On Streams and Incentives: A Synthesis of Individual and Collective Crowd Motion

Arthur van Goethem

Norman Jaklin

Atlas Cook IV

Roland Geraerts

Technical Report UU-CS-2015-005
March 2015

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

On Streams and Incentives: A Synthesis of Individual and Collective Crowd Motion

Arthur van Goethem
TU Eindhoven
a.i.v.goethem@tue.nl

Norman Jaklin
Utrecht University
n.s.jaklin@uu.nl

Atlas Cook IV
University of Hawaii at Manoa
acook4@hawaii.edu

Roland Geraerts
Utrecht University
r.j.geraerts@uu.nl

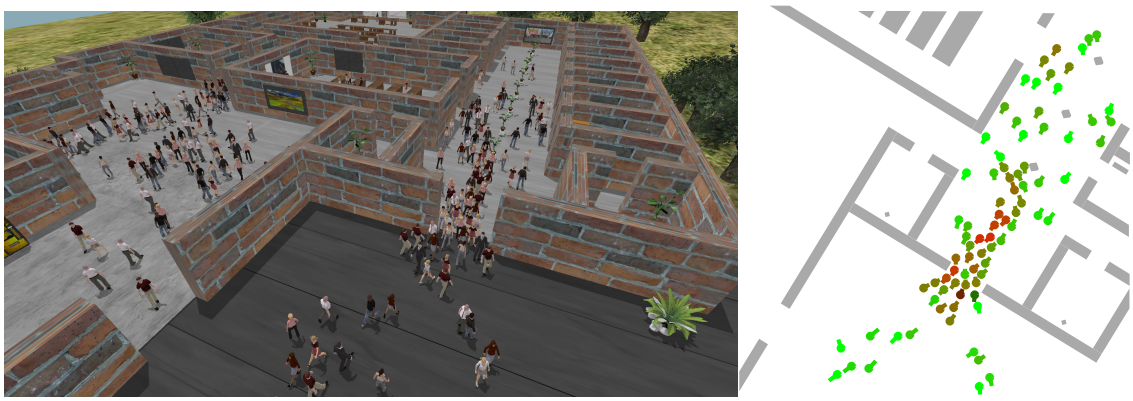

Figure 1: *Left*: A dense crowd of agents collaboratively moves through a narrow doorway. *Right*: A $2D$ representation of the doorway shows that each agent interpolates between individual behavior (green) and coordinated behavior (red).

Abstract

We present a novel crowd simulation model that combines the advantages of agent-based and flow-based paradigms while only relying on local information. Our model can handle arbitrary and dynamically changing crowd densities, and it enables agents to gradually interpolate between individual and coordinated behavior. This interpolation is based on a dynamically changing *incentive* value for each agent. The incentive value reflects an agent's willingness to coordinate with the crowd. A central new concept in our model is the notion of a *stream* of agents. Our model lets agents automatically form multiple streams with nearby agents as the density of the crowd increases. Our model can be used with any existing global path planning and local collision-avoidance method. Experiments show that our model yields energy-efficient and visually convincing crowd behavior for high-density scenarios while maintaining individual agent behavior at lower densities.

1 Introduction

Virtual crowds have become increasingly prominent in the movie industry and in video games. Realistic crowd behavior is also required for urban planning [1], as well as simulation software for safety training and evacuation scenarios [2, 3].

State-of-the-art techniques for crowd simulation can be divided into agent-based simulations and flow-based simulations. Agent-based simulations focus on the behaviors of each individual in the crowd. Among these are methods using velocities [4] and social force models [5]. While these methods usually work well at low to medium densities, they struggle when handling high crowd densities due to a lack of coordination between the agents. By contrast, flow-based simulations aim at simulating collective emergent phenomena by treating the crowd as one large entity. Among

these are regression-based models [6] and techniques based on fluid dynamics [7] or gas-kinetics [8]. These techniques typically perform well with high-density scenarios because they facilitate a high level of coordination among the agents. However, they struggle to handle low- to medium-density scenarios because they omit the individuality of the crowd members.

Contributions. In this work, we propose a new model that combines the advantages of agent-based and flow-based paradigms while only relying on local information. It enables the simulation of large numbers of virtual agents at arbitrary and dynamically changing crowd densities. Our technique preserves the individuality of each agent in any virtual $2D$ or multi-layered $3D$ environment. The model performs as well as existing agent-based models that focus on low- to medium-density scenarios, while also enabling the simulation of large crowds in highly dense situations without any additional requirements or user interference. Compared to existing agent-based models, our model significantly reduces the occurrence of deadlocks in extremely dense scenarios. Our model is flexible and supports existing methods for computing global paths, simulating an agent's individual behavior, and avoiding collisions with other agents. Furthermore, it yields energy-efficient and more realistic crowd movement that displays emergent crowd phenomena such as lane formation and the *edge effect* [9].

2 Related Work

For a general overview of crowd simulation topics, we refer the reader to the books by Thalmann and Musse [10] and Pelechano et al. [11]. In the remainder of this section, we focus on selected work that is related to our model.

One of the first flow-based models was proposed by Hughes [12]. Hughes represented pedestrians as a continuous density field, and crowd dynamics were described using partial differential functions. Treuille et al. [7] proposed a continuum-based crowd simulation model. They used a dynamic potential field to simulate large crowds in real-time. This model yields emergent phenomena such as lane formation. Single autonomous agents can be added to the crowd as dynamic obstacles. Lee et al. [6] present a regression-based model that is able to simulate particular crowd behavior that has been learned from recorded video data of real crowds. Other flow-based approaches come from the robotics community. Kerr and Spears [8] use a simulation model based on gas-kinetics for mobile robots. Pimenta et al. [13] propose a method for swarms of mobile robots that is based on Smoothed Particle Hydrodynamics. All of these flow-based models are able to solve high-density scenarios, but they are not well-suited for low- to medium-density scenarios where the individuality of single agents has a large impact on the overall behavior of the crowd. In addition, these flow-based methods usually have high computational costs when many different goal states are involved.

In addition to flow-based models, a wide range of agent-based crowd simulation models are available. Helbing et al. introduced a social-force model for pedestrian dynamics in [5] and subsequent work [2, 14, 15]. Torrens [16] has proposed a crowd simulation framework that aims at handling higher-level trip planning computations, medium-level computations such as vision and steering, and low-level computations for locomotion and physical collision detection. The HiDAC system by Pelechano et al. [17] combines psychological and geometrical rules with a social- and physical-forces model. Shao and Terzopoulos [18] show how to integrate motor, perceptual, behavioral and cognitive components within one model to simulate pedestrians in an urban environment. Unlike flow-based models, the above social-force models struggle when coordinating the movements of dense crowds. This can lead to non-desired phenomena such as deadlocks, oscillations, slow movements with unnecessary turns and detours, or a high number of collisions.

Lemercier et al. [19] have conducted an experimental study on herding and following behaviors. Models based on real-world pedestrian movements have been proposed by Antonini et al. [20] and Paris et al. [21]. Vizzari et al. [22] combine a group cohesion force with a goal force. Their environment is discrete and uses a floor-field to guide the pedestrians.

An approach similar to ours is the *PLEdestrians* algorithm by Guy et al. [9]. Based on the Principle of Least Effort [23], the authors propose a local greedy strategy that approximates the minimum

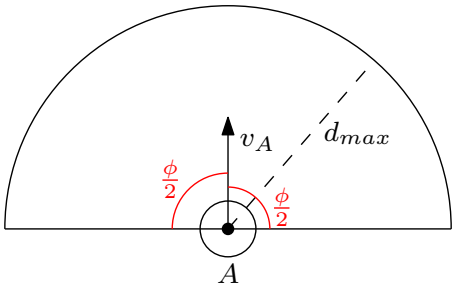

Figure 2: Agent A's *field of view* with a maximum viewing angle ϕ and a maximum look-ahead distance d_{max}, centered around A's velocity v_A.

of a biomechanical energy function in order to compute trajectories for individual agents. By comparison, our model uses a small set of local and easy to compute factors to guide agents along their paths. Both our technique and the technique of Guy et al. [9] exhibit desirable emergent behaviors.

Due to the gap between flow-based models and agent-based models, hybrid methods combining both paradigms have recently been considered. The method by Narain et al. [24] uses a dual representation of the crowd that is based on both individual agents and continuum dynamics. Like our model, agents have individual goals, but they can be forced to deviate from their preferred direction by the flow of the crowd. Contrary to their approach, our model omits continuum dynamics and simulates the tendency of humans to follow each other on a local and agent-based level. Our resulting herding behavior can therefore be related to Reynold's well-known model on flocks, herds and schools [25], while still allowing individual agent behaviors. Furthermore, our model measures local crowd density based on an agent's vision, and this overcomes problems that can occur with a grid-based approach. Kountouriotis et al. [26] proposed to combines flow-based models and agent-based models with a local approach that is similar to ours. In their work, the interpolation between individual and coordinated movement is based solely on crowd density. By computing a perceived local crowd flow that can differ per agent, we achieve a more extensive yet simple interpolation. Our model also allows the inclusion of different collision avoidance and global path-planning methods.

Other related works involve global path planning and collision avoidance among agents. These methods can be used as black boxes within our model. A global path planning method related to our work is the *Indicative Route Method* (IRM) by [4]. Given a global indicative route from an agent's start position to a goal position, the IRM computes an attraction point on the route in each step of the simulation and makes the agent approach this point using steering forces. Collision avoidance among agents is available via a range of velocity-based methods. One of the more popular ones is the ORCA method by [27], which is based on *reciprocal velocity obstacles*. A similar method has been introduced by [28]. It predicts future collisions for each agent and lets an agent take an action that guarantees collision-free movement. Park et al. [29] predict future collisions using a gaze movement angle. Vision-based approaches by [30, 31] use a *field of view* (FOV) for each agent to detect and prevent collisions. In this work, we combine our model with the IRM, and we test it with several velocity- and vision-based collision avoidance methods [27, 28, 30].

3 Preliminaries

We represent each agent as a disk with a variable radius. The center of the disk is the current position of the agent.

Each agent has a *field of view* (FOV), and an agent's steering behavior is based on a number of perceived neighboring agents. This is a fundamental difference from continuum-based and other flow-based methods [7] as these methods assume global knowledge of the environment. We assume that real people mainly execute and adapt their movement to visual input without global

knowledge of the crowd. We therefore believe that a local vision-based approach is well-suited for approximating realistic crowd behavior and simulating emergent phenomena observed in real crowds. Similar to Moussaïd et al. [30], we assume that an agent's FOV is a cone stretching out from the agent's current position, centered on the agent's current velocity vector and bounded by both a maximum look-ahead distance $d_{max} = 8$ meters and a maximum viewing angle $\phi = 180°$. The latter reflects the approximate viewing range people can perceive in real life [32].

4 Overview of our model

In each cycle of the simulation, we compute a force vector for each agent. This force vector is then applied to an agent's current velocity using a time-integration scheme such as Euler integration [33], which guarantees smooth paths [34].
Let A be an arbitrary agent. We perform the following five steps in each simulation cycle:

1. We compute an *individual velocity* for agent A. It represents the velocity A would choose if no other agents were in sight. Our model is independent of the exact method that is used.
2. We compute the *local crowd density* that agent A can perceive; see Section 5.1.
3. We compute the locally *perceived stream velocity* of agents near A; see Section 5.2.
4. We compute A's *incentive* λ. This incentive is used to interpolate between the *individual velocity* from step 1 and the *perceived stream velocity* from step 3; see Section 5.3.
5. The interpolated velocity is passed to a collision-avoidance algorithm. Our model is independent of the exact method that is used.

5 Streams

A central concept we introduce is the notion of local *streams* of agents. Intuitively, streams are flows of people that coordinate their movement by either aligning their paths or following each other. Streams can be observed in real-life situations as crowd density increases; see Figure 3. We base our model on the assumption that people tend to move by following a least-effort principle of energy-minimization [9, 35]. We postulate that actively forming and following streams at high densities is a more energy-efficient strategy compared to pursuing individual goals. This follows because the use of streams leads to fewer collisions and abrupt changes in the direction of movement.
Local crowd density is important for determining an agent's behavior in our model. Section 5.1 describes how we compute local density information using an agent's FOV. In subsequent sections, we discuss how to compute an agent's perceived stream velocity (Section 5.2) and incentive (Section 5.3).

5.1 Computing local density information

After agent A's individual velocity has been computed as an initial step, we calculate the crowd density $\rho \in [0, 1]$. We have tested three different density measurements in a set of preliminary experiments. In this section, we present the method that turned out to be best-suited for our model.
We use the agent's FOV to compute ρ. We determine the set $\mathcal{N}$ of neighboring agents that have their current position inside A's maximum FOV of 180°. By summing up the area $\Delta(N)$ occupied for each agent $N \in \mathcal{N}$ and dividing it by the total area $\Delta(FOV)$ of A's FOV, we determine how much of the FOV is being occupied. Fruin [36] was the first to formalize the impact crowd density has on the safety of pedestrians. Fruin introduced a six-stage *Level-of-Service* system, ranging from free movement without collisions to highly dense situations. According to this system, an FOV occupied to one third can already be considered a highly crowded situation. Thus, we multiply our result by 3 and cap it at a maximum of 1. This yields a maximum density value of 1 as soon as at least one third of A's FOV is occupied by other agents. Formally, we define the crowd density

Figure 3: Example of stream formation in real life situations. People between arrows of the same color belong to the same stream.

ρ as follows:

$$\rho := \min\left(\frac{3}{\Delta(FOV)}\sum_{N\in\mathcal{N}}\Delta(N),\ \ 1\right). \tag{1}$$

5.2 The perceived stream velocity

The next step is to compute the direction and speed of the stream as perceived by agent A. In situations where A is willing to coordinate with the crowd, our model lets A approach the perceived stream whenever the distance from A to the stream members is still large. If, by contrast, A is close to the stream, it will align its direction with the other members and follow the stream. We motivate this in Section 5.2.1, where we initially consider the case where only one single agent is perceived. In Section 5.2.2, we iterate the single agent procedure on each perceived agent. The overall perceived stream velocity is the average of the single stream directions and speeds for each agent.

5.2.1 Perceiving a single agent

Let B be a single agent in A's FOV, and let x_A and x_B be the current positions of A and B, respectively. We define the perceived velocity $v_{per(A,B)}$ as an interpolation between B's actual velocity v_B and a vector $v_{dir(A,B)}$ of the same length that points along the line of sight between A and B. Formally, we let $v_{dir(A,B)} = \overline{(x_B - x_A)} \cdot \|v_B\|$ be the normalized vector between x_A and x_B scaled to the speed of B; see Figure 4. A factor $f_{A,B} = \rho \cdot d_{A,B}$ is used to angularly interpolate between the two vectors. Here, $\rho \in [0,1]$ is the local density in A's FOV, and $d_{A,B} = \frac{\|x_B - x_A\|}{d_{max}}$ is the relative distance between A and B. Thus, $f_{A,B} \in [0,1]$, and we use it to interpolate between v_B and v_{dir} along the smallest angle between the two.

If we assume a density of 1 in the above definition, the factor $f_{A,B}$ only depends on the relative distance between A and B. If B is on the edge of the view-distance of A, i.e. $\|x_B - x_A\| = d_{max}$, then $v_{per(A,B)}$ equals $v_{dir(A,B)}$. This makes A pursue a *follow strategy* because A is attracted to B's current position [25]. If, by contrast, A is close to B, then $v_{per(A,B)}$ is close to v_B, and A picks an *alignment strategy*.

The local density ρ is an extra factor to interpolate between the *follow strategy* and the *alignment strategy*: The higher the number of agents intersecting A's field of view, the more A is inclined to pick the *follow strategy*. This yields more compact crowd formations at higher densities and

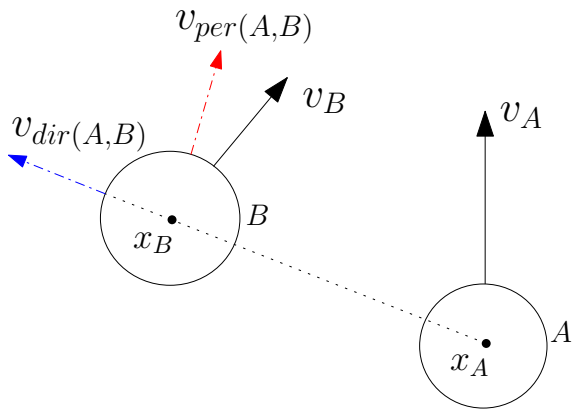

Figure 4: An example of the *perceived velocity* $v_{per(A,B)}$ based on an interpolation between v_B and $v_{dir(A,B)}$.

a wider crowd spread across the available free space at lower densities, which is a phenomenon observed in real crowds [14].

5.2.2 Perceiving the local stream

We define the local stream velocity perceived by agent A as the average of all perceived velocities that are taken into consideration, both with respect to direction and speed. We limit the total number of potential neighbors of A and only consider its five nearest neighbors that are currently in its FOV. This comparably small number corresponds to findings in research for flocks of birds [37], and has been used in related work, e.g. [28].

Let $\mathcal{N}_5$ be a set of up to 5 nearest neighbors of A. We define the *average perceived stream speed* s for A as follows:

$$s := \frac{1}{|\mathcal{N}_5|} \cdot \sum_{N \in \mathcal{N}_5} ||v_{per(A,N)}||. \tag{2}$$

The locally *perceived stream velocity* v_{stream} perceived by agent A is then defined as follows:

$$v_{stream} := s \cdot \frac{\sum_{N \in \mathcal{N}_5} v_{per(A,N)}}{||\sum_{N \in \mathcal{N}_5} v_{per(A,N)}||}, \tag{3}$$

which is the average perceived stream speed times the average direction of all perceived velocities scaled to unit length.

Since we define the local stream velocity as the average of a set of velocities, it can result in a null-vector when the corresponding velocities cancel each other out. In this case, agent A cannot adapt to the stream velocity, even if there is an actual stream of neighboring agents it should coordinate with. To avoid perceived stream velocities canceling each other out, we restrict the maximum angle between the velocities of agents A and B to strictly less than $\frac{\pi}{2}$. Perceived neighbors yielding a larger angle are not taken into consideration. Furthermore, due to the FOV with its viewing angle of π, perceived neighbors reside only in the closed halfplane $H^+(A)$ in front of A, which is induced by the line through x_A perpendicular to A's current velocity v_A. If either of these two restrictions is violated, perceived velocities may cancel each other out.

5.3 Incentive

Now that A's individual velocity v_{indiv} and the perceived stream velocity v_{stream} have been computed, we define the *incentive* $\lambda \in [0, 1]$ of A to interpolate between v_{indiv} and v_{stream}. We discuss how to compute λ in Section 5.3.1. In Section 5.3.2, we describe how to interpolate between v_{indiv} and v_{stream} using λ.

5.3.1 Computing the incentive

The incentive λ is defined by four different factors: *internal motivation* γ, *deviation* Φ, *local density* ρ, and *time spent* τ. We simulate the behavior of an agent A in a way such that – aside from the internal motivation factor – the most dominant factor among Φ, ρ and τ has the highest impact on $A's$ behavior. We define the incentive λ as follows:

$$\lambda := \gamma + (1 - \gamma) \cdot \max\big(\Phi, (1 - \rho)^3, \tau\big). \tag{4}$$

Internal motivation $\gamma \in [0, 1]$ determines a minimum incentive that an agent has at all times. This enables the simulation of various agent profiles such as a hurried agent or a strolling agent.
The local density factor ρ is defined in Section 5.1. For this factor, a non-linear relation with the incentive is desired, thus we use $(1-\rho)^3$ in Equation 4. The *deviation* factor Φ makes agent A leave a stream when v_{stream} deviates too much from its preferred individual velocity v_{indiv}. We introduce a minimum threshold angle ϕ_{min}. Whenever the angle between v_{stream} and v_{indiv} is smaller than ϕ_{min}, the factor Φ will be 0. This yields stream behavior unless the other factors determine a different strategy. If the angle is greater than ϕ_{min}, we gradually increase Φ up to a maximum deviation of $2\phi_{min}$. Angles greater than this threshold correspond to a deviation factor of 1, thus yielding individual steering behavior. Let $\phi_{dev} := \min(\angle(v_{indiv}, v_{stream}),\ 2\pi - \angle(v_{indiv}, v_{stream}))$ be the angle between the velocities. We define the deviation factor Φ as follows:

$$\Phi := \min\left(\max\left(\frac{\phi_{dev} - \phi_{min}}{\phi_{min}}, 0\right), 1\right). \tag{5}$$

The *time spent* factor τ is used to make stream behavior less attractive the longer it takes the agent to reach its goal. We initially calculate the expected time τ_{exp} agent A will need to get to its destination. How this is done depends on how A's individual velocity is calculated, i.e. what method is used as a black box in the initial step of our model. If, for instance, an indicative route is used [4], the expected time can be calculated by weighting the length of the route with the local density ρ. This value can then be mapped to an expected time value according to the agent's preferred speed. We keep track of the actual simulation time τ_{spent} that has passed since A has started moving. We then define the *time spent* factor τ as follows:

$$\tau := \min\left(\max\left(\frac{\tau_{spent} - \tau_{exp}}{\tau_{exp}}, 0\right), 1\right). \tag{6}$$

5.3.2 Using the incentive

Given the incentive λ, we interpolate between v_{indiv} and v_{stream} as follows: We rotate v_{stream} towards v_{indiv} in a similar manner as the interpolation between v_{dir} and v_B is performed for a single neighbor; see Section 5.2. Let β be the smallest angle between the two vectors, and let $\beta_{rot} = \beta\lambda$ be the rotation angle between 0 and β, based on the incentive λ. We then rotate v_{stream} towards v_{indiv} by β_{rot}. In this step, however, the lengths of v_{indiv} and v_{stream} are not the same in general. Therefore, we also linearly interpolate the lengths of these vectors.
We have tested the effects of the incentive on an agent that wishes to cross a large stream of other agents; see Figure 5. This example can be seen in the video that accompanies this work. We turned off the *time spent* computations to enhance the display of the effect of *internal motivation* and *deviation*. With an internal motivation of 1, we get a constant incentive of 1. The agent pushes through the stream to reach its goal position at the opposite side of the crowd. With an internal motivation of 0, and a threshold ϕ_{min} for the deviation factor of $\frac{\pi}{4}$, the agent is dragged away by the stream flow until the deviation factor causes the incentive to rise and makes the agent leave the stream.

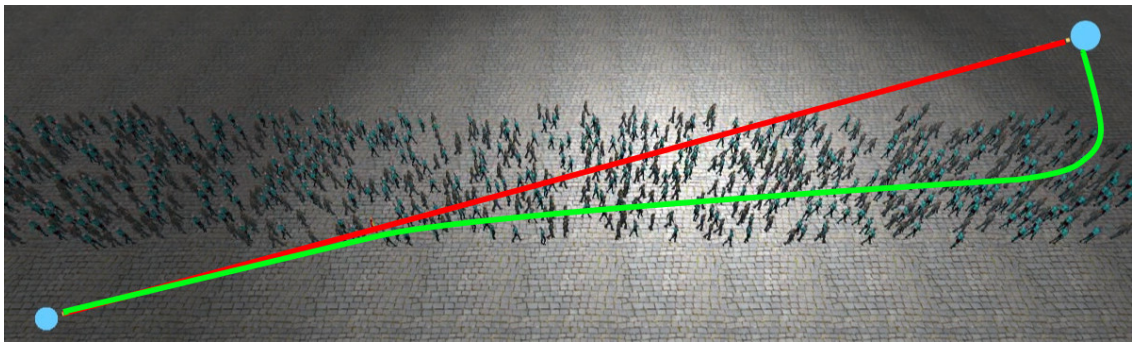

Figure 5: A stream of agents moving from left to right, and an agent trying to follow a path from the bottom left to the upper right corner. Red path: $\gamma = 1$. Green path: $\gamma = 0, \phi_{min} = \frac{\pi}{4}$. We refer the reader to our accompanying video for an animated sequence.

6 Experiments

Our model has been implemented in a framework based on the *Explicit Corridor Map* (ECM) [38]. The ECM is a time- and space-efficient navigation mesh for crowd simulation. All experiments have been conducted on a PC running Windows 7 with a 3.1 GHz AMD FX™ 8120 8-Core CPU, 4 GB RAM and a Sapphire HD 7850 graphics card with 2 GB of onboard GDDR5 memory. We used one CPU core for the computations. To compute a preferred individual velocity for each agent, we combined our model with the Indicative Route Method by Karamouzas et al. [4]. To benchmark and validate our model, we use the *Steerbench* framework by Singh et al. [39]. Our benchmarking score is defined as follows:

$$score = 50c + e + t. \tag{7}$$

It is comprised of the average number of collisions c per agent, the average kinetic energy e, and the average time t spent by an agent. Throughout all experiments, a lower score is considered to be a better result.

6.1 Scenarios

We used five different scenarios; see Figure 6. Preferred speeds were randomly chosen between 0.85 and 2.05 meters per second. In the *merging-streams* scenario, two groups with a total of 250 agents merge to pass through a bottleneck and split again afterward. The goal is to test whether two streams merge and split as an emergent phenomenon within our model. The *crossing-streams* scenario features two groups of 50 agents that approach each other in a perpendicular manner. The goal is to test whether different streams can cross each other without heavy interference. The *hallway1* scenario shows one group of 200 distributed agents, and the *hallway2* scenario shows two groups that each have 100 distributed agents. These agents traverse the hallway in either one direction (*hallway1*) or two opposing directions (*hallway2*). The goal is to test our model in medium-density scenarios. In the *narrow-x* scenarios, we use a narrow hallway of $3m$ and two comparably large groups of x agents ($x = 50$ and $x = 100$) with a radius of $0.25m$ each. The agents try to reach the opposite ends of the hallway. The goal of this experiment is to test our model in high-density situations.

In addition, we have measured the running times of our model in two scenarios, denoted as *military* and *hallway-stress*. *Military* features a 200 x 200 meters footprint of the McKenna MOUT training site at Fort Benning, Georgia, USA; see Figure 6 (bottom). It represents a scene with small passages, open squares and large areas of free space, which could be part of a gaming or simulation application. Agents are placed at the border and pick random goal positions at the opposite side of the scene. The goal is to test whether our model performs at interactive rates in these types of scenarios. In *hallway-stress*, we use a hallway of 30m to provide enough space for

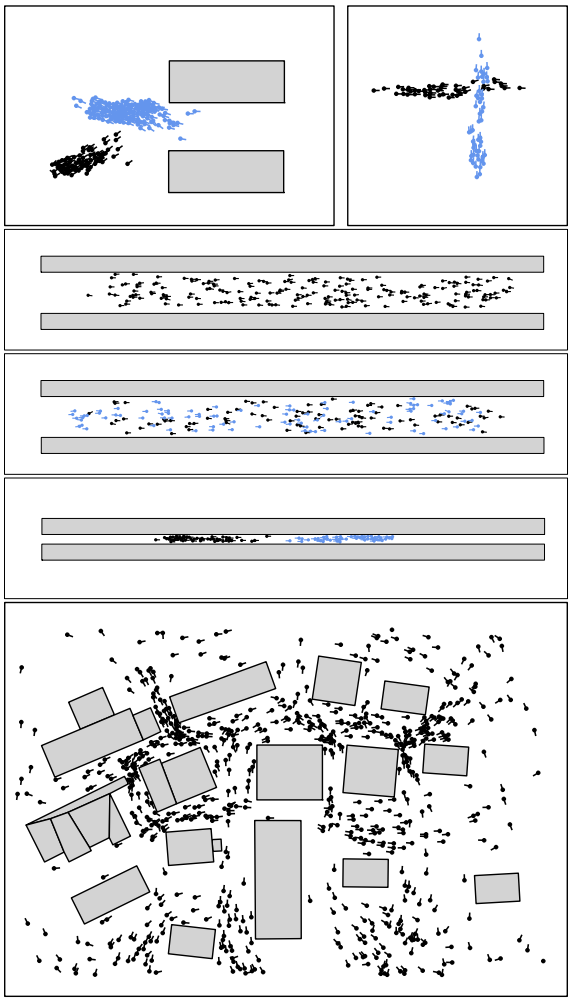

Figure 6: The different scenarios in our experiments are (from top to bottom): *merging-streams, crossing-streams, hallway1, hallway2, narrow-50* and *military*.

a large number of agents. The goal is to test whether our model performs at interactive rates for large numbers of agents when the environment enforces a high level of coordination.

6.2 Comparing different collision-avoidance methods

We have tested our streams model with three popular collision-avoidance methods [27, 28, 30]. For each method, we have used three different density measurements methods, and we have computed the mean Steerbench scores [39], averaged over 50 runs per density measurements and per agent. The mean scores are depicted in Figure 7. While the method by Karamouzas et al. [28] performs equally well in low- to medium-density scenarios, the method by Moussaïd et al. clearly shows better scores in high-density scenarios such as the *narrow-50* example. By contrast, the ORCA method [27] struggles with high-density scenarios. This is mainly due to a significantly higher number of collisions between the agents.

6.3 Testing the effect of streams

We have compared our streams approach to the same scenarios when only individual behavior is being displayed. We use the Indicative Route Method (IRM) [4] together with the collision-

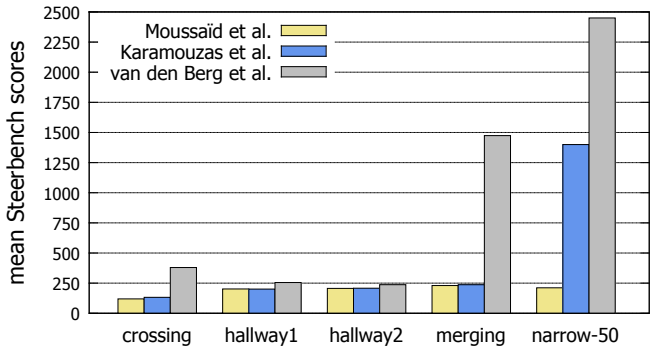

Figure 7: Mean Steerbench scores of the three different collision avoidance methods for our test scenarios. The scores are averaged over 50 runs per density measurements and per agent. In all our experiments, lower scores are better.

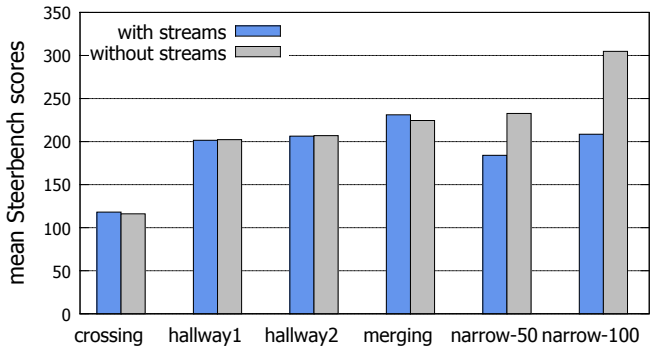

Figure 8: Mean Steerbench scores for the scenarios with our streams model turned on and off. The scores are averaged per agent over 50 runs.

avoidance method by Moussaïd et al. [30] because they yielded the best results in our experiments; see Section 6.2. We tested both low-density scenarios such as *hallway1* and *hallway2* and high-density scenarios such as *narrow-50* and *narrow-100*.

Figure 8 shows the corresponding mean Steerbench scores per agent over 50 runs per scenario. Both models perform equally well in low- to medium-density scenarios such as *crossing-streams*, *hallway1*, *hallway2* and *merging-streams*. This is what we expected because coordination among crowd members is of minor importance. Streams are formed less often in these types of scenarios, and our model tends to behave similarly to the IRM with vision-based collision-avoidance. Only the *merging-streams* scenario yields slightly higher mean Steerbench scores when streams are being used. This is caused by a higher number of collisions when the two streams split. Although this yields higher (worse) Steerbench scores, we believe the behavior matches real-life situations.

In high-density scenarios such as *narrow-50* and *narrow-100*, our model greatly improves crowd coordination, and the number of collisions is significantly reduced. Turning off stream behavior frequently results in a complete deadlock as agents try to exploit any small gaps in the crowd without coordinating with the slowly moving agents in front of them. Figure 9 shows the average percentage of agents that did not reach their goal in a total time of 200 seconds with stream behavior turned on and off. Note that 200 seconds are more than enough to let the agents reach their goal in these comparably small environments when no deadlocks occur. An example of such a deadlock when stream behavior is turned off appears in the accompanying video. When we turn on stream behavior, agents align their directions of movement, and deadlocks occur significantly less often.

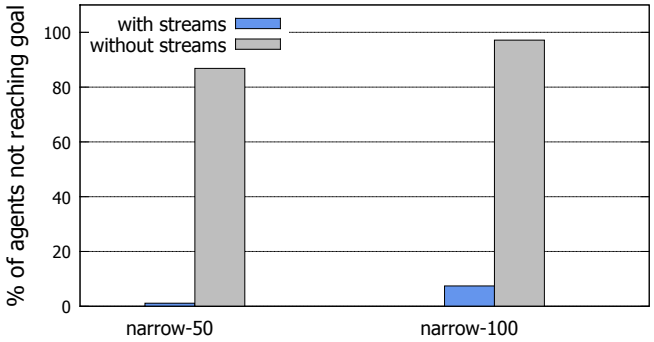

Figure 9: Percentage of agents that did not reach their goal in high-density scenarios with our streams model turned on and off, averaged over 50 runs each.

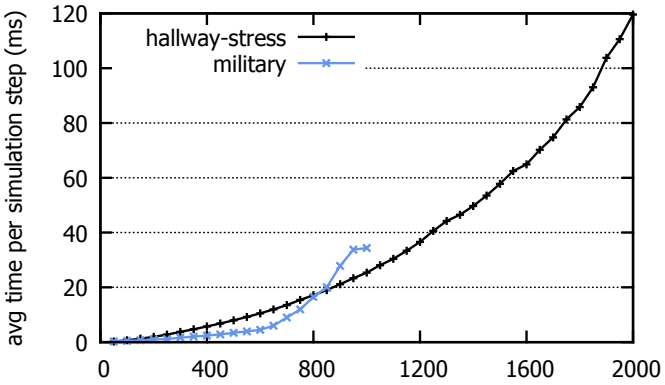

Figure 10: Average running times to compute one step of the simulation (in ms) for an increasing number of agents in the *military* and *hallway-stress* scenarios. Each measurement is the average of 10 runs for the same number of agents. Deadlocks frequently occur for more than 1000 agents in *military*. In the *hallway-stress* scenario, we could simulate up to 2000 agents simultaneously without any deadlocks.

6.4 Performance

Our final experiment shows that our model runs at interactive rates. Figure 10 shows the average running times needed to compute one step of the simulation for an increasing number of agents in the *military* and *hallway-stress* scenarios. Each measurement shows the average step time of 10 runs with a maximum of 5000 steps each. The results show that we could simulate up to 1000 agents at interactive rates in the *military* scenario. For higher numbers, deadlocks frequently started to occur, which is what we expected given the size of the scene compared to the number of agents. In the *hallway-stress* scenario, we could simulate up to 2000 agents simultaneously at interactive rates on a single CPU. We conclude that our model runs at interactive rates in typical gaming or simulation scenarios, even when coordination among the agents is high.

7 Limitations

Many problems with dense crowds are caused by the global path planning step of the simulation cycle. Whenever the global paths of a large number of agents intersect in the same point of the environment, the probability for deadlocks and an overall low throughput is high. This happens at the corners of obstacles when many agents are following the shortest path around these obstacles. Our model cannot prevent deadlocks entirely because it is designed to resolve problems on a local level. It is not designed to let agents dynamically re-plan their global path, or to make use of the full free space around obstacles. Improvements on a higher planning level are the subject of

current research. Results in that field may strengthen the applicability of our model even more in the future.
We have shown that our model allows real-time simulation for up to 2000 autonomous agents in medium-sized environments that contain both narrow passages and areas of open space; see Figure 10. However, since agents are simulated as individuals, computation is still expensive. When the application requires tens of thousands of agents with only a few distinct goals, a flow-based model is the better choice.
Lastly, the coordination of real crowds depends significantly on social factors and group behavior. Our model provides a general framework of how an agent perceives and reacts to neighboring agents on a purely geometrical level, and it does not yet take higher-level aspects into account.

8 Conclusion and Future Work

We have introduced a novel crowd simulation model that combines the advantages of agent-based and flow-based paradigms while only relying on local information. We interpolate an agent's steering strategy between individual behavior and coordination with the crowd. Local streams determine an agent's trajectory when local crowd density is high. This allows the simulation of large numbers of autonomous agents at interactive rates. Our model can handle arbitrary and dynamically changing crowd densities without additional user input. Furthermore, our model supports existing techniques for computing a global path and derived preferred velocities, as well as for handling local collision avoidance. This makes our model flexible and easy to integrate into existing crowd simulation frameworks.
We have validated our model with the Steerbench framework [39] by measuring the average numbers of collisions, expended kinetic energy, and time spent. Experiments show that our model works as well as existing agent-based methods in low- to medium-density scenarios, while showing a clear improvement when handling large crowds in densely packed environments. The occurrence of collisions and deadlocks, as well as the average kinetic energy and time spent is significantly reduced. We conclude that our model yields crowd behavior that matches real-life behavior better than existing agent-based crowd simulation models for dense scenarios. These conclusions are also validated in the accompanying video.
Our model is general and flexible enough to be the basis for many interesting future extensions. While the underlying navigation mesh in our experiments allows for dynamic updates [40], our model does not yet consider dynamic re-planning for agent navigation. Physical interactions between the agents and obstacles could be an interesting extension. High crowd density may lead to breaking fences or walls in emergency situations as discussed in [41]. It would also be interesting to use our model in heterogeneous environments with various terrain or region types [42].
The flexibility to use any global planning method and any local collision-avoidance method as a black box makes our model applicable to a wide range of research fields that require the simulation of autonomous virtual agents. We believe that our model can form a basis for improving crowd movement in future gaming and simulation applications, in CGI-enhanced movies, in urban city planning software, and in safety training applications.

9 Acknowledgements

This research was partially funded by the COMMIT/ project, http://www.commit-nl.nl.

References

[1] G. Aschwanden, J. Halatsch, and G. Schmitt. Crowd simulation for urban planning. *Proceedings of eCAADe 2008*, 2008.

[2] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and Evacuation Dynamics*, 21:21–58, 2002.

[3] X. Pan, C. S. Han, K. Dauber, and K. H. Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2):113–132, 2007.

[4] I. Karamouzas, R. Geraerts, and M. Overmars. Indicative routes for path planning and crowd simulation. *4th International Conference on Foundations of Digital Games*, pages 113–120, 2009.

[5] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[6] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. Group behavior from video: A data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–118, 2007.

[7] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168, 2006.

[8] W. Kerr and D. Spears. Robotic simulation of gases for a surveillance task. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2905–2910, 2005.

[9] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. C. Lin, and D. Manocha. Pledestrians: A least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 119–128. Eurographics Association, 2010.

[10] D. Thalmann and S. R. Musse. *Crowd Simulation, Second Edition.* Springer, 2013.

[11] N. Pelechano, J. Allbeck, and N. Badler. *Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation).* Morgan and Claypool Publishers, 2008.

[12] R. L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35(1):169–182, 2003.

[13] L. C. A. Pimenta, N. Michael, R. C. Mesquita, G. A. S. Pereira, and V. Kumar. Control of swarms based on hydrodynamic models. In *Proceedings of the 2009 IEEE Internation Conference on Robotics and Automation*, pages 1948–1953, 2008.

[14] D. Helbing, P. Molnar, I. J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B*, 28(3):361–384, 2001.

[15] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 39(1):1–24, 2005.

[16] P. M. Torrens. Moving agent pedestrians through space and time. *Annals of the Association of American Geographers*, 102(1):35–66, 2012.

[17] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108, 2007.

[18] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 19–28, 2005.

[19] S. Lemercier, A. Jelic, R. Kulpa, J. Hua, J. Fehrenbach, P. Degond, C. Appert-Rolland, S. Donikian, and J. Pettré. Realistic following behaviors for crowd simulation. In *Computer Graphics Forum*, volume 31, pages 489–498, 2012.

[20] G. Antonini, M. Bierlaire, and M. Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667 – 687, 2006.

[21] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: A predictive approach. *Computer Graphics Forum*, 26(3):665–674, 2007.

[22] G. Vizzari, L. Manenti, and L. Crociani. Adaptive pedestrian behaviour for the preservation of group cohesion. *Complex Adaptive Systems Modeling*, 1(1):1–29, 2013.

[23] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley Press, 1949.

[24] R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 122:1–122:8. ACM, 2009.

[25] C. W. Reynold. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 25–34, 1987.

[26] V. Kountouriotis, S. Thomopoulos, and Y. Papelis. An agent-based crowd behaviour model for real time crowd behaviour simulation. *Pattern Recognition Letters*, 44(0):30 – 38, 2014.

[27] J. van den Berg, S. Guy, M. C. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, pages 3–19. Springer, 2011.

[28] I. Karamouzas, P. Heil, P. van Beek, and M. Overmars. A predictive collision avoidance model for pedestrian simulation. In *Motion in Games*, pages 41–52. Springer, 2009.

[29] J. H. Park, F. A. Rojas, and H. S. Yang. A collision avoidance behavior model for crowd simulation based on psychological findings. *Computer Animation and Virtual Worlds*, 24(3-4):173–183, 2013.

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

[31] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. *ACM Trans. Graph.*, 29(4):123:1–123:9, July 2010.

[32] E. B. Werner and C. G. Rossi. *Manual of visual fields*. Churchill Livingstone New York, 1991.

[33] John C Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2008.

[34] R. Geraerts and M. Overmars. The corridor map method: Real-time high-quality path planning. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 1023–1028, 2007.

[35] G. K. Still. *Crowd Dynamics*. PhD thesis, University of Warwick, 2000.

[36] J. J. Fruin. *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[37] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4):1232–1237, 2008.

[38] R. Geraerts. Planning short paths with clearance using Explicit Corridors. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 1997–2004, 2010.

[39] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. Steerbench: A benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds*, 20(5-6):533–548, 2009.

[40] W. G. van Toll, A. Cook IV, and R. Geraerts. A navigation mesh for dynamic environments. *Computer Animation and Virtual Worlds*, 23(6):535–546, 2012.

[41] S. Kim, S. Guy, and D. Manocha. Velocity-based modeling of physical interactions in multi-agent simulations. *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 125–133, 2013.

[42] N. Jaklin, A. Cook IV, and R. Geraerts. Real-time path planning in heterogeneous environments. *Computer Animation and Virtual Worlds*, 24:285–295, 2013.