. 2
- [KWS*11] KAPADIA M., WANG M., SINGH S., REINMAN G., FALOUTSOS P.: Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proc. of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 53–62. 3, 9
- [LCHL07] LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: a data-driven approach to crowd simulation. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symp. on Computer animation* (2007), pp. 109–118. 3
- [LCL07] LERNER A., CHRYSANTHOU Y., LISCHINSKI D.: Crowds by example. *Computer Graphics Forum (Proceedings of Eurographics)* 26, 3 (2007). 3
- [LCSCO09] LERNER A., CHRYSANTHOU Y., SHAMIR A., COHEN-OR D.: Data driven evaluation of crowds. In *Proc. of the 2nd International Workshop on Motion in Games* (Berlin, Heidelberg, 2009), MIG '09, Springer-Verlag, pp. 75–83. 2
- [LD04] LAMARCHE F., DONIKIAN S.: Crowds of virtual humans : a new approach for real time navigation in complex and structured environments. *Eurographics'04: Computer Graphics Forum* 23, 3 (September 2004), 509–518. 2
- [LJK*12] LEMERCIER S., JELIC A., KULPA R., HUA J., FEHRENBACH J., DEGOND P., APPERT-ROLLAND C., DONIKIAN S., PETTRÉ J.: Realistic following behaviors for crowd simulation. *Comp. Graph. Forum* 31 (May 2012), 489–498. 2
- [MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. In *ACM Transactions On Graphics (TOG)* (2004), vol. 23, ACM, pp. 449–456. 5
- [OPOD10] ONDŘEJ J., PETTRÉ J., OLIVIER A.-H., DONIKIAN S.: A synthetic-vision based steering approach for crowd simulation. *ACM Trans. Graph.* 29, 4 (July 2010), 123–123. 2, 4
- [PAB07] PELECHANO N., ALLBECK J. M., BADLER N. I.: Controlling individual agents in high-density crowd simulation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 99–108. 2
- [PCBS11] PLAUE M., CHEN M., BÄRWOLFF G., SCHWANDT H.: Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings. In *Proc. of the 2011 ISPRS conference on Photogrammetric image analysis* (Berlin, Heidelberg, 2011), PIA'11, Springer-Verlag, pp. 285–296. 5
- [PESVG09] PELLEGRINI S., ESS A., SCHINDLER K., VAN GOOL L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th Int. Conf. on* (29 2009–oct. 2 2009), pp. 261–268. 2
- [PKB07] POLI R., KENNEDY J., BLACKWELL T.: Particle swarm optimization. *Swarm intelligence* 1, 1 (2007), 33–57. 5
- [POO*09] PETTRÉ J., ONDŘEJ J., OLIVIER A.-H., CRETUAL A., DONIKIAN S.: Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, ACM, pp. 189–198. 2, 4, 5
- [Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87* (New York, NY, USA, 1987), ACM, pp. 25–34. 2
- [Rey99] REYNOLDS C. W.: Steering behaviors for autonomous characters. In *Game Developers Conference 1999* (1999). 2, 3
- [sha] Shark. http://image.diku.dk/shark/sphinx_pages/build/html/index.html. 12
- [SKFR09] SINGH S., KAPADIA M., FALOUTSOS P., REINMAN G.: An open framework for developing, evaluating, and sharing steering algorithms. In *Proceedings of the 2nd International Workshop on Motion in Games* (Berlin, Heidelberg, 2009), MIG '09, Springer-Verlag, pp. 158–169. 3, 9
- [SKRF11] SINGH S., KAPADIA M., REINMAN G., FALOUTSOS P.: Footstep navigation for dynamic crowds. In *Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2011), I3D '11, ACM, pp. 203–203. 2
- [TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 1160–1168. 2
- [VDBGML11] VAN DEN BERG J., GUY S. J., LIN M., MANOCHA D.: Reciprocal n-body collision avoidance. In *Robotics Research*. Springer, 2011, pp. 3–19. 3
- [vPS*08] VAN DEN BERG J., PATIL S., SEWALL J., MANOCHA D., LIN M.: Interactive navigation of individual agents in crowded environments. *Symposium on Interactive 3D Graphics and Games (I3D 2008)* (2008). 2
- [Wil45] WILCOXON F.: Individual comparisons by ranking methods. *Biometrics bulletin* 1, 6 (1945), 80–83. 13
- [YT07] YU Q., TERZOPOULOS D.: A decision network framework for the behavioral animation of virtual humans. In *Symposium on Computer animation* (2007), pp. 119–128. 2
- [ZKSS12] ZHANG J., KLINGSCH W., SCHADSCHNEIDER A., SEYFRIED A.: Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. *Journal of Statistical Mechanics: Theory and Experiment* 2012, 02 (2012), P02002. 13