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

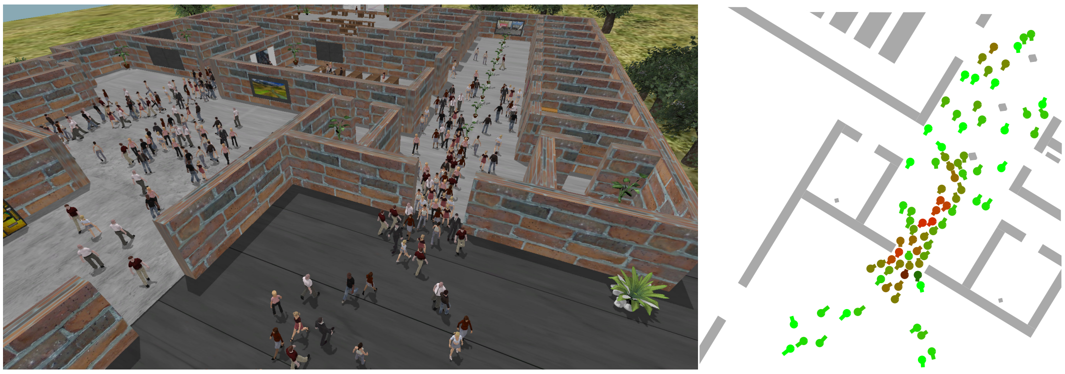

Figure 1: *Left*: A dense crowd of agents collaboratively moves through a narrow doorway. *Right*: A $2D$ representation of the doorway shows that each agent interpolates between individual behavior (green) and coordinated behavior (red).

Abstract

We present a 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. Our model can be used with any existing global path planning and local collision-avoidance method. We show that our model reduces the occurrence of deadlocks and yields visually convincing crowd behavior for high-density scenarios while maintaining individual agent behavior at lower densities.

Keywords: crowd simulation, multi-agent system, autonomous virtual agents

1 Introduction

Crowd simulation models can be divided into agent-based simulations and flow-based simulations. Agent-based simulations focus on the behaviors of each individual in the crowd. 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 a crowd as one large entity. 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. 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* [1].

2 Overview of our model

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), which 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 $\phi = 180°$.

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 3.1.
3. We compute the locally *perceived stream velocity* of agents near A; see Section 3.2.
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 3.3.
5. The interpolated velocity is passed to a collision-avoidance algorithm. Our model is independent of the exact method that is used.

3 Streams

We define *streams* as flows of people that coordinate their movement by either aligning their paths or following each other. This leads to fewer collisions and abrupt changes in the direction of movement. A dominant factor is the local density ρ.

3.1 Computing local density information

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 FOV. We sum up the area $\Delta(N)$ occupied for each agent $N \in \mathcal{N}$ and divide it by the total area $\Delta(FOV)$ of A's FOV. A 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. Formally, we define the crowd density ρ as follows:

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

3.2 The perceived stream velocity

Let B be a single agent in A's FOV, and let x_A and x_B be their current positions, respectively. We define the perceived velocity $v_{per(A,B)}$ as an interpolation between B's 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; see Figure 2. Let $\rho \in [0,1]$ be the local density in A's FOV, and let $d_{A,B} = \frac{\|x_B - x_A\|}{d_{max}}$ be the relative distance between A and B. A factor $f_{A,B} = \rho \cdot d_{A,B}$ is used to angularly interpolate between v_B and $v_{dir(A,B)}$. The larger ρ is the more A is inclined to pick a *follow strategy* rather than an *alignment* strategy.

Let $\mathcal{N}_5$ be a set of up to 5 nearest neighbors of A. To avoid perceived stream velocities canceling each other out, we restrict the angle between the velocities of A and each neighbor to strictly less than $\frac{\pi}{2}$. We define the *average perceived stream speed* s as follows:

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

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

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

3.3 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:

$$\lambda := \gamma + (1-\gamma) \cdot \max\left(\Phi, (1-\rho)^3, \tau\right). \qquad (4)$$

Internal motivation $\gamma \in [0,1]$ determines a minimum incentive that an agent has at all times. For the local density ρ, a non-linear relation with the incentive is desired, and we use $(1-\rho)^3$.

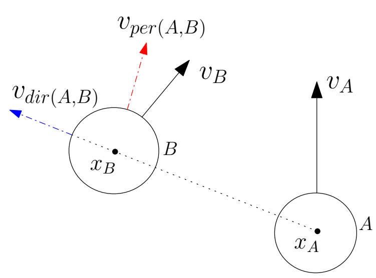

Figure 2: An example of the *perceived velocity* $v_{per(A,B)}$ based on an interpolation between v_B and $v_{dir(A,B)}$.

The *deviation* factor Φ makes agent A leave a stream when v_{stream} deviates too much from v_{indiv}. We use a 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 ϕ_{dev} be the smallest angle between v_{indiv} and v_{stream}. 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. We keep track of the actual simulation time τ_{spent} that has passed since A has started moving. We define the *time spent* factor τ as follows:

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

Finally, let $\beta = \phi_{dev}\lambda$ be the deviation angle angle scaled by the incentive. We rotate v_{stream} towards v_{indiv} by β. In general, the lengths of v_{indiv} and v_{stream} are not equal. Therefore, we also linearly interpolate the lengths of these vectors. The resulting velocity is the new velocity for agent A in the next simulation cycle.

4 Experiments

Our model has been implemented in a framework based on the *Explicit Corridor Map* [2]. We use one CPU core of a PC running Windows 7 with a 3.1 GHz AMD FX$^{\text{TM}}$ 8120 8-Core CPU, 4 GB RAM and a Sapphire HD 7850 graphics card with 2 GB of onboard GDDR5 memory. To compute v_{indiv}, we combined our model with the Indicative Route Method (IRM) [3]. To benchmark and validate our model, we use the *Steerbench* framework [4]. 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. A lower score is considered to be a better result.

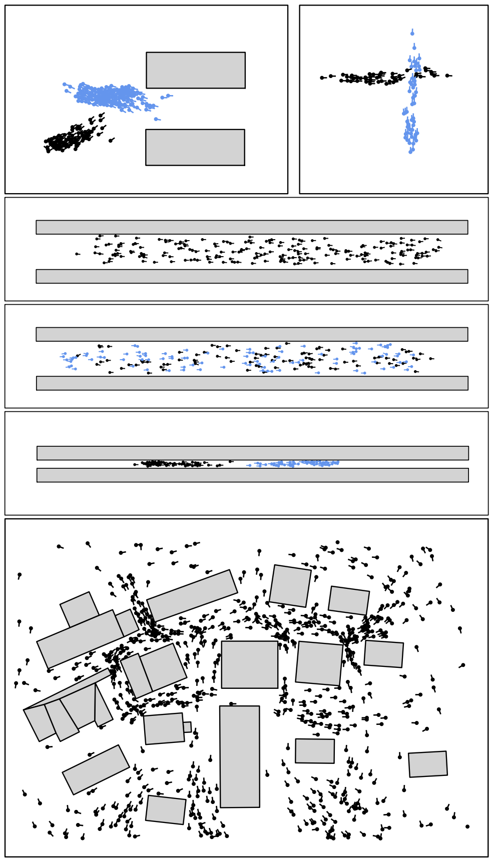

Figure 3: The different scenarios in our experiments are (from top to bottom): *merging-streams, crossing-streams, hallway1, hallway2, narrow-50* and *military*.

We used six different scenarios; see Figure 3. Preferred speeds were randomly chosen between 0.85 and 2.05 meters per second. We have tested our model with three popular collision-avoidance methods [5, 6, 7]; see Figure 4. We have also compared our model to the same scenarios when only individual behavior is being displayed. Here, we use the IRM together with the collision-avoidance method by Moussaïd et al. [7] because this yielded the best results. Figure 5 shows the corresponding mean Steerbench scores per agent over 50 runs per scenario. Figure 6 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. Figure 7 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. Our model runs at interactive rates in typical gaming or simulation scenarios, even when coordination among the agents is high.

5 Conclusion and Future Work

We have introduced a crowd simulation model that interpolates an agent's steering strategy between in-

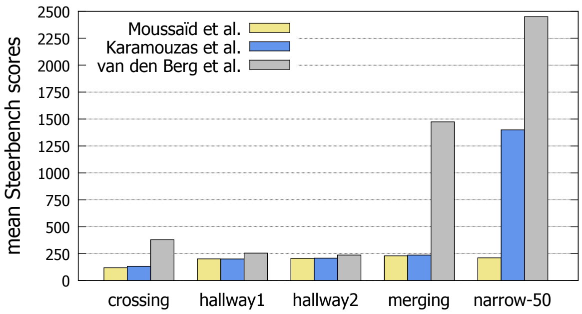

Figure 4: Mean Steerbench scores of the three different collision avoidance methods for our test scenarios. The scores are averaged over 50 runs per agent. In all our experiments, lower scores are better.

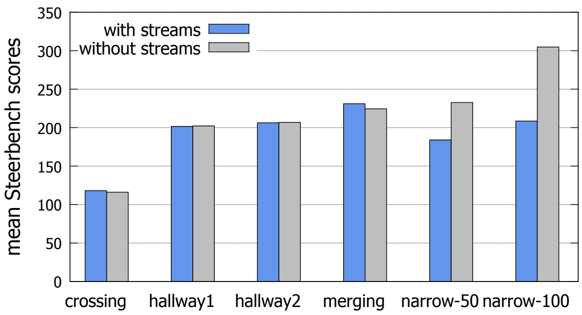

Figure 5: Mean Steerbench scores for the scenarios with our streams model turned on and off. The scores are averaged per agent over 50 runs.

dividual 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.

We have validated our model with the Steerbench framework [4] 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. These conclusions are also validated in the accompanying video.

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. For further details on our model, we refer the interested reader to the full-length version of this paper [8].

Acknowledgements

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

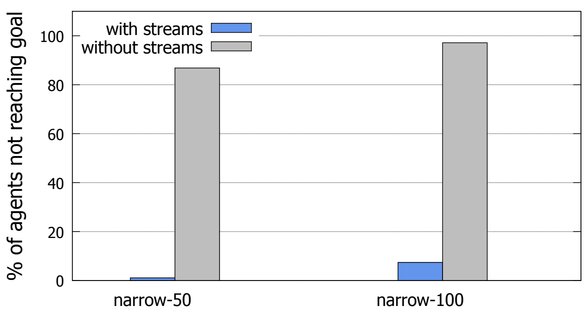

Figure 6: Percentage of agents that did not reach their goal in high-density scenarios with our streams model turned on and off, averaged over 50 runs each.

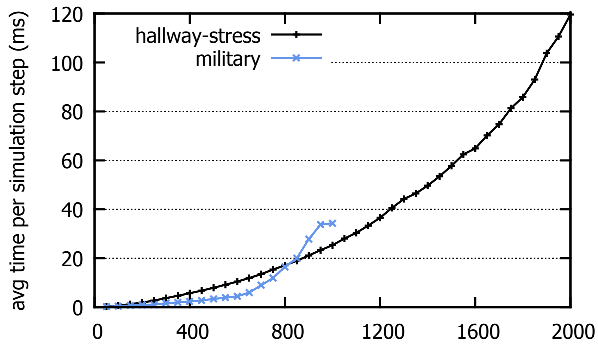

Figure 7: Average running times to compute one step of the simulation (in ms) for an increasing number of agents in the *military* and *hallway-stress* scenarios. Each measurement is the average of 10 runs for the same number of agents. Deadlocks frequently occur for more than 1000 agents in *military*. In the *hallway-stress* scenario, we could simulate up to 2000 agents simultaneously without any deadlocks.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Arthur van Goethem, Norman Jaklin, Atlas Cook IV, and Roland Geraerts. On streams and incentives: A synthesis of individual and collective crowd motion. Technical Report UU-CS-2015-005, Department of Information and Computing Sciences, Utrecht University, March 2015.