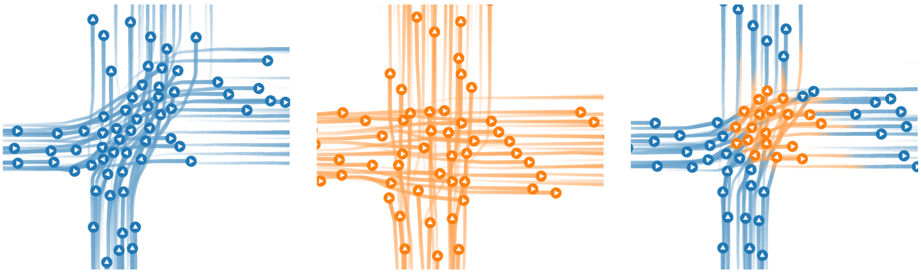
DOI: 10.1111/cgf.14469
EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

Volume 41 (2022), Number 2

Dynamic Combination of Crowd Steering Policies Based on Context

B. Cabrero-Daniel[1], R. Marques[2], L. Hoyet[3], J. Pettré[3], and J. Blat[1]

[1]Universitat Pompeu Fabra
[2]Universitat de Barcelona
[3]Inria, Univ. Rennes, CNRS, IRISA



(a) *PL* ($S_{QF} = 67$) **(b)** *RVO* ($S_{QF} = 84$) **(c)** *PL and RVO* ($S_{QF} = 92$)

Figure 1: *Simulation of two agent flows crossing at 90º using three different motion strategies, and their respective score S_{QF} according to a quality function. (a) Agents are steered using the Power Law model (PL, blue). Note that some agents are dragged along a diagonal towards the up-right direction, hence deviating them from their goal. (b) Trajectories are generated using the Reciprocal Velocity Obstacles (RVO, orange). Note that, in this case, characters tend to move apart too much from each other. (c) Using our approach, characters dynamically switch motion policy depending on their local context, hence overcoming the motion artifacts displayed in (a) and (b). In this example, characters use the PL model, but switch to RVO in the 90º crossing context. The agents' color encodes their current policy. Our dynamic adaptation results in an increase of the overall quality score, S_{QF}.*

Abstract

Simulating crowds requires controlling a very large number of trajectories of characters and is usually performed using crowd steering algorithms. The question of choosing the right algorithm with the right parameter values is of crucial importance given the large impact on the quality of results. In this paper, we study the performance of a number of steering policies (i.e., simulation algorithm and its parameters) in a variety of contexts, resorting to an existing quality function able to automatically evaluate simulation results. This analysis allows us to map contexts to the performance of steering policies. Based on this mapping, we demonstrate that distributing the best performing policies among characters improves the resulting simulations. Furthermore, we also propose a solution to dynamically adjust the policies, for each agent independently and while the simulation is running, based on the local context each agent is currently in. We demonstrate significant improvements of simulation results compared to previous work that would optimize parameters once for the whole simulation, or pick an optimized, but unique and static, policy for a given global simulation context.

CCS Concepts

• ***Computing methodologies*** → ***Simulation evaluation;*** *Motion path planning; Agent / discrete models; Multi-agent systems;*

1. Introduction

Crowd simulators are useful to populate large environments with autonomous characters, e.g., to create lively and plausible scenes. It has long been established that the simulation algorithms and their parameters have a direct impact on the resulting quality of animations. Extensive research has shown that each algorithm performs well in a limited range of scenarios, e.g., some perform better in high density cases than others. It is also known that using appropriate parameter values for a specific scenario can improve performance. Nevertheless, as the number of published steering algorithms and techniques continues to increase, so does the difficulty of comparing their performances. This difficulty gave rise to a number of works exploring the questions of evaluating scenario coverage [KBS*16], setting simulation parameters [WJGO*14] and picking the best performing algorithm [KSHG18]. Previous work, though, generally considered scenarios in a global manner (e.g., an evacuation, a narrow corridor, or a complex pedestrian crossing) and usually propose a unique technique for setting up a simulation (e.g., an algorithm and parameter values to best simulate a scenario). However, the question of mixing multiple steering algorithms and of dynamically adjusting parameters during simulation has been quite unexplored.

The goal of this study is to determine the benefit of dynamically adjusting characters' steering policy (i.e., specific algorithm with specific parameter values) so that characters adapt their motion to a region of a scenario that we call *context*. We consider a simple abstract context definition, based on local density and the flow direction (in a uniform region of the scenario). An example of a changing abstract context could be being part of a dense flow of people crossing another flow. Initially, we study the quality performance of various steering algorithms on a diverse sample of the context space. To measure this, we evaluate the simulated trajectories with a quality metric [CDMH*21]. We demonstrate that quality is further increased when characters periodically estimate their per-agent local context and adapt their policy (i.e., their behaviour) to their surroundings.

Our contributions are the following ones:

- We propose the definition of a local context for crowd simulation agents based on density and main directions of local flows.
- We demonstrate that this context definition is enough to discriminate the performance of various steering policies.
- We compute and provide a mapping from context to best performing steering policy.
- We propose a mechanism to switch between different policies, and demonstrate the benefit of using multiple policies in a single simulation.

The remaining of our paper is organized as follows: Section 2 gives an overview of crowd steering algorithms, trajectory evaluation techniques, and related topics. Then, Section 3 lays out the theoretical basis for this work and introduces some concepts that will be used throughout the paper. Section 4 is concerned with the methodology used for this study. The findings of the research are presented in Section 5, where the quantitative and qualitative results are described and compared, and Section 6 discusses the results and highlights some of their implications. We also analyse the trade-off between quality improvement and computational overhead due to context adaptation. Finally, some recommendations for future work and suggestions can be read in Section 7.

2. State of the art

This paper proposes novel ways in which traditional crowd steering algorithms can be used to improve the resulting simulations. We discuss here a number of relevant previous works related to traditional steering algorithms, the evaluation of trajectories created by such algorithms, how this information can be used to refine crowd simulations, etc.

2.1. Steering algorithms

The crowd simulation research field is concerned with understanding, predicting and reproducing the motion of real human crowds. Crowd simulators are based on several classes of algorithms which are designed to generate realistic trajectories of numerous moving characters. Various approaches to this problem have been proposed. *Macroscopic* approaches consider crowds as a whole, modeling it as a single continuous moving matter [Hug03, TCP06]. *Microscopic* crowd simulation algorithms set the principles by which agents move individually and global crowd motion effects are expected to emerge from the interactions between agents. In Reynold's [Rey87] seminal work each boid followed the mean velocity field generated by neighbours. The number of categories of simulation algorithms rapidly grew with force-based models [HM95, KSG14], velocity-based models [PPD07, vMM08, KHBO09], vision-based models [OPOD10, DMCN*17], or data-driven models [LCL07, CC14]. These are few examples of a large body of literature.

Existing state-of-the-art steering algorithms are difficult to test and compare due to their very different strategies and implementations. To propose a standardisation for such algorithms, the authors of [vTGG*20] propose a holistic interpretation by transforming them into parametric cost functions in velocity space. The behaviours obtained with the algorithms in velocity space are very close to those obtained with the original algorithms. In this paper, the steering algorithms use the implementation in [vTGG*20].

There is a growing body of literature that recognises the different performance of steering algorithms in different scenarios [vTP21, YLG*20]. Numerous studies try to find the best parameters for existing steering algorithms, often comparing the results using data-based performance metrics [GVDBL*12]. The objective of these works is to aid the selection of policies in order to improve the trajectories resulting from simulation. Some authors have even proposed strategies to profit from two steering strategies. For instance, van Toll et al. [vTBSP20] combine agent-based (the Social Forces model) and particle-based approaches (Smoothed Hydrodynamic Particles) through abstraction layers in order to improve the behaviour in high density scenarios.

2.2. Quality metrics

Crowd simulations result in large sets of individual animation trajectories. Their quality depends on a number of rules by which agents move (simulation models), as well as parameter values to control the simulation. They are not intuitive nor easy to tune and often depend on the scenario to be simulated. Our objective is to propose a method to evaluate these simulation results, regardless of the method by which they are generated. We can distinguish various approaches to the evaluation of crowd simulations. A group of approaches uses paths of real crowds, and evaluate the ability of simulators to reproduce them. The question of comparison metrics is central, and several solutions have been proposed [GVDBL*12, WJGO*14, CKGC14]: these metrics consider crowd movement at different scales and take into account the variability of behaviors. However, there are drawbacks associated with the use of reference data, e.g., the ability of steering algorithms to replicate some patterns existing in real trajectory data and the limited amount of pedestrian trajectories available, which can lead to overfitted results.

Research on policy selection (per-scenario or per-character) has been mostly restricted to limited comparisons between synthetic trajectories and real data. A broader perspective has been adopted by some authors that, instead of focusing on agent trajectories, measure crowd motion characteristics such as the ratio between the density and the average speed in different cultures [JARLP12, CSC09, KWS*11]. In this work, we use the perceptually validated quality function proposed in [CDMH*21] to evaluate the simulation results. This metric abstracts from real data by studying the distribution of a number of motion features and then penalising character motions deviating from what is found to be expected in real human pedestrian trajectories. By using the proposed quality function we remove the need of directly relying in real data to evaluate synthetic trajectories and we can evaluate previously unseen interaction types, always in the scope of ambient crowds (groups of pedestrians that do not show any specific behaviour other than walking to their goal and avoidance maneuvers). The benefit of this approach is that no real trajectory data needs to be gathered in order to evaluate synthetic trajectories and the simulated trajectories do not need to resemble the ones found in the original data-sets. Moreover, this metric takes into account different characteristics of character motion that, once combined, lead to a demonstrated correlation between the metric and non-expert perception of trajectory quality, unlike, to the best of our knowledge, other existing metrics.

2.3. Policy assignment

Closest to our approach is the work of Kapadia et al. [KSHG18], which compares the performance of steering algorithms (using default parameters) in terms of distance-to-real-data in different scenarios. They compute how closely each steering algorithm is able to replicate the real trajectory of that character. This gives an insight of which algorithm (out of 6) works best in a type of scenario, e.g., a medium density area when entering a bottleneck corridor. The main difference with our approach is that they pick the best steering algorithm for a previously unseen scenario by studying the initial positions of characters and predicting what type of scenario it is – out of the data sets they use – and then selecting the steering algorithm with higher likely accuracy. Instead, we identify the context of each character at each time step and pick the optimal steering algorithm for the context, obtaining better quality simulations. To evaluate trajectories, the authors of [KSHG18] propose a simulation accuracy metric, based on the Entropy metric by [GVDBL*12], that measures the ability of a steering algorithm to create a trajectory similar to that found in real data. Instead, we rely on the metric proposed by Cabrero-Daniel et al. [CDMH*21] that abstracts from real data. Moreover, the characterisation of characters' contexts, a key point in this work, is different from that of [KSHG18]. Its authors derive a compact and continuous representation of pedestrian interactions directly from data based on per-agent *minimal predicted distances* (MPD). Instead, we model the context as the description of the dynamics in a neighbourhood of the agent, at each time, and we expect to cover a wide variety of different local interactions.

3. Overview

The objective of this work is to propose and compare a number of crowd simulation strategies, including a dynamic adaptation of characters' policy to their *local context*. We propose a policy adaptation technique to improve the overall quality of *crowd trajectories* simulated with traditional steering algorithms. Through the rest of this paper, we discuss and prove how the quality of simulations can be improved by increasing the adaptability of the characters to their surrounding environment.

We demonstrate the usefulness of our approach in the scope of *ambient crowds*, which are defined as groups of pedestrians that do not show any specific behaviour such as, e.g., queuing. The characters which compose the crowd are both homogeneous in the sense that the crowd is composed of similar-sized adults, and heterogeneous because each agent has its own attributes and objectives. Other types of contexts, for specific scenarios or applications, could be defined and evaluated in a similar way, and is further discussed in Section 6.

In this work, we analyse the performance of different navigation strategies using an existing quality metric, *QF* by [CDMH*21], that evaluates the quality of trajectories (orange boxes in Figure 2). Then, we propose an "abstract" context recogniser which allows agents to adapt their navigation strategy depending on their local context (blue loop in Figure 2).

With all this information, we evaluate the relative performance of four crowd simulation strategies: (i) all characters in the crowd sharing the steering policy (baseline); (ii) optimising the steering policy for each character (to maximize the simulation quality); (iii) dynamically adjusting the policy of each character to its current context; and (iv) assigning policies, also according to context, following a probability distribution.

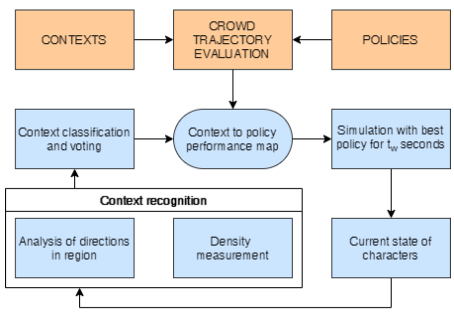


Figure 2: *Overview figure for context to policy mapping (orange), policy distribution among characters, and the context-adaptation loop (blue).*

4. Learning context-specific policies

The following part of this paper describes in detail the creation a context-to-model map through the evaluation of the resulting trajectories. The analysis is made by studying the quality performance of steering algorithms in a variety of contexts (orange boxes in Figure 2). Section 4.1 presents the concept of context and details which types of contexts will be used in the paper. A number of steering algorithms is evaluated in each context of Figure 3. The experimental setup for this analysis is then presented in Section 4.2, together with the results.

4.1. Contexts

Evidence from several studies suggests that the performance of different steering algorithms varies depending on the type of simulation they are used for. Crowds can be simulated in a variety of 2D *scenarios*, the latter being defined as a set of N agents together with their initial positions and internal properties (e.g., comfort speed, maximum acceleration, goal direction, radius, etc.). These scenarios can be arbitrarily complex and include a wide range of character interactions such as crossings, bottlenecks, etc. Within a given scenario, different zones can exhibit different interaction features at different points in time. Therefore, we could potentially identify several different *contexts* within the same scenario at a given simulation step. We define contexts as uniform spatial regions to which we associate a characterization of its trajectories based on the local density and the directions of the main flows of neighbouring agents.

To evaluate the performance of different algorithms in different contexts, we discretize the continuous context space into a representative subset of common pedestrian interactions. To this end, we consider three classes of contexts: (i) crossing of two unidirectional flows (F) of agents; (ii) crossing of two bidirectional flows (BF) of agents, where each bidirectional flow contains agents going along the flow in opposite directions; (iii) and unstructured contexts (FN). The crossing contexts are characterized by a bearing angle, which measures the angle at which the two flows cross each other. In order to cover a variety of interaction types, we have defined a total of 6 bearing angles for unidirectional flows crossing (ranging from 0 to 170°), and 4 bearing angles for bidirectional flows crossing (ranging from 0 to 90°). Moreover, we consider three levels of density of agents (low, medium and high: 0.5, 1 and 2 p/m^2, respectively) for each context, making a total of 33 representative contexts (11 per density level). The contexts for each level are shown in Figure 3.

4.2. Context to policy performance map

This section discusses the performance of the navigation policies (i.e., a steering algorithm and its parameters) in each of the 33 different contexts considered in this work.

We decided to evaluate the performance of the following set of representative crowd simulation algorithms for all the contexts defined in the previous section. Each of these algorithms (in ascending order of computational complexity) is implemented in velocity space [vTGG*20], and will be tuned for each context based on the procedure described in the next section:

- Universal Power Law (PL) [KSG14]
- Optimal Reciprocal Collision Avoidance (ORCA) [vdBGLM11]
- TtcaDca (TTCA), based on the vision-based algorithm by [DMCN*17]
- Social Forces (SF) [HM95]
- Moussaid (Mou) [MHT11]
- PLEdestrians (PLE) [GCC*10]
- Reciprocal Velocity Obstacles (RVO) [vMM08]
- Karamouzas (Kar) [KO11]
- Paris (Par) [PPD07]

The best parameter setting for each of the considered steering algorithms and for each context is found by maximizing the quality function QF through the iterative process described in [CDMH*21]. During this optimization process, the algorithm parameters are not constrained, meaning that we compare algorithms at the "best of their abilities". The performance of each algorithm in each is stored and represented as "score maps" which are summarised in Table 1. This map only needs to be computed once and can easily be updated to introduce new steering algorithms. The appropriate parameter values for each policy and context are presented in the Supplementary Material.

In order to learn the parameters for each algorithm, we simulated crowds in toric worlds: finite planes where the movement is "wrapped around" i.e. if a character leaves the plane on one side, it appears on the other (and interactions in boundary areas are controlled). We use toric worlds in order to simulate continuous flows and uniform density crowds. In this work, we use 10x10 meter toric world simulations, where the number of characters depends on the density of the respective context (first column in Table 1). To find the best parameters for a specific steering algorithm and a specific context, a genetic algorithm is used. In the learning process, each simulation run is initialized with some random variations in the initial character positions, to provide some slight variations of the simulated context. Moreover, the initial seconds of each simulation are discarded in our measurements, as they might contain small artifacts which are not representative of the actual context, e.g., the initial position of characters might lead to strange avoidance maneuvers at the start of the simulation.

To measure the performance of a policy in a context, we study the QF score of 300 seconds of trajectories simulated in each context. Performances for each context and steering algorithm are summarized in Table 1, which is one of the core contributions of this work. In the event that two algorithms have similar average quality for a given context, the algorithm with less computational complexity is preferred. An example of this is a unidirectional flow with low density where the Universal Power Law (PL) is preferred. Similarly, ORCA is often chosen over RVO for being more time efficient. Note that ORCA and RVO outperform other algorithms in high density scenarios. On the other hand vision based models, like [MHT11], tend to work better in more complex scenarios, like unstructured crossings. As shown in the following section, this information can be used in a crowd simulation, to adapt the characters' policy mid-simulation and hence increase the quality of the final result.

5. Application to Policy Selection

Our goal in this section is to demonstrate that it is possible to improve the quality of crowd simulations in any scenario, i.e., beyond one unique context. In Section 5.1, we propose a direct application of the context-to-policy map for this. This technique is based on

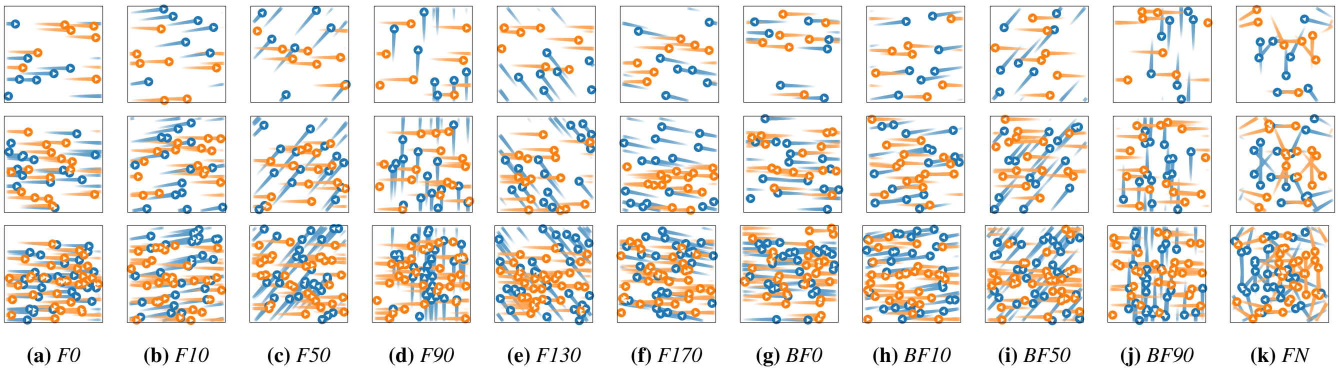


Figure 3: *Representative contexts defined in this work, where a context is defined by the local density and the directions of the main flows of neighbouring agents. Density increases with rows. F stands for Uniform Flow, BF for Bidirectional Flow, and FN for unstructured scenarios (N directions).*

automatically detecting the local agent context and assigning the agent the respective optimal policy. Then, we study whether we can further improve the results of the optimal policy per context found in Table 1. To this end, instead of using a single policy for all agents in a context, we assign the agents a mix of the two best policies found for each context (Section 5.2). The relative evaluation of these strategies, over a set of benchmark scenarios, is discussed in Section 5.3.

5.1. Dynamic context adaptation

When simulating a crowd in a given scenario, it is likely that several different types of interactions between agents will emerge in different sub-regions of the scenario and at different points in time. Therefore, from the perspective of an agent, the context (i.e., the other agents' motion features in a sub-region around the agent) is likely to be dynamic and to change several times during the simulation. We will refer to the dynamic context around a specific character as *local context*. To illustrate this from the perspective of an agent, let us focus, for example, on Figure 1. In this scenario we have two flows crossing at a 90-degree angle and the characters' "local context" changes from a single flow "context" to a crossing flows at 90 degrees "context" and back to the single flow "context". This observation, coupled with the results presented in Table 1 which show that the optimal policy is context-dependent, motivates our goal of detecting the agents' context during the simulation and, subsequently, to use this information to dynamically adapt the agents' steering policy. Our approach to this problem is described in the following sections.

5.1.1. Context detection

Our approach to the problem of run-time detection of the local context around each agent is: (i) first, we detect the type of motion on a small circular region around the agent of interest; (ii) then, we use this information to map the local context to one of the 33 studied contexts presented in Section 4. The relevant features required to perform this mapping operation are the local density and the local distribution of walking directions. The local density is computed by defining a radius r around the current agent, and determining the ratio between the number of agents inside that area and the area of the circle. We have experimentally found that using $r = 4$m leads to good results. The local distribution of walking directions is determined by considering the walking direction of agents inside the circular area, filtered over a window of 1 second. Then we extract the main flows resulting from this set of directions by analytically studying the distribution of directions and extracting the main modes. Once we obtain the main directions of the flows, we classify the context depending on the number of flows present (e.g., 1 for F0, 2 for other unidirectional flow crossings, and 4 for bidirectional flow crossings). Then, in the case of unidirectional and bidirectional flow crossing contexts, we compute the bearing angle between the two flows. If only two main directions are found, the difference between the bearing angles is used to classify the local context into one of the flow crossing contexts. Otherwise, the context is considered to be an unstructured scenario (NF). The density and angle between character flows is used to classify the local context into one of the previously defined context bins. One more step is performed to select the policy to use in the current step, π_s, using a simple voting system: after a character classifies its per-agent local context into one of the studied contexts, it finds neighbours within a radius. The more common context among those characters is chosen as the character's local context and is then used to select the best policy for the local context of each character.

5.1.2. Smooth policy transition

Changing the policy of a character c, in the middle of the simulation depending on its local context, lc_c, is prone to cause artifacts. This is because two steering algorithms, for very similar situations in consecutive time steps, might compute very different next velocities, $\mathbf{v}'_1$ and $\mathbf{v}'_2$. This could lead to sharp changes in the direction of characters when they enter a new context (related to flickering in direction). In order to ease the transition between algorithms we propose a transition strategy based on overlapping segments and algorithm combination in cost space.

The framework presented in [vTGG*20] uses combinable cost functions (in velocity space) to reproduce a number of steering algorithms, including those listed in Section 4.2. This way, characters

		Steering Algorithm								
Density	**Flow**	PL	ORCA	TTCA	SF	Mou	PLE	RVO	Kar	Par
0.5	F0	**94**	94	89	93	94	86	93	91	86
0.5	F10	92	92	89	91	92	85	92	**93**	86
0.5	F50	90	**91**	85	87	91	88	91	88	85
0.5	F90	91	90	89	90	91	87	**92**	89	85
0.5	F130	90	90	88	90	**91**	87	91	88	84
0.5	F170	**90**	90	86	88	91	87	90	87	82
0.5	BF0	91	92	88	88	91	88	**93**	88	85
0.5	BF10	87	89	83	84	**90**	85	88	85	81
0.5	BF50	84	86	79	78	87	**81**	85	82	76
0.5	BF90	88	87	82	**90**	88	85	88	83	82
0.5	FN	85	85	77	80	**86**	83	85	81	78
1.0	F0	90	92	80	79	91	87	**93**	86	81
1.0	F10	90	**93**	83	87	92	88	91	88	85
1.0	F50	88	89	75	82	89	86	88	**90**	84
1.0	F90	87	90	72	77	89	84	89	**91**	82
1.0	F130	87	88	76	81	88	85	88	**89**	83
1.0	F170	85	**89**	73	78	88	85	86	81	80
1.0	BF0	86	**91**	69	73	88	86	89	82	80
1.0	BF10	83	**85**	74	80	84	82	83	79	80
1.0	BF50	76	77	64	71	**78**	76	77	72	73
1.0	BF90	77	78	59	66	**79**	76	77	73	71
1.0	FN	79	**82**	72	71	81	78	80	76	76
2.0	F0	87	**93**	76	78	92	88	91	84	84
2.0	F10	82	91	71	70	90	84	**92**	81	76
2.0	F50	82	84	74	75	84	80	**85**	79	78
2.0	F90	77	**81**	68	66	80	76	79	74	72
2.0	F130	73	**79**	63	53	77	73	77	71	68
2.0	F170	76	**78**	67	70	77	74	76	72	71
2.0	BF0	76	**81**	66	64	80	76	79	73	71
2.0	BF10	69	73	61	49	73	67	**74**	68	64
2.0	BF50	62	70	64	**72**	71	68	70	64	54
2.0	BF90	61	64	71	**72**	61	66	69	58	53
2.0	FN	54	55	**75**	50	53	52	75	44	58

Table 1: *Context to the performance of each steering algorithm with appropriate parameters. The best performing algorithm is used to map each context to a steering policy. The score in each cell is computed using the quality function QF proposed in [CDMH*21].*

transitioning from one context to another can progressively use less an algorithm and more another algorithm and compute the next velocity, $\mathbf{v}'$. The weights for this transition are given by:

$$\begin{aligned} \omega(t) &= (1+e^{-kt})^{-1} \\ \mathbf{v}' &\leftarrow \omega(t)\pi_1 + (1-\omega(t))\pi_2 \end{aligned} \tag{1}$$

where t stands for the percentage of completion of the transition (mapped from -0.5 to 0.5) and k is the steepness of the transition. In our work, we experimentally found that $k = 9$ results in smooth transitions between algorithms, as can be seen in the Supplementary Videos. The selected $\mathbf{v}'$ for a character corresponds to an admissible velocity that minimises the combined costs of the two algorithms in velocity space.

Nevertheless, some steering algorithms might not be directly compatible for they can return opposite next velocities for the same character state, e.g., avoiding maneuvers turning right or left. A typical example of this is combining a velocity-based model like RVO with a force-based model like Social Forces. Figure 4 illustrates this problem, where colored areas represent a simplification of the regions in velocity space where values of the costs functions are minimal. In this example, combining the two policies directly, would not necessarily make sense: the selected $\mathbf{v}'$, optimal for both algorithms (the $\mathbf{v}'$ with lowest overall cost), could mean "not turning" (which could lead to a collision) or even reducing the walking speed to a stop.

Instead of directly combining the steering algorithms, we study the predicted next velocity for both algorithms separately and look for inconsistencies in the outputs of the two algorithms (i.e., if the angle between v_1 and v_2 is greater than a threshold, th). If the two steering algorithms return opposing solutions, the next velocity is selected among the two by studying whether they are consistent with the previous motion and depending on the value of t. If no inconsistencies are present, the velocities computed by the two algorithms are combined using Eq. (1).

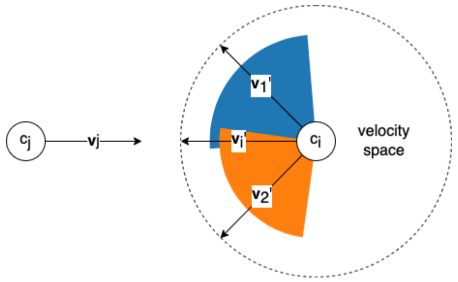


Figure 4: *Representation of the issues with selecting the next velocity* $\mathbf{v}_i'$ *for character* c_i *when using motion combination in velocity space. The best region in velocity space for the first algorithm (blue) slightly overlaps the best region in velocity space for the second algorithm (orange). Nevertheless, the velocities lying in the overlapping area (best for both steering algorithms at once) lead to a collision with* c_j.

Algorithm 1: Policy adaptation algorithm for a given agent.

Input : $\mathcal{P}$: context to policy map
$\hat{\pi}$: policy in previous time window
p: agent position
r: radius of circular area defining local context
t_w: number of simulation steps to be performed
t_s: number of simulation steps for policy transition

1 $\mathcal{A} \leftarrow$ getSubRegion(p, r)
2 **for** *each agent* $n \in \mathcal{A}$ **do**
3     $\mathcal{A}_n \leftarrow$ getSubRegion(n, r)
4     $c_n \leftarrow$ detectLocalConext$(\mathcal{A}_n)$
5 **end**
6 $\mathcal{C} \leftarrow$ voteContext$(\{c_1, \dots, c_N\})$
7 $\pi \leftarrow \mathcal{P}(\mathcal{C})$
8 $t_t \leftarrow 1$
9 **if** $\pi \neq \hat{\pi}$ **then**
10   **for** t *in* $[1, t_s]$ **do**
11         $\mathbf{v}' \leftarrow$ smoothVelocity$(\hat{\pi}, \pi, t)$
12   simulate$(t, \mathbf{v}')$
13   **end**
14     $t_t \leftarrow t_t + t_s$
15 **end**
16 **for** t *in* $[t_t, t_w]$ **do**
17     $\mathbf{v}' \leftarrow$ getVelocity(π, t)
18   simulate$(t, \mathbf{v}')$
19 **end**

5.1.3. Simulation loop

Our simulation loop with dynamic context-based policy adaptation is shown in Algorithm 1. The time window (measured in simulation steps) in which the same policy is applied is given by t_w. The algorithm starts by delimiting the circular area $\mathcal{A}$ with radius r centered on the position p of the current agent (line 1). Then, in lines 2 to 5, the local context of the N agents within the area $\mathcal{A}$ is detected, based on the motion features of each area $\mathcal{A}_n$ centered on each of the agents n. Note that this includes the current agent for which we want to adapt the policy. A voting step follows, in which the final local context $\mathcal{C}$ of the current agent is chosen based on the set of contexts $\{c_1, \dots, c_N\}$ detected for all agents within the area $\mathcal{A}$ (line 6). This context is then used to determine the new motion policy π (line 7), using the context to policy map $\mathcal{P}$ described in Section 4. In lines 9 to 15, the transition between the previous policy $\hat{\pi}$ and the new policy π is smoothed out during t_s steps as explained in Section 5.1.2, in case strong motion discrepancies are detected, hence avoiding undesired discontinuities in the simulation. Finally, the remaining simulation steps within the time window t_w are performed using the context-adapted policy π that is used to compute the next velocity of the given agent at time t (lines 16 to 19). This process is represented with blue boxes in Figure 2.

5.1.4. Results

To present the results of our approach, we first illustrate them with the specific case of a two-flow crossing (from left to right and from bottom to top), simulated either with PL, RVO, or our approach (see Figure 1). In Figure 1a, we can see that the PL method struggles in regions where the two flows cross, making the characters move diagonally. This is penalised by *QF* because characters deviate from their desired direction for too long, even excessively moving away from their goal. On the other hand, in Figure 1b using RVO we can see that even if the flows are able to cross each other, characters tend to move apart too much from each other (larger spread of characters across the two flows). Finally, Figure 1c shows an example of our policy switching based on the "score map" presented in Table 1, where PL is used in low density, unidirectional contexts, and RVO is used in low density, 90 degrees crossings. Overall, Figure 1c shows that characters do not deviate too much from their goal direction and switch back to PL as soon as they exit the crossing area, as the majority of characters around them now lead to a change in the distribution of directions (leading to a change of context). This also enables characters to switch back to the less computationally complex PL method, that works well for unidirectional flows.

We can interpret policy adaptation in two ways: (i) changing the steering policy to be able to deal with complex interactions or (ii) "relaxing" the algorithm when the scenario does not require a more time consuming algorithm to correctly solve the interactions. In an extreme case, when distances between neighbours are acceptable (an interaction range of 3.5 meters is commonly used in the literature) and all agents have the same comfort speed one could use a goal reaching force (without avoidance maneuvers) because there would be no predicted collisions nor unreasonable values for other features.

5.2. Mapping context to a distribution of policies

If the steering algorithm and its parameter values are not shared among all characters, the crowd is heterogeneous and characters exhibit different behaviours, typically leading to better simulation results [WJGO*14, GVDBL*12]. The following sections are concerned with producing heterogeneous crowds within each context using the information contained in Table 1. In particular, Table 1 can be used to map contexts to a distribution of policies, instead of mapping a context to a single (optimal) policy as described in the previous sections.

We propose and evaluate two strategies to assign different policies to agents in a context. In the first strategy, presented in Section 5.2.1, we randomly assign each agent one of the two best policies learnt for each context. In the second strategy, presented in Section 5.2.2, we replace the random assignment of policy to each agent by an optimization process which determines to which particular agents should each of the two policies be assigned, so as to maximize the QF score.

5.2.1. Random selection of agent policies

To determine to which extent it is beneficial to combine different policies within the same context we have measured the simulation quality per context using a combination of two policies: the optimal policy p^* for the context, and the second best policy p'^* found for the same context according to Table 1. Each agent randomly picks one of these two policies following a probability distribution aimed at keeping the ratio between p^* and p'^* at a desired level.

Figure 5 (blue) shows the average quality across all contexts of crowd trajectories where characters randomly pick among p^* and p'^*. The results have been generated considering different desired ratios between p^* and p'^*, ranging from all agents choosing policy p^* to all agents choosing policy p'^*. The blue bars show that the average QF score is higher when characters share the same policy, compared to the case where characters with different policies coexist in the same context. Contrary to expectations, no significant increase in the quality score, S_{QF}, was found compared with using a single policy optimised for a specific context.

5.2.2. Optimized selection of agent policies

Further statistical tests revealed that the average crowd trajectory quality across contexts could be improved by distributing the policies among characters in an informed way. The goal of this learning process is to maximise the quality of the resulting trajectories while maintaining the proportion of characters using each of the two best performing steering algorithms. The relation between the proportion of characters using each of the two best performing steering algorithms and the resulting average quality is represented in Figure 5 (orange colored bars). We can therefore conclude that, in contrast to a random assignment of steering algorithms, an informed distribution of the two bests algorithms for each context leads to an improvement in quality. The proportion of characters using a "complimentary" steering algorithm for a particular scenario seems to affect the resulting trajectories' quality. An explanation for this improvement could be that the crowd simulator avoids some artifacts by changing the steering policy of the affected characters.

We can therefore conclude that the quality of trajectories simulated in a specific context can be further improved when different characters use different policies, even if only a small percentage of characters use a different steering algorithm. Nevertheless, the small increase in S_{QF} is likely to be related to the optimisation of the policies per-context which is done in homogeneous contexts where all agents shared the same policy. There is a risk that the

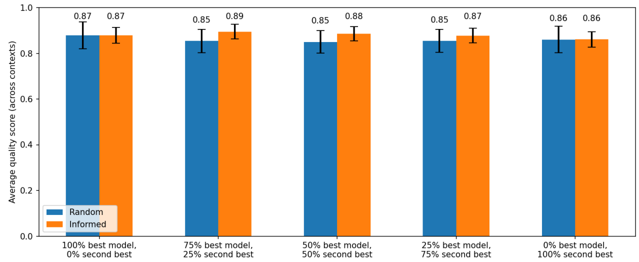


Figure 5: *Average quality (mapped from 0 to 1, across all contexts) depending on the proportion of characters using the best performing steering algorithm and the second best performing algorithm.*

used policies are not well adapted to contexts where characters use different steering strategies, such as in the experiments conducted in this section. As discussed in Section 6, the quality might be further improved if instead of using the policies tuned in Section 4.2, a mixture-of-policies for each context was learnt instead.

5.3. Strategy comparison

To quantitatively assess the effectiveness of the different motion strategies proposed in our paper we have evaluated their performance in a variety of scenarios. This quantitative evaluation of relies on QF, and on the following benchmark scenarios:

- Two groups of characters moving in opposite directions and crossing in the center. When the groups overlap, the density increases; after crossing, the density returns to the original value.
- Four groups of characters move towards the opposite side of the world, passing through the center. In the crossing, the local context of characters is a bidirectional flows crossing.
- Circle crossing: characters are disposed in a circle around the center of the world; their goal is to reach the opposite side.
- Two unidirectional flows move in the same direction and overlap. Characters in one of the flows have a higher desired speed.
- Random scenarios where every character in the crowd has its own initial position, desired direction and comfort speed.

For each scenario, several evaluations with different initial agent positions are made. The tested motion strategies are:

- S_0, where all characters use the same policy during all the simulation. This strategy should be seen as a baseline strategy;
- S_1, where each character has its own policy which is kept constant throughout the simulation; the algorithm and parameter values are optimised per character with QF as individual fitness measurement.
- S_2, which corresponds to the case where agents can dynamically switch policies during the simulation based on the context to policy map, as described in Section 5.1 and Algorithm 1;
- S_3, which corresponds to the case where agents can dynamically switch policies during the simulation but, in contrast with S_2, each context is mapped to a distribution of policies (instead of being mapped to a single policy), as described in Section 5.2.

The results for these four strategies are presented in Table 2. They show that, compared to the baseline method (S_0), the simulation

Strategy	Mean	Variance
Policy sharing (S_0)	61	9
Policy per-character (S_1)	78 (+28%)	6
Switching algorithms (S_2)	87 (+43%)	6
Switching distr. (S_3)	89 (+46%)	13

Table 2: *Average quality of the trajectories resulting from using each of the studied strategies in multiple scenarios. The percentages in-between brackets for S_1, S_2 and S_3 show the improvement brought by these strategies with respect to the base-line strategy S_0.*

quality can be improved by an average of 28% by simply tuning the policy so as to fit the current scenario, even if all agents use the same policy (S_1). This is in-line with the findings reported by previous works such as [WJGO*14] or [KSHG18]. Moreover, Table 2 also confirms the significant benefit brought by dynamically adapting the agents' policy based on their local context (S_3 and S_4). Such strategy, which advances the state-of-the-art by exploring an alternative way to describe the agents' local context and using *QF* to perform calibration, brings improvements of up to 46% in terms of average simulation quality with respect to the base-line method.

6. Limitations and future work

Context. The motion of a crowd can always be decomposed into a number of flows. Following this idea, we introduce a definition of *context* which is based on few main properties of these flows: their density and relative angle. Through the study of a discrete set of contexts, we demonstrate that those properties indeed discriminate various algorithms in their capacity of handling them correctly. We however left for future work a number of other context features that are likely to also influence simulation quality, such as flow rate or non-uniform distributions of flows. We could also have considered other features like the presence of groups or a higher heterogeneity in agents behaviors, etc. A new set of contexts, specifically designed for group behaviour, or to provide a more diverse dataset could also be added. This would allow creating additional benchmarks, targeting different behaviors. Notwithstanding their relatively simple characterisation, this work offers a proof of concept for a dynamic adaptation of policies based on local context.

Trajectory Evaluation. Not using data (as is the case of our proposed approach) has many advantages such as, for example, not having to collect and process real trajectory data, or not risking over-fitting to a particular data set. However the chosen trajectory evaluation metric (*QF* [CDMH*21]) limits the scope of evaluation to ambient crowds. The approach that we propose might thus be less suited for specific contexts or behaviours not considered in [CDMH*21]. The current *QF*, aimed at ambient crowds (no queuing, no running, no grouping, etc.), could be replaced by a different loss that specifically targets, for instance, group behaviour. Thanks to the modular architecture of the system, any of its components could be easily replaced by another deemed more appropriate, e.g., other evaluation techniques (already explored in previous works or not) could be considered.

Algorithms. In this paper, we study a variety of crowd steering algorithms. Some approaches, such as data-driven or reinforcement learning, are not covered in this framework and were not considered. This type of approaches, that implicitly generate human-like trajectories, do not follow similar parameter tuning and evaluation procedure. Future work is required to adjust our method to these categories of simulation techniques, however, our results offer an insight for improving their design. For example, it may be useful to decompose policies, training or data-sets according to contexts.

Parameter values. Simulation results are influenced by the adaptation time window (i.e., the frequency of the adaptations) and simulation time step (i.e., can affect the performance of some steering algorithms). The examination of these features is interesting, but it is beyond the scope of this paper. To be able to combine steering algorithms and validate the approach, these values are fixed in our experiments, which can limit the flexibility of specific algorithms; nevertheless, adapting the system to a multi-time-step setting could be an interesting direction for future work.

Policy combination. In this work, existing steering methods are dynamically distributed across characters in a context. This application is, to the best of our knowledge, the first work where improvements are reported using a context-based dynamic combination of crowd steering policies. The results in Figure 5 and Table 2 show little *QF* improvement from distributing the steering algorithms across characters, though. Further studies will be needed to to improve the results in policy-to-character assignment.

Simulation speed. Both policy switching and mixing have computational overheads (process illustrated in Figure 2), which slows down simulation. Nevertheless, this technique takes advantage of a "relaxation" of the navigation strategy when the constraints allow for it, e.g., using a simpler algorithm when the density is lower thus reducing computational time while maintaining the overall simulation quality. Further studies, which focus on context recognition and assignation in a computationally light manner, will need to be undertaken.

Policy switching. One could think about many techniques to switch, or even mix, navigation policies. We here prove that a simple method to switch policies is already effective, but many more could be explored, e.g. using multiple tuning policies at all times and voting the next velocity, v', for each character. Moreover, the transition between policies are synchronized for they happen every t_w seconds. A strategy where characters would recognise the context for every time step in the simulation would avoid potential artifacts due to characters switching their policy simultaneously.

Policy optimisation. We optimized policies taking into account contexts and we showed that mixing policies may further improve results, though slightly. The natural following step would be to jointly optimize a mixture of policies for each abstract context. It is also possible that a new policy, found without real data by directly maximising *QF*, could be derived so that it over-performs existing methods. Moreover, a different evaluation function could be used to take into account how real humans adjust their navigation to that of their neighbours, e.g., being more careful among inattentive people.

7. Conclusion

In this paper, we have proposed a framework to dynamically adapt the motion policy of characters when simulating large virtual crowds. Our approach is based on a context to policy map which shows for the agents' local context to a set of optimized policies, that are learnt once and for all in a previous step without requiring any real motion data. To this end, we have proposed a discretization of the full context space into a subset of 33 representative contexts and learned the optimal performing policies for each of the contexts. During the simulation, the agents' context is automatically detected and mapped to an optimized policy, which results in a crowd where characters dynamically adapt their motion strategy depending on their situation. Our results demonstrate the benefits of our approach for the crowd simulation quality, exhibiting a significant crowd quality improvement both visually and in terms of a quantitative perceptually-based quality function. Furthermore, the data-independence of our approach opens the path to easily build on and extend our framework to other contexts and policies, which can potentially trigger future research.

Acknowledgement

As Serra Húnter Fellow, Ricardo Marques acknowledges the support of the Serra Húnter Programme to this work. This work was partially funded by TIN2018-095232-B-C21, and SGR-2017 1742.

References

[CC14] Charalambous P., Chrysanthou Y.: The PAG crowd: A graph based approach for efficient data-driven crowd simulation. *Computer Graphics Forum 33*, 8 (2014), 95–108. 2

[CDMH*21] Cabrero Daniel B., Marques R., Hoyet L., Pettré J., Blat J.: *A Perceptually-Validated Metric for Crowd Trajectory Quality Evaluation*, vol. 4. Association for Computing Machinery, New York, NY, USA, sep 2021. URL: https://doi.org/10.1145/3480136, doi:10.1145/3480136. 2, 3, 4, 6, 9

[CKGC14] Charalambous P., Karamouzas I., Guy S., Chrysanthou Y.: A data-driven framework for visual crowd analysis. *Computer Graphics Forum 33* (10 2014). doi:10.1111/cgf.12472. 2

[CSC09] Chattaraj U., Seyfried A., Chakroborty P.: Comparison of pedestrian fundamental diagram across cultures. *Advances in Complex Systems (ACS) 12* (06 2009), 393–405. doi:10.1142/S0219525909002209. 3

[DMCN*17] Dutra T., Marques R., Cavalcante-Neto J., Vidal C., Pettre J.: Gradient-based steering for vision-based crowd simulation algorithms. *Computer Graphics Forum 36* (05 2017). doi:10.1111/cgf.13130. 2, 4

[GCC*10] Guy S. J., Chhugani J., Curtis S., Dubey P., Lin M., Manocha D.: Pledestrians: A least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2010), SCA '10, Eurographics Association, p. 119–128. 4

[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 31*, 6 (2012), 1–11. 2, 3, 7

[HM95] Helbing D., Molnár P.: Social force model for pedestrian dynamics. *Physical Review E 51*, 5 (may 1995), 4282–4286. URL: link.aps.org/doi/10.1103/PhysRevE.51.4282, doi:10.1103/PhysRevE.51.4282. 2, 4

[Hug03] Hughes R. L.: The flow of human crowds. *Annual review of fluid mechanics 35*, 1 (2003), 169–182. 2

[JARLP12] Jelić A., Appert-Rolland C., Lemercier S., Pettre J.: Properties of pedestrians walking in line: Fundamental diagrams. *Physical review. E, Statistical, nonlinear, and soft matter physics 85* (03 2012), 036111. doi:10.1103/PhysRevE.85.036111. 3

[KBS*16] Kapadia M., Berseth G., Singh S., Reinman G., Faloutsos P.: Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Simulating Heterogeneous Crowds with Interactive Behaviors*. AK Peters/CRC Press, 2016, pp. 193–210. 2

[KHBO09] Karamouzas I., Heil P., Beek P., Overmars M. H.: A predictive collision avoidance model for pedestrian simulation. In *Proceedings of the 2nd International Workshop on Motion in Games* (Berlin, Heidelberg, 2009), MIG '09, Springer-Verlag, p. 41–52. URL: https://doi.org/10.1007/978-3-642-10347-6_4, doi:10.1007/978-3-642-10347-6_4. 2

[KO11] Karamouzas I., Overmars M.: Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (2011), 394–406. 4

[KSG14] Karamouzas I., Skinner B., Guy S. J.: Universal Power Law Governing Pedestrian Interactions. *Physical Review Letters 113*, 23 (dec 2014), 238701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.238701https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.113.238701, doi:10.1103/PhysRevLett.113.238701. 2, 4

[KSHG18] Karamouzas I., Sohre N., Hu R., Guy S. J.: Crowd space: a predictive crowd analysis technique. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–14. 2, 3, 9

[KWS*11] Kapadia M., Wang M., Singh S., Reinman G., Faloutsos P.: Scenario space: Characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, Association for Computing Machinery, p. 53–62. URL: https://doi.org/10.1145/2019406.2019414, doi:10.1145/2019406.2019414. 3

[LCL07] Lerner A., Chrysanthou Y., Lischinski D.: Crowds by example. *Computer Graphics Forum 26*, 3 (2007), 655–664. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01089.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01089.x, doi:10.1111/j.1467-8659.2007.01089.x. 2

[MHT11] Moussaïd M., Helbing D., Theraulaz G.: How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences 108*, 17 (2011), 6884–6888. 4

[OPOD10] Ondřej J., Pettré J., Olivier A.-H., Donikian S.: A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics 29*, 4 (2010), 123. 2

[PPD07] Paris S., Pettre J., Donikian S.: Pedestrian reactive navigation for crowd simulation: a predictive approach abstract. *Comput. Graph. Forum 26* (09 2007), 665–674. doi:10.1111/j.1467-8659.2007.01090.x. 2, 4

[Rey87] Reynolds C. W.: Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 25–34. URL: https://doi.org/10.1145/37402.37406, doi:10.1145/37402.37406. 2

[TCP06] Treuille A., Cooper S., Popović Z.: Continuum crowds. *ACM Trans. Graph. 25*, 3 (July 2006), 1160–1168. URL: https://doi.org/10.1145/1141911.1142008, doi:10.1145/1141911.1142008. 2

[vdBGLM11] van den Berg J., Guy S. J., Lin M., Manocha D.: Reciprocal n-body collision avoidance. In *Robotics Research* (Berlin, Heidelberg, 2011), Pradalier C., Siegwart R., Hirzinger G., (Eds.), Springer Berlin Heidelberg, pp. 3–19. 4

[vMM08] VAN DEN BERG J., MING LIN, MANOCHA D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation* (New York, NY, May 2008), IEEE, pp. 1928–1935. doi:10.1109/ROBOT.2008.4543489. 2, 4

[vTBSP20] VAN TOLL W., BRAGA C., SOLENTHALER B., PETTRÉ J.: Extreme-density crowd simulation: Combining agents with smoothed particle hydrodynamics. In *Motion, Interaction and Games* (New York, NY, USA, 2020), MIG '20, Association for Computing Machinery. URL: https://doi.org/10.1145/3424636.3426896, doi:10.1145/3424636.3426896. 2

[vTGG*20] VAN TOLL W., GRZESKOWIAK F., GANDÍA A. L., AMIRIAN J., BERTON F., BRUNEAU J., DANIEL B. C., JOVANE A., PETTRÉ J.: Generalized microscropic crowd simulation using costs in velocity space. In *Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2020), I3D '20, Association for Computing Machinery. URL: https://doi.org/10.1145/3384382.3384532, doi:10.1145/3384382.3384532. 2, 4, 5

[vTP21] VAN TOLL W., PETTRÉ J.: Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s. *Computer Graphics Forum 40*, 2 (2021). URL: https://hal.inria.fr/hal-03197198. 2

[WJGO*14] WOLINSKI D., J. GUY S., OLIVIER A.-H., LIN M., MANOCHA D., PETTRÉ J.: Parameter estimation and comparative evaluation of crowd simulations. *Comput. Graph. Forum 33*, 2 (May 2014), 303–312. URL: http://dx.doi.org/10.1111/cgf.12328, doi:10.1111/cgf.12328. 2, 7, 9

[YLG*20] YANG S., LI T., GONG X., PENG B., HU J.: A review on crowd simulation and modeling. *Graphical Models 111* (2020). 2