

Getting out of the way: collision-avoiding pedestrian models compared to the real world

Gregor Lämmel¹ and Matthias Plaue²

¹ Transport Systems Planning and Transport Telematics, TU Berlin
Sekt. SG12, Salzufer 17-19
10587 Berlin, Germany
laemmel@vsp.tu-berlin.de

² Department of Mathematics, TU Berlin
Sekt. 6-4, Straße des 17. Juni 136
10623 Berlin, Germany
plaue@math.tu-berlin.de

<http://www.math.tu-berlin.de/projekte/smdpc>

Abstract. Numerical simulation of human crowds is a challenging task and a number of models to simulate pedestrian dynamics on a microscopic level have been established. One aim of those models is to reproduce a realistic, and in particular collision-free, movement of crowds in complex environments. This work investigates three approaches on their capability to reproduce a collision-free movement of pedestrian crowds in complex dynamic environments. The baseline model is the well-known social force model. While in the social force model pedestrians do not explicitly avoid each other, the second model extends the social force model to avoid collisions explicitly. The observed collision-avoiding behavior produced by the third model is reached by calculating velocity obstacles. These are obstacles in the velocity space, meaning that if a pedestrian chooses a velocity that lies inside the velocity obstacle, then a collision occur at some time. This work discusses the models and their integration in a multi-agent simulation framework. The models are tested on data from a real-world experiment conducted at Technische Universität Berlin. In this experiment, two pedestrian flows intersected at an angle of 90 degrees. The models' performance with regard to the reproduction of a realistic crowds movement and their computational complexity are discussed in this work.

Keywords: intersecting pedestrian flows, multi-agent simulation, human crowd experiments

1 Introduction

One possible approach to model the movement of crowds are Cellular Automata (CA) [14]. CA models represent the environment with a grid-like structure, where each cell of the grid may contain at most one pedestrian at a time. CA models have often been used for the simulation of evacuation scenarios. A common

problem with existing CA models, however, is that they do not model complex wayfinding, since in most CA models wayfinding is implemented via a globally defined potential field. In theory, it would be possible to assign an individual potential field to each pedestrian. This approach, however, would be too complex in terms of computational costs for large scenarios.

Other simulation concepts use (discretized) differential equations similar to equations known from the description of molecular dynamics [10, 8]. Probably the best known model based on the molecular dynamics analogy is the social force model [11]. In social force model simulations, each pedestrian has a desired velocity towards a desired destination and adapts her current velocity accordingly. A pedestrian's incentive to avoid obstacles such as other pedestrians is modeled by repelling forces. Force-based models are well understood and have reasonable computational costs.

There is another class of models that try to achieve collision-free pedestrian movement in complex environments. These models are based on the so-called configuration space obstacle approach and have their foundation in computational geometry and robotics. In this context the configuration space describes all possible locations a pedestrian can reach. Locations that cannot be reached are the so-called configuration space obstacles. For this approach the pedestrians and the obstacles in the environment (e.g., walls and other pedestrians) are represented as a set of simple polygons. A path through the environment is collision-free if and only if the path does not intersect the Minkowski sum of the polygonal obstacles with the polygonal representation of the pedestrian reflected in her reference point [1].

An extension to the configuration space obstacle approach is the velocity obstacle approach [5]. Similar to the configuration space obstacles the velocity obstacles describe all velocities a pedestrian can choose that will lead to a collision at some point in time assuming straight movement and no acceleration or rest of the pedestrians. In the velocity obstacle approach every pedestrian chooses at each point in time a velocity that avoids collision and is close to the desired velocity.

This work investigates three approaches with regard to their capability of reproducing collision-free realistic pedestrian movement. First, there is the social force model as it is introduced in [11]. While the social force model does not let the pedestrians avoid collisions explicitly the collision-avoiding force-based model proposed by [25] takes potential collisions into account. This means the repelling forces do not only depend on the locations of the obstacles and pedestrians but also on their velocities. When calculating the actual force that affects a given pedestrian the model calculates the minimal time of closest approach to all other pedestrians and obstacles in the environment. All pedestrians are then projected up to this point in time assuming constant movement. The influencing force on the pedestrian in question is calculated based on this projection and weighted by the inverse of the minimal time of closest approach. This leads to a behavior that tries to avoid the next potential collision under the assumption of constant movement. Like in other force-based models, too, each pedestrian has a

desired velocity to which she adapts in a given time if free movement is possible. However, most existing force-based models do not address the important problem of complex wayfinding. In our approach, complex wayfinding is implemented via a navigation graph and a least cost path search: a force that lets a pedestrian move along the navigation graph replaces the simple desired velocity vector. Thus, it is possible to simulate pedestrian movements to multiple destinations in a complex environment.

The third approach discussed in this work is based on the reciprocal velocity obstacle approach [24]. But instead of calculating valid velocities during every time step the pedestrians choose valid accelerations preferably close to the desired one. This approach leads to a similar result like choosing a valid velocity directly but lets the model be better integrated with force-based models. This integration is needed since the calculation of the desired velocity and the short-range interaction is still based on forces. Like with the first and the second approach, the desired movement is based on a force that lets each agent move along a navigation graph.

We compare the results of simulations performed by the three models with data obtained from a real-world experiment conducted at Technische Universität Berlin, see Fig. 1.

The remainder of this work is organized as follows. Section 2 gives a discussion of all three simulation models and describes extensions that have been made in order to integrate those models in a multi-agent simulation framework for the simulation of pedestrian crowds in complex dynamic environments. Experimental results are shown in Section 3. A discussion on this results is given in Section 4. Section 5 concludes this work and gives a short outlook.

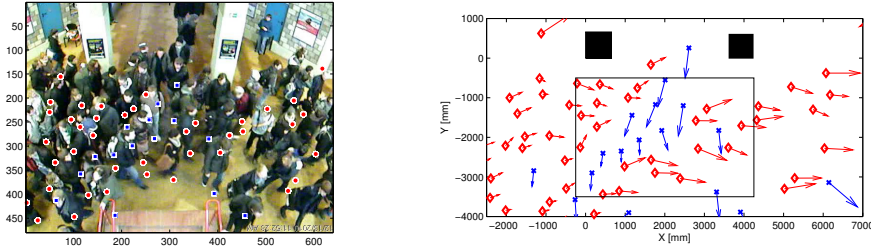


Fig. 1. Experiment with two pedestrian flows intersecting at 90 degrees at time step $t = 88.3$ s.. Red diamonds: group A, blue crosses: group B. Arrows indicate the velocity of the pedestrians.

2 Simulation models and framework

The simulation models investigated are all fully integrated into MATSim. MATSim stands for Multi-Agent Transport Simulation and provides a toolbox for various kinds of transport simulations (see, e.g., [23]). MATSim’s major field of application is the simulation of vehicular traffic in large-scale scenarios.

In recent times MATSim has also been used for the simulation of pedestrians in the evacuation context [17, 16, 4]. The underlying traffic flow simulation relies in all this works on a queue model [6]. The queue model has the advantage of being highly efficient while retaining important physical parameters like free speed travel time or flow capacity of links. However, when it comes to crossing pedestrian streams it seems to be questionable if a queue model can simulate those streams adequately.

This work extends MATSim with the capability of simulating pedestrians in a two-dimensional continuous space. The following sub-sections discuss the agent movement model. We distinguish between two stages in the movement models. The first stage deals with the low level movement of the agents, meaning collision avoidance, velocity adaptation and so on. The second stage deals with the more high level movement of the agents, meaning moving along a given route. While for all three models the second stage is implemented in the same manner, for the first stage two general approaches are investigated. The social force model and the collision-predicting force model are driven by repelling and attracting forces, which are “pushing” the agents through the environment. In contrast to this approach, the reciprocal velocity model calculates velocities that let the pedestrians avoid collisions. Detailed model descriptions are given in the following.

2.1 Social Force Model

Here and in the following the space that a pedestrian occupies in the 2D space is approximated by a circular disk. The true shape of a pedestrian indisputable differs from this simplification. Some authors argue that the space is better approximated by an ellipse [13] and what is more this ellipse should be velocity dependent [3]. While a velocity dependent elliptical representation seems to be a plausible choice, since pedestrians need more space in walking direction when moving faster, it is also more consuming in terms of computational costs. For that reason we stick with the simpler circular representation for the benefit of faster computation.

The general force model is defined according to an analogue of Newton’s law ($F = m \cdot a$).

$$\mathbf{f}_i(t) = \frac{m_i}{\tau}(\mathbf{v}_i^0(t) - \mathbf{v}_i(t)) + \sum_{j \neq i} \mathbf{f}_{ij}(t), \quad (1)$$

where $\mathbf{v}_i^0(t)$ is the desired velocity vector for agent i at time t . The term $\mathbf{v}_i(t)$ denotes the agent’s actual velocity at time t . The time constant τ describes the time that is needed to adjust the actual velocity to the desired velocity. The

second term of the equation builds the sum over all influential entities j in the environment (i.e., other agents, walls, and obstacles). Each of those entities emit a repelling (or attracting) force to agent i .

Many different force models have been discussed in recent years (see, e.g., [19] for an overview), most of them are build on the so-called social force model introduced by [11]. The basic social force model implicitly reproduces collision-avoiding behavior as it can be observed in real-world situations. It has been shown that the model works particularly well in high density conditions, such as one observes in evacuation situations [9]. The social force model is defined as follows:

$$\mathbf{f}_i(t) = \frac{m_i}{\tau}(\mathbf{v}_i^0(t) - \mathbf{v}_i(t)) + \sum_{j \neq i} \mathbf{f}_{ij}(t) + \sum_W \mathbf{f}_{iW}(t), \quad (2)$$

where \mathbf{f}_{ij} denotes the repelling force emitted by agent j towards agent i and \mathbf{f}_{iW} denotes the repelling force emitted by obstacle W (e.g., walls) towards agent i . Both forces are calculated in the same way. For that reason only the calculation of \mathbf{f}_{ij} is discussed in detail here. The interested reader is referred to [9] for a detailed discussion.

$$\mathbf{f}_{ij}(t) = A_i \exp((r_{ij} - d_{ij})/B_i) \mathbf{n}_{ij} + g(k(r_{ij} - d_{ij}) \mathbf{n}_{ij} + \kappa(r_{ij} - d_{ij}) \Delta v_{ji}^t \mathbf{t}_{ji}). \quad (3)$$

The first term of the equation's right hand side describes the so-called repelling social force between agent i and agent j ; A_i and B_i are constants, $r_{ij} = r_i + r_j$ is the sum of i 's and j 's radius, and d_{ij} is the distance between the disks centers of i and j . The term $\mathbf{n}_{ij} = \mathbf{d}_{ij}/d_{ij}$ describes the unit vector pointing from agent j to agent i . The second term of the equation's right hand side expresses the physical force between agent i and agent j . The physical force only acts if a collision actually occurs. This is guaranteed by the function $g(X)$, which is zero as long as no collision occurs and equal to its argument else. If a collision occurs there are two additional forces. First, there is a strong repelling force given by $k(r_{ij} - d_{ij}) \mathbf{n}_{ij}$, with k a constant. Second, there is a strong tangential force given by $\kappa(r_{ij} - d_{ij}) \Delta v_{ji}^t \mathbf{t}_{ji}$. Again, κ is a constant, \mathbf{t}_{ji} is the tangential unit vector ($= \mathbf{n}_{ij}$ rotated by $\pi/2$), and Δv_{ji}^t is the tangential velocity difference of agent i 's and j 's velocities. The social force model is straightforward to implement and has reasonable computational costs. One disadvantage of the model is that potential collisions are not explicitly considered. A novel approach that relies on the social forces paradigm but considers potential collision explicitly has been proposed by [25].

2.2 Collision-predicting force model

The collision-predicting force model proposed by [25] also calculates repelling forces but it projects the present scene into the future in order to detect potential collisions.

In the model each agent i computes for each other agent j in the environment the angle θ_{ij} between \mathbf{d}_{ij} and \mathbf{v}_{ij} . Where \mathbf{d}_{ij} denotes the vector pointing from agent j 's position to agent i 's position and \mathbf{v}_{ij} denotes the relative velocity

between both agents. If $|\theta_{ij}| > \pi/4$ then $t_{ij} = \infty$. Otherwise t_{ij} reflects the time of the closest approach of both agents under the assumption that neither agent i nor agent j changes her velocity or direction of movement. The agent then takes the minimum of these times $t_i = \min_j(t_{ij})$.

Afterwards the agent computes the configuration of the environment for t_i by again assuming that none of the pedestrians change the velocity or direction of movement. Let $\mathbf{d}'_{ij}(t_i)$ denote the predicted vector pointing from agent i to agent j at time t_i . The agent j 's influence on agent i (second term in Eq. (1)) is

$$\mathbf{f}_{ij}(t) = A_i \frac{v_i(t)}{t_i} \exp(-d_{ij}(t)/B_i) \frac{\mathbf{d}'_{ij}(t_i)}{d'_{ij}(t_i)}. \quad (4)$$

The constants A_i and B_i are free parameters in the model. Eq. (4) is not only applicable for other agents in the environment but also for any object that is not moving, like walls or obstacles.

2.3 Reciprocal velocity obstacle model

Another approach to model a collision-avoiding behavior in a pedestrian simulation extends a well-known approach to calculate collision-free movement of robots in complex geometries. Consider for now that there is only one agent who wants to navigate through a static environment without colliding with any obstacle (e.g., walls, furniture, etc.). The problem for the agent is to decide for any location in the environment if she would collide with an obstacle if she would be at this location. If the agent would have no dimension (i.e., point shape) then this problem would be straightforward to solve. However, since the agents represent pedestrians in the underlying context, they have a polygonal shape. To decide whether an agent collides with an obstacle at a certain location in the environment one must decide whether the polygon representing the agent intersects with some other polygon (i.e., obstacle). The computation used to determine whether two polygons intersect is a rather complex operation and would lead to long computation times.

Instead of calculating the intersections directly one can determine collision-free locations by a simple approach that relies on so-called configuration space obstacles. For a detailed description of this approach the reader is referred to [1]. The basic procedure works as follows: First, each agent is assigned to a reference point relative to her shape. If the agent has a circle shape then this could for example be the center of the circle. Second, the shape of the agent has to be reflected in its reference point. In the case of a circle with the reference point locate at the center this operation has no effect. Third, for every obstacle in the environment one has to compute the Minkowski sum of the obstacle and the agent's reflected shape. The Minkowski sum of two polygons A and B in Euclidean space is defined as $A \oplus B = \{\mathbf{a} + \mathbf{b} | \mathbf{a} \in A, \mathbf{b} \in B\}$. The result of this operation is a set of so-called configuration space obstacles. The position of an agent is collision-free or admissible as long as her reference point lies in no

configuration space obstacle. With this method it is easy to calculate collision-free paths for a single agent in complex geometric environments. When it comes to a pedestrian simulation with a great number of agents, this approach does not longer work.

An extension of the configuration space approach for dynamic environments is the so-called velocity obstacle approach [5]. A velocity obstacle is a set of velocities that would lead to a collision at some future time under the assumption that none of the moving entities change their velocity. The authors demonstrate that the velocity obstacle approach result in near-optimal collision-free paths with respect to distance or travel time. However, if two agents approach in a simulation and both of them try to avoid a collision using the velocity obstacle approach, the agents' movements start oscillating between moving in the desired direction and dodging each other. The reason is that both of the agents try to avoid each other, which leads to an undesirable feedback.

A method that deals with this issue is the reciprocal velocity obstacle approach [24]. The reciprocal velocity obstacle approach takes the collision-avoiding behavior of the other agents into account by assuming that the other agents use the same method to avoid collisions. In other words if two agents are about to collide, then both of them take a share of responsibility to avoid the collision. Such a behavior seems to be realistic also for real-world situations where pedestrians perform collision avoiding movements. The reciprocal velocity obstacle approach calculates for every agent at every time step a velocity obstacle. If a velocity lies inside the obstacle, then a collision would occur at some future time. Again, under the assumption of constant movement of all agents. The authors note that there may be situations where a collision-free velocity is not possible. Therefore they propose a penalty approach that calculates a penalty based on divergence from the desired velocity and the time to the next collision for a set of candidate velocities. The penalty calculation for an agent i at time t and velocity \mathbf{v}'_i takes the following form:

$$penalty_i(\mathbf{v}'_i(t)) = w_i \frac{1}{tc_i(\mathbf{v}'_i(t))} + \|\mathbf{v}_i^0(t) - \mathbf{v}'_i(t)\|. \quad (5)$$

In the simulation the agents then choose the intended velocity $\mathbf{v}_i(t)$ out of a set of candidate velocities so that \mathbf{v}_i minimizes the penalty.

$$\mathbf{v}_i = \arg \min_{\mathbf{v}'_i} penalty_i(\mathbf{v}'_i) \quad (6)$$

It is obvious that the penalty approach does not guaranty a collision-free movement. This means the simulation model has to deal with collisions somehow. In this work we propose combining the reciprocal velocity approach with a social force like agent interaction model. This yields in a hybrid model that calculates a force based on velocities that are chosen according to the reciprocal velocity approach and deals with collisions by applying agent interaction forces similar to the social force model. The approach ends up with a single force that works

on the agent in question.

$$\mathbf{f}_i(t) = \frac{m_i}{\tau}(\mathbf{v}_i^0(t) - \mathbf{v}_i(t)) + \sum_{j \neq i} \mathbf{f}_{ij}(t), \quad (7)$$

where $\mathbf{v}_i(t)$ is chosen according to Eq. 6 and the agent interaction force is $\mathbf{f}_{ij}(t)$:

$$\mathbf{f}_{ij}(t) = g(A_i \exp((r_{ij} - d_{ij})/B_i) \mathbf{n}_{ij}), \quad (8)$$

with all operators, constants, and variables defined in the same way as they are defined in Eq. 3.

2.4 Agent navigation and desired velocity determination

In the current setup every agent starts at a link in the navigation graph. A simple approach would be to let \mathbf{v}_i^0 point towards the to-node of the start link at the beginning. As soon as the node is reached, \mathbf{v}_i^0 points to the to-node of the next link and so on until the agent reaches her destination. However, in [7] it has been shown that such an approach leads to an unrealistic behavior in situations when agents are moving next to each other. The reason is that those agents are pulled together at close range to a node, and after the node is passed their trajectories diverge again. The authors proposed a force system that follows the route in the navigation graph. The basic idea is that each agent keeps a shadow tag on the navigation graph, which moves along the graph as the agents move forward. Furthermore, the agents are connected by a virtual rubber strap to their corresponding shadow tag. The agents are driven by a driving force that works parallel to the link which has the shadow tag on it. The virtual rubber strap pulls the agents towards the shadow tag if they diverge too much from the link. This leads to the following force:

$$\mathbf{f}_i^{move}(t) = \frac{m_i}{\tau}(\mathbf{v}_i^0(t) - \mathbf{v}_i(t)) + \mathbf{A}_{path} \mathbf{e}^{d_i^p(t)/B_{path}} \mathbf{d}_i^p(t), \quad (9)$$

where $d_i^p(t)$ is the agent i 's distance to the current link, and $\mathbf{d}_i^p(t)$ is the perpendicular unit vector pointing from the agent to the current link.

For the social force model and the collision-predicting force model this approach is straightforward to implement. One only has to replace the term $\frac{m_i}{\tau}(\mathbf{v}_i^0(t) - \mathbf{v}_i(t))$ in Eq. 2 by \mathbf{f}_i^{move} . However, for the reciprocal velocity obstacle model the integration is more elaborate. We propose to derive the desired velocity $\mathbf{v}_i^0(t)$ for the penalty calculation in Eq. 5 from \mathbf{f}_i^{move} . This is achieved by implementing a “simulated” acceleration of agent i by \mathbf{f}_i^{move}/m_i for one simulation step and taking the resulting velocity as the intended velocity $\mathbf{v}_i^0(t)$ for Eq. 5. Details for the calculation of accelerations and velocity updates are given in the next section.

2.5 Integration into MATSim

The models discussed so far calculate forces or velocities as a function of time in a continuous time model. However, in order to integrate the model into a

computer simulation the time has to be discretized. This means that an agent’s velocity will only be updated at discrete time steps. There are different ways how to update velocities at discrete time steps. A commonly used way is the Euler–Cromer method (see, e.g., [2]), which is based on the following equations:

$$\mathbf{a}_i(t) = \frac{\mathbf{f}_i(t)}{m_i}, \quad (10)$$

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + h\mathbf{a}_i(t), \quad (11)$$

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{v}_i(t+h). \quad (12)$$

Here, $\mathbf{r}_i(t)$ is the agent i ’s position at time t and h is the time step size.

3 Experiments

In 2010, a series of human crowd experiments were conducted at Technische Universität Berlin, with the main goal of evaluating simulation models of intersecting pedestrian flows. In the experiment which we use here, the participants have been divided into two groups (group A , 142 subjects, and group B , 83 subjects), and each of the groups was instructed to walk along a given path. These paths were arranged such that the two pedestrian groups intersected at an angle of about 90 degrees. The scene was recorded with three networked and temporally synchronized video cameras. The individual trajectories captured by each camera have been extracted afterwards from the video data by photogrammetric means. Afterwards, the paths of the pedestrians captured by different cameras were then merged by matching trajectories of minimal spatio-temporal distance. The latter procedure was facilitated by the Kuhn–Munkres algorithm, also known as the Hungarian method [15, 18]. For details, we refer to [21, 22]. See Fig. 1 for a snapshot of the experiment.

In Section 4, we evaluate the performance of the force-based models and the reciprocal velocity obstacle model by comparison with the real-world data in terms of the pedestrian density and flow. In order to compute these data, we first calculate a dynamic local density and flow field from the trajectories via kernel density estimation similar to [12] but with variable kernel bandwidth – for details on the method, see [20].

In order to test the simulation models discussed in the previous section. A digital model of the experiment area has been implemented and for each person of group A and group B an artificial agent is created. The simulations are performed on a laptop computer with a 2.66 GHz dual core CPU. The results of the simulation models compared with experiment or given in Fig. 3, 4, and 5 respectively. A discussion on the results is given in the next section. Another important issue is the runtime performance of the simulation models. Our implementation of the social force model needs about 3 seconds to simulate the experiment, the collision-predicting force model needs 16 seconds, and the reciprocal velocity obstacle model needs 5 seconds. The simulation times are reasonable for the social force model and the reciprocal velocity obstacle model. The collision-predicting force model, however, could become too slow for larger scenarios.

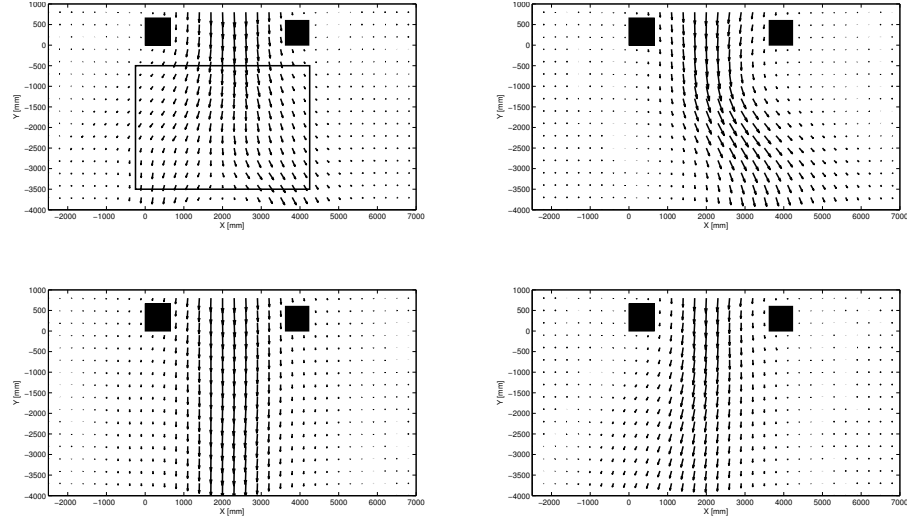


Fig. 2. Temporally averaged pedestrian flow field of group *B*. Upper left: experiment, upper right: Helbing simulation [9], lower left: van den Berg [24], lower right: Zanlungo [25].

4 Discussion

There are two general observations in the results. First, all of the models deliver usable results in terms of travel time. This is best shown in Fig. 3. On average the last pedestrian of either groups leaves the scene after about 110 s. The trends of the density curves are also similar even though none of the models reach such high densities as observed in the experiment. The reason for this lies in the nature of the simulation models which leads to the second observation. The simulation models can be divided into two classes. On the one hand there is the social force model, where the pedestrians act like colliding rubber balls. This behavior lead to a strong deviation from the desired movement direction (see Fig. 5). On the other hand there are the collision-predicting force model and the reciprocal velocity obstacle model. In both models the agents' strategy to avoid collisions is more efficient than can be observed in reality. Thus, less congestion is observed. This is indicated by lower densities compared to the experiment and the social force model (see Fig. 3). One reason for the over-optimization could be caused by the fact that the simulation models do not explicitly distinguish between pedestrians who are moving in the same direction and those who are intersecting the desired path. Furthermore, when the movement for a pedestrian is calculated, all other pedestrians are considered as individual obstacles and not as groups of obstacles. The hypothesis would be that humans make this distinction and that they try to avoid penetrating groups of persons that apparently belong together. If this hypothesis is true, then the simulation models should

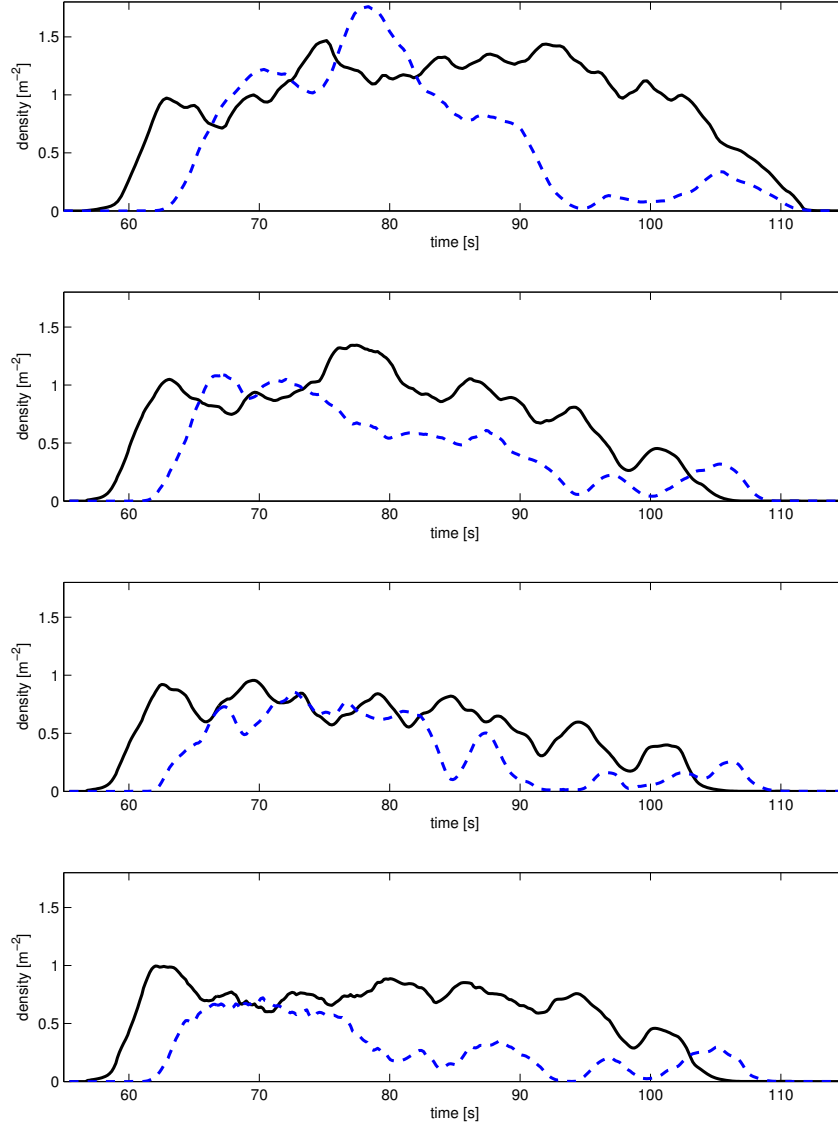


Fig. 3. Pedestrian density vs. time, averaged across the region marked in Figs. 1 and 2. Black line: group A, dashed blue line: group B. From top to bottom: experiment; simulations: Helbing [9], van den Berg [24], Zanlungo [25].

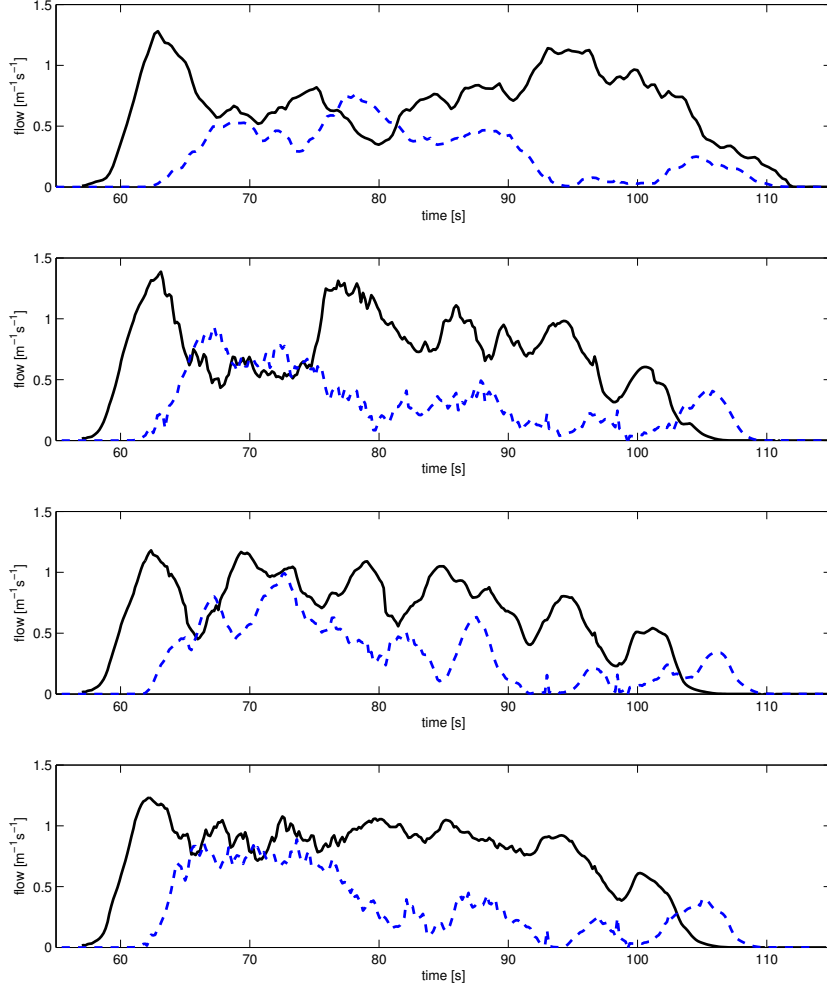


Fig. 4. Main flow components vs. time, averaged across the region marked in Figs. 1 and 2. Black line: flow of group A in x -direction, dashed blue line: flow of group B in $-y$ -direction. From top to bottom: experiment; simulations: Helbing [9], van den Berg [24], Zanlungo [25].

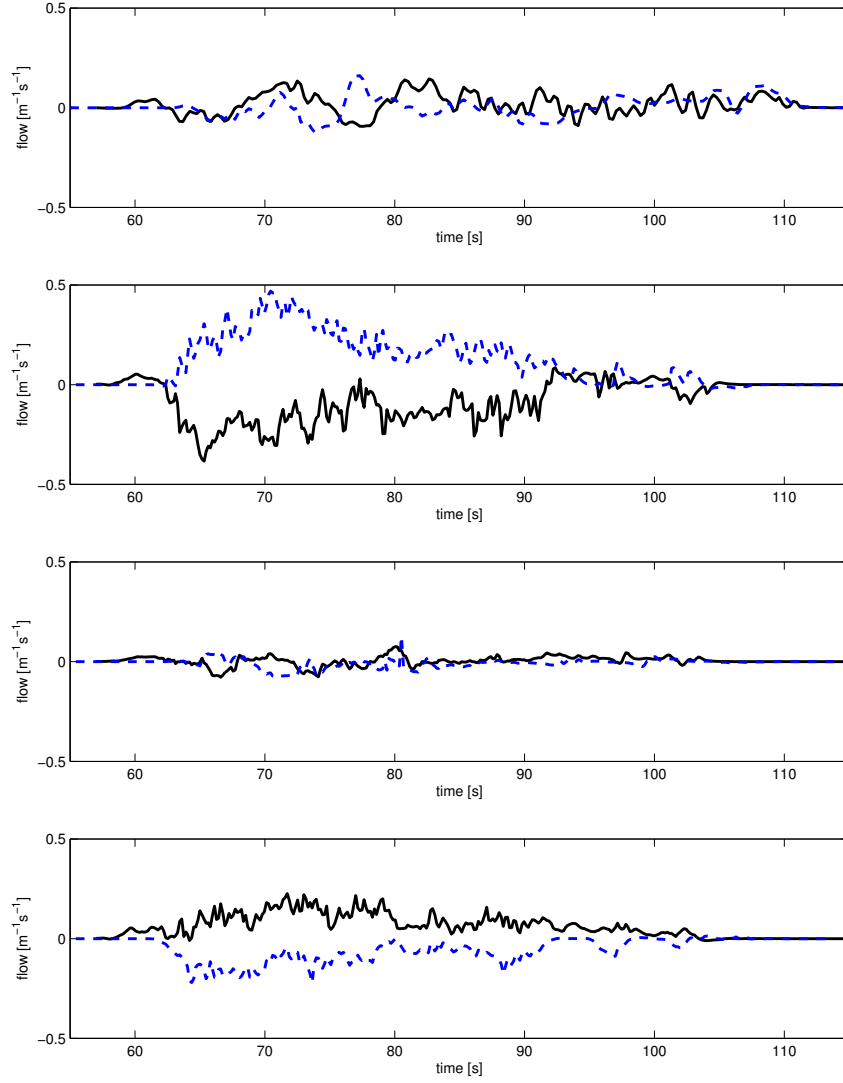


Fig. 5. Secondary flow components vs. time, averaged across the region marked in Figs. 1 and 2. Black line: flow of group *A* in *y*-direction, dashed blue line: flow of group *B* in *x*-direction. From top to bottom: experiment; simulations: Helbing [9], van den Berg [24], Zanlungo [25].

be extended accordingly. However, there are further investigations necessary to make a conclusive statement about this hypothesis.

5 Conclusion and Outlook

This contribution discusses three different approaches for the simulation of crossing pedestrian streams. The probably best-known simulation model in the pedestrian context is the social force model. The social force model does not consider collisions explicitly. The second model is an extension to the social force model, which does consider collision explicitly and calculates social forces accordingly. The third model, the reciprocal velocity obstacle model, lets agents avoid collisions by calculating admissible velocities close to the desired one. While none of the models is novel, several adaptations were necessary to integrate them into a multi-agent simulation, where the agents can choose paths to arbitrary destinations. Furthermore, the reciprocal velocity obstacle model has been extended to deal with collisions similar to the social force model, if those occur. The results are plausible even so improvements are possible. In future work it is planned to test the models on other pedestrian experiments. One focus could be the test of the group hypothesis stated in the previous section. If the hypothesis is true, then the development of a simulation model that takes groups of pedestrians into account when calculating collision-avoiding movements could also be the topic of future research.

Acknowledgments

This project was funded in part by the German Ministry for Education and Research (BMBF) under grant 13N11382 (“GRIPS”) and by the German Research Foundation (DFG) under grants NA682/5-1, SCHW548/5-1 and BA1189/4-1.

References

1. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, M.: Computational Geometry. Springer (2000)
2. Cheney, W., Kincaid, D.: Numerical Mathematics and Computing. Thomson, fifth edn. (2004)
3. Chraïbi, M., Seyfried, A., Schadschneider, A.: Generalized centrifugal-force model for pedestrian dynamics. *Phys. Rev. E* 82, 046111 (Oct 2010), <http://link.aps.org/doi/10.1103/PhysRevE.82.046111>
4. Dobler, C., Kowald, M., Schüssler, N., Axhausen, K.: Within-day replanning of exceptional events. *TRR* (2012)
5. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research* 17(7), 760–772 (1998)
6. Gawron, C.: An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C* 9(3), 393–407 (1998)

7. Gloor, C., Mauron, L., Nagel, K.: A pedestrian simulation for hiking in the alps. In: Proceedings of the Swiss Transport Research Conference (STRC). Monte Verita, CH (2003), see www.strc.ch
8. Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-organized pedestrian crowd dynamics: experiments, simulations and design solutions. *Transportation Science* 39, 1–24 (2005)
9. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. *Nature* 407, 487–490 (2000)
10. Helbing, D., Farkas, I., Molnar, P., Vicsek, T.: Simulation of pedestrian crowds in normal and evacuation situations. In: Schreckenberg, M., Sharma, S.D. (eds.) *Pedestrian and Evacuation Dynamics*, pp. 21–58. Proceedings of the 1st international conference, Duisburg, 2001, Springer (2002)
11. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Phys. Rev. E* 51(5), 4282–4286 (1995)
12. Helbing, D., Johansson, A., Al-Abideen, H.Z.: Dynamics of crowd disasters: An empirical study. *Phys. Rev. E* 75, 046109 (2007)
13. Johansson, A., Helbing, D., Shukla, P.: Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems* 10, 271–288 (2007)
14. Klüpfel, H., Meyer-König, T., Keßel, A., Schreckenberg, M.: Simulating evacuation processes and comparison to empirical results. In: Fukui et al, M. (ed.) *Traffic and granular flow '01*, pp. 449–454. Springer, Berlin Heidelberg New York (2003)
15. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly* 2, 83–97 (1955)
16. Lämmel, G.: Escaping the Tsunami: Evacuation Strategies for Large Urban Areas. Concepts and Implementation of a Multi-Agent Based Approach. Ph.D. thesis, TU Berlin (2011), <http://opus.kobv.de/tuberlin/volltexte/2011/3270/>
17. Lämmel, G., Grether, D., Nagel, K.: The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations. *Transportation Research Part C: Emerging Technologies* 18(1), 84–98 (2010)
18. Munkres, J.: Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5(1), 32–38 (1957)
19. Oleson, R., Kaup, D., Clark, T., Malone, L., Boloni, L.: Social potential models for traffic and transportation. In: Bazzan, A., Klügl, F. (eds.) *Multi-agent systems for traffic and transportation engineering*, chap. VII, pp. 155–175. Information Science Reference (IGI Global) (2008)
20. Plaue, M., Bärwolff, G., Schwandt, H.: On measuring pedestrian density and flow fields in dense as well as sparse crowds (2012), to appear in *Proc. PED2012*
21. Plaue, M., Chen, M., Bärwolff, G., Schwandt, H.: Multi-view extraction of dynamic pedestrian density fields (2012), preprint
22. Plaue, M., Chen, M., Bärwolff, G., Schwandt, H.: Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings. In: *Proc. PIA11. LNCS*, vol. 6952, pp. 285–296 (2011)
23. Raney, B., Nagel, K.: An improved framework for large-scale multi-agent simulations of travel behaviour. In: Rietveld, P., Jourquin, B., Westin, K. (eds.) *Towards better performing European Transportation Systems*. Routledge, London (2006)
24. van den Berg, J., Lin, M.C., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: *Proc. IEEE Intl. Conf. on Robotics and Automation*. pp. 1928–1935 (2008)
25. Zanolungo, F., Ikeda, T., Kanda, T.: Social force model with explicit collision prediction. *EPL (Europhysics Letters)* 93(6), 68005 (2011)