

# Efficient Dynamic Floor Field Methods for Microscopic Pedestrian Crowd Simulations

Dirk Hartmann<sup>1,\*</sup> and Peter Hasel<sup>1</sup>

<sup>1</sup> Siemens AG, Corporate Technology, 80200 Munich, Germany.

Received 20 May 2013; Accepted (in revised version) 29 January 2014

Communicated by Chi-Wang Shu

Available online 16 April 2014

---

**Abstract.** Floor field methods are one of the most popular medium-scale navigation concepts in microscopic pedestrian simulators. Recently introduced dynamic floor field methods have significantly increased the realism of such simulations, i.e. agreement of spatio-temporal patterns of pedestrian densities in simulations with real world observations. These methods update floor fields continuously taking other pedestrians into account. This implies that computational times are mainly determined by the calculation of floor fields. In this work, we propose a new computational approach for the construction of dynamic floor fields. The approach is based on the one hand on adaptive grid concepts and on the other hand on a directed calculation of floor fields, i.e. the calculation is restricted to the domain of interest. Combining both techniques the computational complexity can be reduced by a factor of 10 as demonstrated by several realistic scenarios. Thus on-line simulations, a requirement of many applications, are possible for moderate realistic scenarios.

**AMS subject classifications:** 52B10, 65D18, 68U05, 68U07

**Key words:** Pedestrian dynamics, cellular automata, adaptive grids, floor field methods, fast marching method, navigation.

---

## 1 Introduction

The study of pedestrian crowd dynamics is an emerging field in complexity science [18, 35, 36, 42, 52]. Theoretical insights into crowd behaviour are valuable sources for improving such diverse fields as the operation of large buildings and infrastructures, the optimization of passenger exchange times, or the optimization of egress strategies [52].

---

\*Corresponding author. *Email addresses:* hartmann.dirk@siemens.com (D. Hartmann), peter.hasel@googlemail.com (P. Hasel)

With respect to these applications the realism of the underlying models is of crucial importance, since it ultimately determines the utility and reliability of computational predictions [44]. For a broad overview on the topic of pedestrian crowd simulations we refer e.g. to [2, 3, 23, 30, 46, 52].

Theoretical approaches to pedestrian crowds either adopt a density based description (e.g. [7, 10, 12, 13, 22, 25, 27–30, 58, 59]), i.e. adopt a continuum macroscopic point of view, or explicitly resolve the behaviours of single pedestrians (e.g. [24, 34]), i.e. adopt an individual point of view. Within this work we will restrict ourselves to individual-based models, often referred as microscopic models. Very recently also a hybrid multi-scale approach combining the two modelling concepts has been proposed [10].

The most popular microscopic approaches are continuum social force field models going back to [24], e.g. [8, 9, 37, 41, 45, 63], and discrete cellular automata models, e.g. [4, 5, 16, 21, 26, 32, 34, 43, 52, 56, 60]. One of the central (computational) challenges of microscopic approaches is the navigation of single pedestrians through complex topologies (for a review of navigation strategies in microscopic pedestrian crowd models we refer e.g. to [32, 46, 52]). Navigation of pedestrians typically addresses different spatial scales: long range, medium range and small range aspects. Long range navigation considers more strategic aspects, e.g. navigation using maps, floor plans or signage rather than direct visual information, and can be modelled quite well by a sequence of intermediate destinations. The typical scale is  $10 \sim 200\text{m}$ . Usually graph based approaches are used to model long range navigation decisions [33] (and references therein) or even to optimize long range routing [17, 19]. Medium scale navigation addresses the navigation from one intermediate destination to the next along the shortest or fastest path, e.g. from one graph node to the next. Thus medium scale navigation is based mostly on visibility, i.e. typical scales are  $5 \sim 50\text{m}$ . The most popular approach for medium scale navigation is floor field based navigation. Floor fields are scalar continuum fields, usually of a static nature, and are defined independently of pedestrians present in the scenario, e.g. [4, 5, 32, 34, 52, 56, 60]. Interactions of pedestrian is typically modelled via dynamic short range repulsion (focusing on distances up to  $1 \sim 10\text{m}$ ). In contrast to the long range part, this short range interaction is highly dynamic. It is typically modelled using social force models (or corresponding potentials in cellular automaton models)

A common criticism [36] is that these classical navigation strategies do not mimic natural movement behaviour of humans. Most humans take other pedestrians into account once they are visible (independently how far these are away) and not only once they are close enough, i.e. in the short range of  $1 \sim 10\text{m}$ . That is, other pedestrians are included on short as well as medium scale navigation decisions rather than only in short scale navigation decision. This difference typically leads to artefacts in microscopic simulations based on short range interactions of pedestrians (cf. Fig. 1(a)).

Recently, [21] and [36] have proposed independently a dynamic medium range navigation strategy in the context of floor field methods in cellular automata based simulations. (A generalization of these methods to other microscopic pedestrian simulations is possible.) Both works are inspired by concepts for pedestrian navigation originally intro-

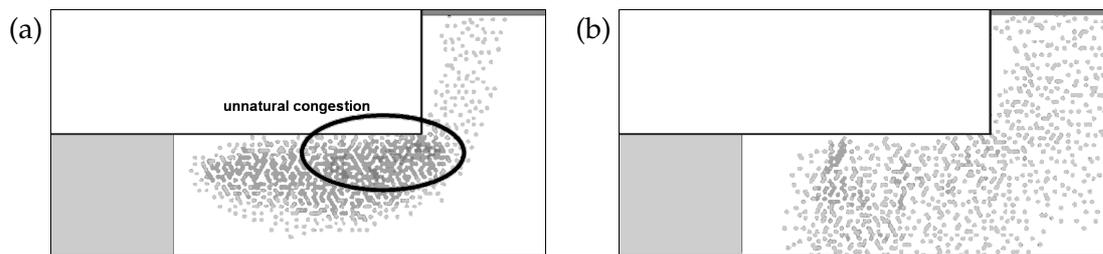


Figure 1: Pedestrians travelling around a corner (from bottom left to top right) – Following shortest paths simulations might yield insufficient results if pedestrian densities are not considered during the calculation of the floor field (a), i.e. unnatural congestions could emerge. Taking the densities into account (dynamic floor field methods) more realistic patterns are recovered (b)

duced earlier in continuum models, e.g [27–29,58,59]. Combining density based models for navigation with microscopic models for pedestrian movement, hybrid models for microscopic pedestrian dynamics are obtained. As a consequence, both approaches take other pedestrians already within medium range navigation decisions into account. The dichotomy of local versus non-local interactions is resolved. Thus the two approaches yield significant more realistic results in terms spatio-temporal patterns of pedestrian densities (cf. Fig. 1(b)). Within this work, we refer to a high degree of realism if spatio-temporal patterns of densities – not tracks of single pedestrians – show a good agreement with patterns observed in everyday life. Unfortunately, the availability of corresponding data is very limited such that the notion of realism is at the moment somewhat arbitrary. As already stressed above, the degree of reality of the movement patterns obtained by pedestrian simulators is of crucial importance. Evacuation times are only reliable and robust, if unrealistic congestions can be ruled out.

Although dynamic floor field methods explicitly considering dynamics of other pedestrians show more realistic results, their application is limited so far to academic scenarios due to their computational complexity. At each time step of the simulation a floor field is determined for the whole domain of interest. Thus the overall computational complexity of the microscopic pedestrian simulation is mainly determined by the computation of floor fields. The computational complexity might be further reduced by recalculating the navigation field not every time step but rather every few time steps. This however might lead to oscillating spatio-temporal patterns with a periodicity of the times between updates (cf. Appendix B), i.e. realism would be sacrificed.

In this article, we introduce new computational concepts improving significantly the computational efficiency of dynamic (and also static) floor field methods. Here, we rely on the relative simple cellular automaton outlined in [34], which has been realized in Java. Furthermore, we will restrict us to simple origin destination relations, i.e. typically one origin and one destination, for the sake of clarity. Naturally, the approach can be extended to most other microscopic approaches, e.g. modelling more complex sociological behaviour.

The novel contributions of our work are:

- adaptive calculation of floor fields for navigation (Section 3);
- directed calculation of floor fields reducing the region in which floor fields have to be calculated to a minimum (Section 4);
- inertial floor field methods allowing to reduce the frequency which floor fields have to be updated (Appendix B).

The combination of the three concepts leads to a significant reduction of computational times (factor  $10 \sim 50$ ). Thus realistic scenarios can be computed with acceptable (from an end-user point of view [44]) computational effort. In difference to purely mathematical oriented approaches, we do not aim at improving computational speed-up without impacting the convergence of the macro-scale evolution of pedestrian densities, but rather aim at computational efficiency without loosing realism of the microscopic pedestrian simulations. It is important to note that the underlying mathematical model can only formalize the assumed crowd navigation logic to a certain degree. In contrast to most physical phenomena, the modelling of social phenomena is very likely to be subject to non-negligible modelling errors. Therefore, we accept sacrifice of formal convergence as long as the results are sufficiently accurate for the simulation purposes, the prediction of crowd evolution.

The article is structured as follows: In Section 2 we review the concept of static and dynamic floor field methods, i.e. field based navigation. Then we show how to combine floor field methods with the concept of grid adaptivity in Section 3. Together with a directed calculation (Section 4), i.e. a restriction of the calculation to minimal domains, a performance increase by a factor 10 can be achieved. This increase in computational performance is shown in Section 5 simulating pedestrian flows on the Marienplatz in Munich. Furthermore, we shortly revisit the cellular automaton model of [34] in Appendix A. In Appendix B we introduce the idea of inertial floor fields, allowing a significant decrease of the frequency navigation fields have to be updated, i.e. recalculated. This leads to a reduction of computational efforts in combination with the other two methods by a factor  $\sim 50$  in total.

## 2 Floor field methods

Let us shortly revisit the concept of floor field based navigation. To do so, we consider a single destination to which pedestrians are moving towards within the domain of interest  $\Omega$ . The main idea of floor field methods is to construct for all  $\vec{x} \in \Omega$  a continuous scalar field  $F(\vec{x}, t)$  – the floor field – which takes a minimum at the destination, given by the curve  $\Gamma \subset \Omega$ , and which is strictly increasing moving away from the destination. This floor field could depend on  $t$ , but typically it is considered to be static. If no other pedestrians are present, the navigation strategy of a single pedestrian reduces to moving down the

gradient of the scalar field  $F(\vec{x})$ . The influence of other pedestrians is usually modelled via additional local interaction potentials. Thus in general pedestrians move down the gradient of the summed potentials (for more details c.f Appendix A).

Here, we follow the approach of [21]: The main idea is to construct navigation fields in the context of microscopic pedestrian simulations by solving the Eikonal equation using the fast marching method (FMM) [54], i.e. the navigation field and thus fastest/shortest paths are determined by calculating arrival times of waves expanding from the destination. In contrast to most other microscopic approaches (considering static floor fields), e.g. [4, 5, 32, 34, 43, 52, 56, 60], this method adopts a dynamic point of view, i.e. dynamically recalculates the navigation field taking other pedestrians into account (dynamic floor field method). This leads to an increased realism in simulations. An equivalent ansatz has been independently developed by [36] and similar ideas in a continuum setting can be found e.g. in [26, 28, 58].

First, we will consider the static case and then the generalized dynamic case. The latter is inherently more complex from a computational point of view, and efficient techniques to deal with this challenge is the major focus of this contribution.

## 2.1 Static floor fields based on fast marching methods

The central idea of [21] is to construct an appropriate floor field  $F(\vec{x})$  by an estimation of travel/arrival times using the Eikonal equation:

$$V(\vec{x})|\nabla F(\vec{x})|=1 \quad \text{in } \Omega, \quad (2.1a)$$

$$F(\vec{x})=0 \quad \text{on } \Gamma, \quad (2.1b)$$

where  $F(\vec{x})$  is the arrival time of a wave originating from  $\Gamma$  in  $T=0$  and which is spreading with a normal speed  $V(\vec{x})$  in the domain  $\Omega$ . The Eikonal equation (2.1) can be solved efficiently on the dual grid of the cellular automaton, i.e. on the grid given by the graph  $\mathcal{A}_{l,\vec{x}_0}$  (cf. Appendix A and Fig. 2a) using the fast marching method (FMM), cf. [31, 55] for the FMM on triangulated/simplicial meshes and [54] for a general overview. Thus  $F(\vec{x})$  is determined only in cell centers  $\vec{x}_i \in V_{l,\vec{x}_0}^A$  and is approximated piecewise linear in between, i.e.

$$F(\vec{x}) \in \{F \in C(\Omega) : F|_T \in P_1(T) \quad \forall T \in \mathcal{T}\}, \quad (2.2)$$

where  $\mathcal{T}$  is the set of triangles spanned by the cellular automaton graph  $\mathcal{A}_{l,\vec{x}_0}$ . Assuming that pedestrians move with the same expected/estimated speed  $V$  in the whole  $\Omega$  one chooses  $V(\vec{x}) \equiv V$ . (Of course a spatial dependence is also possible, e.g. due to inclinations.) Compared to most other approaches, e.g. [4, 5, 32], using Dijkstra's algorithm [14] to determine navigation fields, i.e. navigation fields measure distances/travel times in the 1-metric (Manhattan metric), the approach of [21] shows significantly more realistic results.

The central idea of the FMM is to systematically construct an approximation of  $F(\vec{x})$  using only values upwind in the direction of information propagation. Similar to stan-

standard upwind discretization schemes, it is thus ensured that information is only transported from small values of  $F(\vec{x})$  to large values [54]. From an algorithmic point of view, the FMM basically reduces to iteratively finding the vertex  $\vec{x}_i$  with the smallest  $F(\vec{x}_i)$  from a set of vertices. Making use of min-heap data structures [53] an efficient implementation of the FMM with complexity  $N \log N$ , where  $N$  is the number of grid points, can be achieved. That is, the FMM can be considered as a modified heap sort algorithm sorting vertices according to the key

$$\kappa(\vec{x}_i) = F(\vec{x}_i). \quad (2.3)$$

Using standard template libraries of modern programming languages allows a fast and from a computational point of view efficient implementation of the fast marching method.

## 2.2 Dynamics floor field methods

Static floor fields neglect the influence of other pedestrians on the medium scale navigation behaviour. Most approaches model this influence separately using short range repulsion ( $1 \sim 10\text{m}$ ), i.e. interactions are considered to be local. However, human navigation typically takes other pedestrians and especially congestions on long ranges ( $5 \sim 50\text{m}$ ) into account, i.e. interactions are non-local. Thus many microscopic pedestrian simulators typically reflect unrealistic movement patterns (cf. Fig. 1(a)), i.e. spatio-temporal patterns of pedestrian densities do not agree.

To enhance the realism and hence the predictive capabilities of microscopic approaches the concept of dynamic floor fields has been introduced [21, 36], resolving the dichotomy between local and non-local interactions. The central idea of [21] (and similar of [36]) is to consider in equation (2.1) an estimated travel speed  $V(\vec{x}, t)$  depending on the local pedestrian density which comes significantly closer to real navigation, i.e. the floor field  $F(\vec{x}, t)$  depends on space and time. A typical ansatz is  $V(\vec{x}, t) = (1 + \omega \rho(\vec{x}, t))^{-1}$  with constant  $\omega$  and local pedestrian density  $\rho(\vec{x}, t)$ . Different methods can be used for an estimation of  $\rho(\vec{x}, t)$ , e.g. local averaging or following [39]. Here, we consider  $\rho(\vec{x}, t)$  to be piecewise constant on cells of the cellular automaton, with either  $\rho = 0 \text{ ped./m}^2$  if the cell is not occupied or  $\rho = 5.4 \text{ ped./m}^2$  if the cell is occupied. This long range consideration of other pedestrians within dynamic floor field methods significantly enhances the realism of microscopic pedestrian simulators (cf. Fig. 1(b)).

Static floor fields are calculated in the initialization phase of the simulation, thus computational efficiency of floor field calculation does not play a crucial role. Introducing the more realistic dynamic floor fields, floor fields have to be recalculated every time step (or at least every couple of time steps) since pedestrian distributions change in time. Thus the computational efficiency of dynamic floor field methods is the bottleneck of the overall efficiency of microscopic pedestrian simulators. Therefore, increasing their computational efficiency is one of the central challenges in microscopic pedestrian simulators.

### 3 Adaptive floor fields

In the field of scientific computing, grid adaptivity has been proven a valuable approach to reduce computational complexity and thus computational times [40]. Since the construction of floor fields basically reduces to the solution of the Eikonal equation (2.1), the application of adaptive grid refinement is a promising approach in the field of microscopic pedestrian simulators.

The grid on which the solution of the Eikonal equation is computed in the context of a cellular automaton is given by the dual grid of the cellular automaton lattice, i.e. the graph  $\mathcal{A}_{l, \vec{x}_0}$  shown in Fig. 2(a). Obstacles blocking the way of pedestrians are modelled by cutting connections, i.e. edges, from one cell to a neighbouring cell (cf. Fig. 2(b) and Appendix A). Using an adaptive grid concept coarsened grids have to be a subset of this microscopic triangulation. Otherwise obstacles would not be considered correctly. Thus, we propose to take the opposite way of classical adaptive grid strategies: instead of refining the computational grid in critical regions, we coarsen the grid in uncritical regions (cf. Fig. 2(c) and (d)).

Here, we restrict us to a static adaptation of computational grids. That is, in the initialization phase the grid is adapted and then is kept for the whole simulation. Since the solution of the Eikonal equation and thus in turn the navigation of pedestrians depends on local pedestrian densities, the ultimate goal would be a dynamic adaptive scheme, which adjusts the computational grid also according to pedestrian densities.

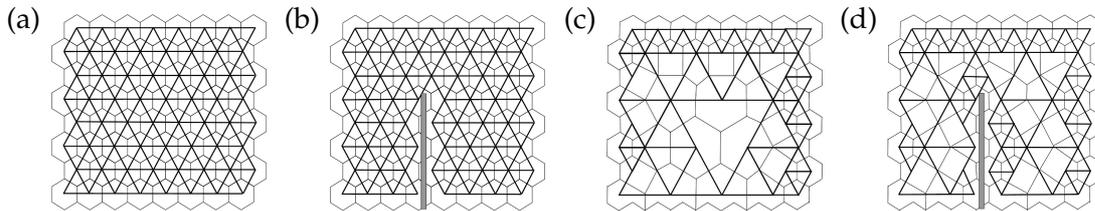


Figure 2: Grids used by the fast marching algorithms and the corresponding Voronoi diagrams to determine pedestrian densities – a non adaptive grids for a simple rectangular domain with  $\mathcal{L} = \mathcal{A}$  (a) as well as a more complex domain with  $\mathcal{L} \neq \mathcal{A}$  (b) and corresponding adaptive grids (c) and (d) are shown.

#### 3.1 Algorithm

The coarsening starts from the graph  $\mathcal{A}^0 = \mathcal{A}_{l, \vec{x}_0}$ . It is a subset of the lattice graph  $\mathcal{L}_{l, \vec{x}_0}$ . All triangles belonging to the graph will be referred to triangles of level  $n = 1$  in the following. In general, we will consider a triangle to be of level  $n$  if it belongs to the lattice graph  $\mathcal{L}_{2^n l, \vec{x}_0}$ , i.e. a triangle of level  $n+1$  consists of edges of length  $2 \cdot 2^n \cdot l$  and triangles which are composed of 4 triangles of the level  $n$ .

Let us now outline the coarsening algorithm in detail: The algorithm basically tries to coarsen the grid by forming larger triangles of level  $n+1$  from 4 triangles of level  $n$ . Here, we require that only coarse triangles are allowed which would be included in the

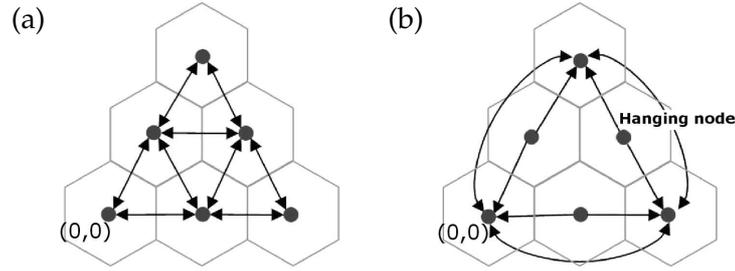


Figure 3: Schematic coarsening of 4 triangles of level  $n=0$  to one of level  $n=1$  with updated neighbourhood relations used by the FMM. Note that not all neighbourhood relations are two folded.

coarser lattice grid  $\mathcal{L}_{2^{n+1}l, \vec{x}_0}$  (cf. Fig. 3). To decide whether the triangles can be coarsened we have to distinguish two separate cases: Coarsening triangles from level  $n=0$  to  $n=1$ , one has to ensure that within the larger triangle one does neither find any obstacles, walls nor destinations. Otherwise this information would be lost. Coarsening from level  $n$  to  $n+1$  with  $n > 0$  it is sufficient to check that the new triangle of level  $n+1$  will consist of 4 triangles of level  $n$  and that neighbouring triangles are not smaller than level  $n$ . The latter condition ensures that no obtuse angles are present in the computational grid, as required by the FMM. The process of coarsening leads to a sequence of grids/graphs  $\mathcal{A}^0, \mathcal{A}^1, \dots, \mathcal{A}^n$  with vertices  $V_{2^{n+1}l, \vec{x}_0}^{\mathcal{A}^{n+1}} \subset V_{2^n l, \vec{x}_0}^{\mathcal{A}^n}$ . The coarsest grid, which by definition consists of triangles of different levels smaller or equal  $n$ , is then used to compute the corresponding floor field with the FMM.

We would like to point out, that this algorithm leads inevitably to hanging nodes, as shown in Fig. 3. Neighbourhood relations used for the FMM are not two fold any more, i.e. they are directed and a vertex must not necessarily be a neighbouring vertex of a neighbour vertex. That is the sets  $E_{2^n l, \vec{x}_0}^{\mathcal{A}^n}$  are directed. Hanging nodes (cf. Fig. 3) are assumed to have only one sided neighbour relations. Thus, neighbourhood relations have to be stored for each vertex separately. When coarsening the triangles the corresponding data structures are adjusted accordingly, i.e. neighbourhood relations for the FMM are updated. This approach implies that degenerate triangles could occur. Before updating we therefore check the validity of angles. If the angle is obtuse, we fall back to updating the node according to the distance to the neighbour with the smallest value, i.e. a quasi 1D update scheme. We would like to underline, that this makes the application of all prior formal convergence proofs inapplicable. However, our numerical results in the next section confirm that the convergence is still there for our class of examples.

In the coarsened grid not all cell centers  $\vec{x}_i$  of the cellular automaton lattice are represented any more as vertexes of the coarsened grid. Thus relation (2.2) has to be adopted accordingly, i.e.

$$F(\vec{x}) \in \{F \in C(\Omega) : F|_T \in P_1(T) \quad \forall T \in \mathcal{T}^n\}, \tag{3.1}$$

where  $\mathcal{T}^n$  is the set of triangles spanned by the cellular automaton graph  $\mathcal{A}^n$ . That is, using linear interpolation the floor field value can be determined efficiently.

Dynamic floor field methods consider other pedestrians during the construction of navigation fields. Thus we need a proper notion of pedestrian densities. Here, we adopt the following approach assuming an adaptive grid of level  $n$ . Let  $\mathcal{C}^n$  be the set of Voronoi cells of the vertices  $V_{2^n l, \vec{x}_0}^{\mathcal{A}^n}$  of the adaptive cellular automaton grid  $\mathcal{A}^n$  of level  $n$ , i.e.  $\mathcal{C}^n = \{C_1^n, C_2^n, \dots\}$  with

$$C_i^n = \{\vec{x} \in \Omega \mid d(\vec{x}, \vec{x}_i) \leq d(\vec{x}, \vec{x}_j) \quad \forall j \neq i\}, \quad (3.2)$$

with vertices  $\vec{x}_i \in V_{2^n l, \vec{x}_0}^{\mathcal{A}}$ . The pedestrian density  $\rho(\vec{x}, t)$  is considered to be a piecewise constant function on the set of Voronoi cells, i.e.

$$\rho(\vec{x}, t) \in \{BV(\Omega) : \rho|_C \in P_0(C) \quad \forall C \in \mathcal{C}^n\}. \quad (3.3)$$

The natural approach is then to choose  $\rho(\vec{x}, t)$  such that in each cell  $C_i^n$  it corresponds to the average density of the cell, i.e.  $\rho_i = \# \text{ ped. in } C_i^n / \|C_i^n\|$ . As before, we choose  $V(\vec{x}_i) = (1 + \omega \rho(\vec{x}_i))^{-1}$  in (2.1) with constant  $\omega$ . Adopting this approach, on the finest level  $n=0$  this definition agrees with the definition given in Section 2.

Summarising, the ansatz to calculate navigation fields with the FMM using an adaptive grid needs only minor modifications compared to algorithms relying on non-adaptive grids. The main difference is the introduction of one sided neighbourhood relations. At the same time, even considering relatively coarse grids results differ only slightly. Interaction of pedestrians on the small scale is still covered by local short scale repulsion typically considered in most microscopic models. However, without the additional short range repulsion, the spatial average of larger Voronoi cells leads to unrealistic traces of pedestrians around other pedestrians, i.e. an unrealistic short range navigation.

### 3.2 Examples

First of all, let us consider the most simple scenario, a square domain with a single pedestrian travelling from the bottom to the top. Since we consider a single pedestrian a recalculation of the navigation field is not necessary, i.e. we consider a static floor field ansatz.

In the first step, we have verified the convergence of the scheme (cf. Fig. 4). As expected the error is decreasing when decreasing the discretization length scale. However, we would like to stress that the length scale of discretization is determined by the spatial discretization of the cellular automaton underlying the pedestrian simulation. This in turn is restricted by the maximum density of pedestrians per square meter. According to [57] it is therefore bounded from below by  $l = 0.46\text{m}$ . Additionally, we have investigated how the computational complexity, i.e. the computation time, scales with the size of the domain. The result is shown in Fig. 5. Although the adaptive fast marching method (aFMM) is significantly faster than the classical FMM, the complete performance of the aFMM is worse, since one also has to construct an adaptive grid before computing the floor field using the aFMM (Both steps are shown separately in Fig. 5). The construction

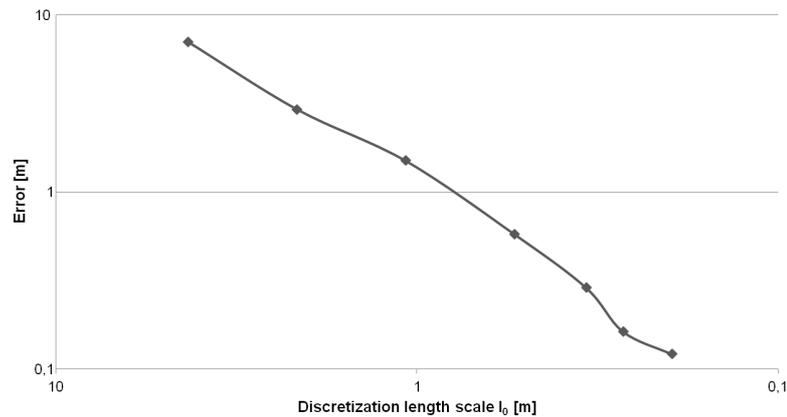


Figure 4: Absolute error of the FMM of a path around an obstacle (65.6m) for different discretization length scales  $l_0$ .

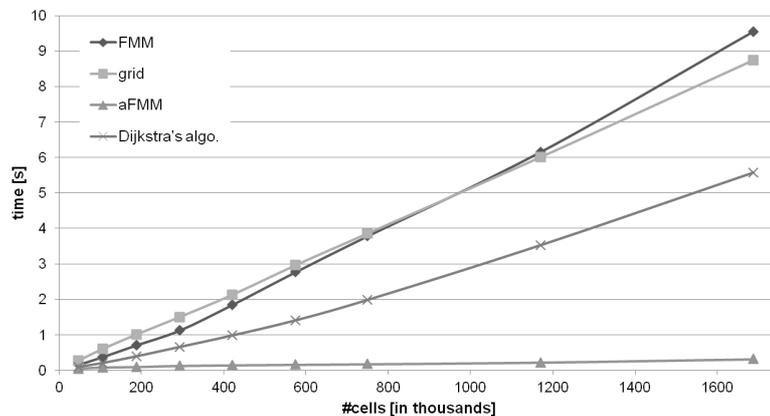


Figure 5: Scaling of computational complexity with increasing size of a simple square domain (from  $100\text{m} \times 100\text{m}$  to  $600\text{m} \times 600\text{m}$  and destination at the bottom). Computation times for the calculation of a single static floor field using different algorithms (leading to navigation fields of different quality) are shown.

of an adaptive grid is as complex as the solution of the original FMM. However, considering scenarios where a dynamic recalculation of the floor field is necessary, cf. Section 2.2, the aFMM outperforms the classical FMM since the adaptive grid has to be constructed only once. Furthermore, also the classical Dijkstra's algorithm is outperformed in the case of a dynamically recalculated floor field as shown in Fig. 5. We would like to emphasize that the different algorithms lead to navigation fields of different quality. Thus the comparison considers solely computational efforts disregarding the quality of the navigation field. Furthermore, we would like to point out, that a comparison with [6] shows a significant difference in computational efficiency. Thus the computational efficiency is very likely to be improved switching to a different programming language than Java.

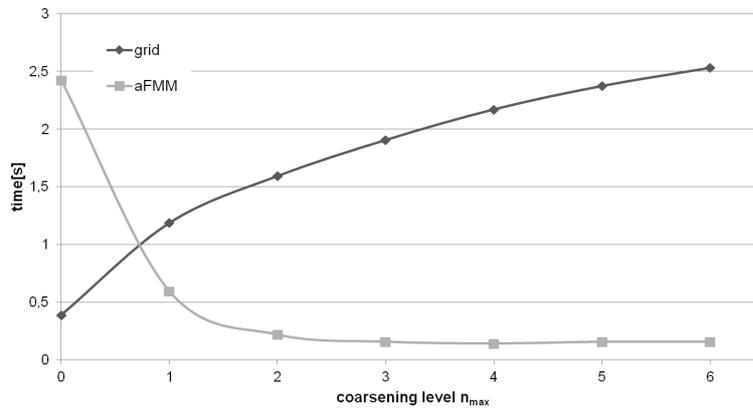


Figure 6: Scaling of computational complexity with increasing level  $n_{\max}$  of coarsening for a  $300\text{m} \times 300\text{m}$  domain (destination at the bottom). Computation times for the calculation of a single static floor field are shown.

In the next step, we have fixed the size of the domain to  $300\text{m} \times 300\text{m}$  and furthermore limited the maximal coarsening level allowed to  $n_{\max}$ . Results for different levels of  $n_{\max}$  are shown in Fig. 6. One observes that the gain of computational complexity beyond a coarsening level of  $n_{\max} = 3$  is rather negligible.

As a third example, we consider a pedestrian crowd travelling around a corner – one of the most fundamental scenarios requiring a continuous recalculation of floor fields for navigation. In Fig. 7 the corresponding example is shown for different levels  $n_{\max}$  of coarsening. The algorithm shows promising results up to level  $n_{\max} = 3$ , i.e. pedestrian

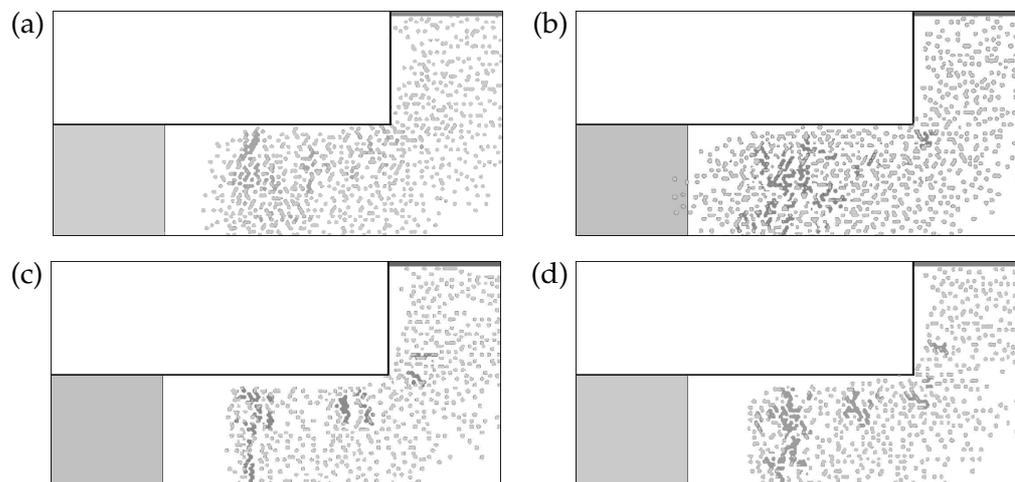


Figure 7: Pedestrians travelling around a corner (from bottom left to the top right) – results of different levels of adaptivity:  $n_{\max} = 0$  (a),  $n_{\max} = 1$  (b),  $n_{\max} = 2$  (c),  $n_{\max} = 3$  (d).

streams show the expected behaviour and do not differ qualitatively from simulations using non-adaptive grids. Thus the accuracy is sufficient with respect to our requirements. For larger  $n_{\max}$  (results not shown) the pedestrian patterns show less realistic results since a detailed and relative exact representation of pedestrian densities is not possible. For small  $n_{\max}$  the local repulsion ensures a realistic short range path planning. However, for larger levels the short range planning is insufficient to cope with the lower accuracy of the approximation introduced by grid coarsening ultimately leading to unrealistic patterns. Introducing dynamic adaptive schemes which also dynamically adjust the scheme according to local pedestrian densities would likely resolve this issue. However, since for levels  $n_{\max} > 3$  the speed-up using an adaptive FMM compared to  $n_{\max} = 3$  is rather small, a restriction to  $n_{\max} = 3$  is acceptable.

## 4 Directed floor fields

Following the classical FMM a floor field, respectively navigation field, is determined in the whole domain  $\Omega$ . However, in many situations pedestrian paths cover only a small portion of the domain of interest – which unfortunately does not have to be known in advance. Large areas are not entered by pedestrians and thus a floor field is calculated in vain.

In the context of path planning the directed calculation of navigation fields similar to the concept of the  $A^*$  algorithm [20] has been well established in the last years, e.g. [11,15,47–51,61,62]. The central idea is to sort cells  $\vec{x}_i$  in the FMM algorithm not according to a key given by the field value  $F(\vec{x}_i)$  but rather according to a key being a combination of  $F(\vec{x}_i)$  and the direct distance to the current position of the underwater vehicle  $\vec{p}$ , i.e. instead of (2.3) the following key is chosen

$$\kappa(\vec{x}_i, p) = \alpha F(\vec{x}_i) + (1 - \alpha)\beta d_{\text{direct}}(\vec{x}_i, \vec{p}). \quad (4.1)$$

Here,  $0 \leq \alpha \leq 1$  and  $\beta > 0$  are constants determining the detailed weight. In the case  $\alpha = 1$  the original FMM is recovered. Choosing  $\alpha < 1$ , vertices being closer to the underwater vehicle are preferred and thus the floor field is calculated with a preference in the direction of the underwater vehicle.

In the presence of many obstacles, it is however questionable whether the use of direct distances not considering obstacles leads to minimal areas, where the navigation field is calculated, as well as realistic pedestrian flows (cf. Fig. 8). Therefore, we suggest to use the exact distance  $d_{\text{exact}}$  of the shortest path around obstacles and not the direct distance  $d_{\text{direct}}$  as in (4.1). The corresponding distance field  $d_{\text{exact}}$  can be calculated efficiently using again the FMM evolving now from the origin (cf. Fig. 9(b)). Since the corresponding distance field has to be calculated in the whole domain, efficiency is only gained if the floor field used for navigation is computed sufficiently often ( $d_{\text{exact}}$  needs to be evaluated only once).

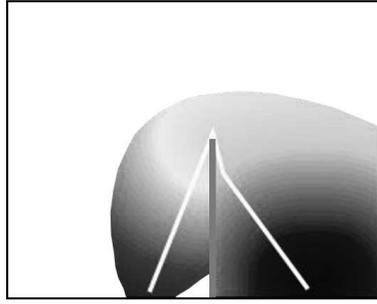


Figure 8: Directed floor fields for a simple scenario with a origin (lower left), destination (lower right) and an obstacle/wall in between using the approach of [48], i.e. the key (4.1) considering direct distances (not considering walls) to the origin. Here, we have chosen  $\alpha=0.6$  and  $\beta=1.0$ . The path taken by pedestrians is exemplary shown by a white line.

#### 4.1 Algorithm

Let us assume a single origin (at position  $\vec{s}$ ) and a single destination  $\Gamma$  for the moment. Instead of using the distance to the current position of pedestrians in (4.1) one uses the exact distance from the current cell  $\vec{x}_i$  to the origin  $\vec{s}$ , i.e.  $d_{\text{exact}}(\vec{x}_i, \vec{s})$  (which is calculated using the FMM evolving from the origin and needs to be evaluated only once since origins are fixed). Thus the floor field is calculated using a directed FMM evolving from the destination with the following key

$$\kappa(\vec{x}_i, s) = \alpha F(\vec{x}_i) + (1 - \alpha) \beta d_{\text{exact}}(\vec{x}_i, \vec{s}). \quad (4.2)$$

Furthermore, we adopt the following “lazy” strategy to determine navigation fields within microscopic pedestrian simulations: In the initialization phase of the simulation, we calculate the distance field  $d_{\text{exact}}(\vec{x}_i, \vec{s})$  using a classical FMM. However, with respect to the navigation Field  $F(\vec{x}, t)$  we set up only the initial heap structure for the FMM\*, i.e. “the seeds”. Once a navigation field value for a cell, which has not been calculated so far, is requested by a pedestrian, the FMM starts/continues the calculation of the navigation floor field until a value for the corresponding cell has been determined. This “lazy” strategy ensures that the number of cells for which a floor field is calculated is minimal, i.e. the computational effort spend is as small as possible.

We would like to emphasize that the A\*-style modifications of FMM adopted here produce additional errors compared to the unrestricted FMM and that those errors do not disappear under grid refinement [11], i.e. convergence of the scheme to the solution of the Eikonal equation is not given. Recently, [62] have shown that a scaled-down version of the naive (Euclidean distance-based) heuristic does not produce any such errors on triangulated meshes. Since their study did not consider adaptively coarsened meshes, a direct transfer of their method is not straight forward. However, the results in the following example section, cf. paths taken by pedestrians in Appendix B, show that the additional errors introduced by these modifications are negligible with respect to predicting pedestrian flows.

According to the notion  $A^*$ , we will denote the directed fast marching method with the key given in (4.2) in the following FMM\*.

## 4.2 Examples

Results for different values of  $\alpha$  ( $\beta = 1$ ) are shown in Fig. 9 for a simple scenario with one obstacle. One of the main differences between the approach used here and the ideas outlined in [48] is the use of exact distances  $d_{\text{exact}}$  rather than direct distances  $d_{\text{direct}}$ . Comparing the two approaches, cf. Figs. 9(d) and 8 ( $\alpha = 0.6$  and  $\beta = 1.0$ ), we find significant better results in terms of the region where a floor field is calculated (scaling with the computational effort) and the quality of the paths chosen by single pedestrians.

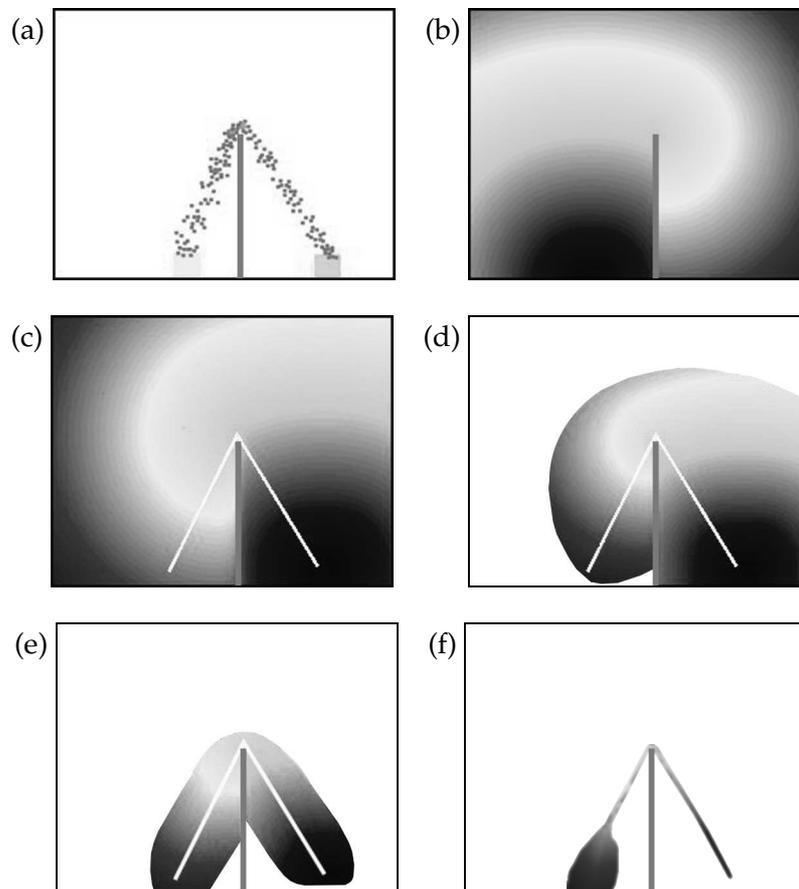


Figure 9: Directed floor fields for the scenario shown in (a) (a snapshot of the simulation is shown) based on shortest path distances to the origin (b). Depending on  $\alpha$  different floor fields are obtained ( $\beta = 1$ ):  $\alpha = 1.0$  (c),  $\alpha = 0.6$  (d),  $\alpha = 0.5$  (e),  $\alpha = 0.4$  (f). The paths taken by a single pedestrians using the calculated fields shown in Figs. (c)-(e) is visualized by a white line. The paths are more realistic than the path obtained by the classical algorithm [48], cf. Fig. 8.

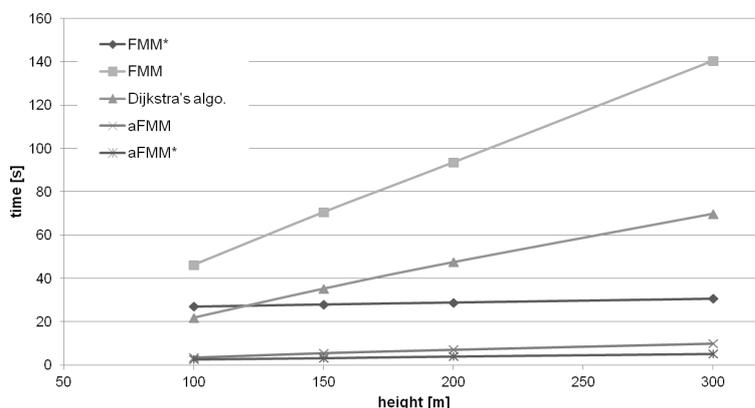


Figure 10: Scaling of computational complexity (overall simulation time) with increasing size of the domain (increasing the height of the scenario shown in Fig. 9a where the wall, origin, and destination are kept fixed).

Adopting the approach outlined here, on the one hand the quality of movement patterns is as exact as it is for the case of an undirected calculation of floor fields – at least up to the eye metric. Thus although we have sacrificed formal convergence for calculating the navigation fields, the accuracy is sufficient with respect to the behaviour of pedestrian crowds. On the other hand, the computational efficiency increases significantly. In Fig. 10, we compare the computational efficiency of the different algorithms and how they scale with the height of the scenario considered in Fig. 9 (without the initialization phase, which includes the calculation of start fields for each origin). The computation times in the case of the FMM\* are increasing only mildly with respect to the size of the domain, as expected.

## 5 An Example – The Marienplatz in Munich

Above we have shown how the efficiency of floor field methods for navigation in microscopic pedestrian simulators can be increased significantly using grid adaptivity and a directed FMM. The examples considered in Sections 3.2 and 4.2 have been more or less of an academic nature. To investigate the performance gain in realistic scenarios we consider pedestrian streams on the Marienplatz in Munich. Pedestrians are entering the scenario from subway exits and are moving towards the Mariensäule or other destinations further away. A snapshot of a simulation is shown in Fig. 11(a). The overall computation times shown in Fig. 11(b) indicate that a factor of 10 is gained for realistic scenarios using the approaches outlined in this paper.

The concept introduced in this paper allows to significantly increase the computational efficiency of dynamic floor field methods, thus that dynamic floor field methods can now be applied for realistic medium scale scenarios with acceptable computational times.

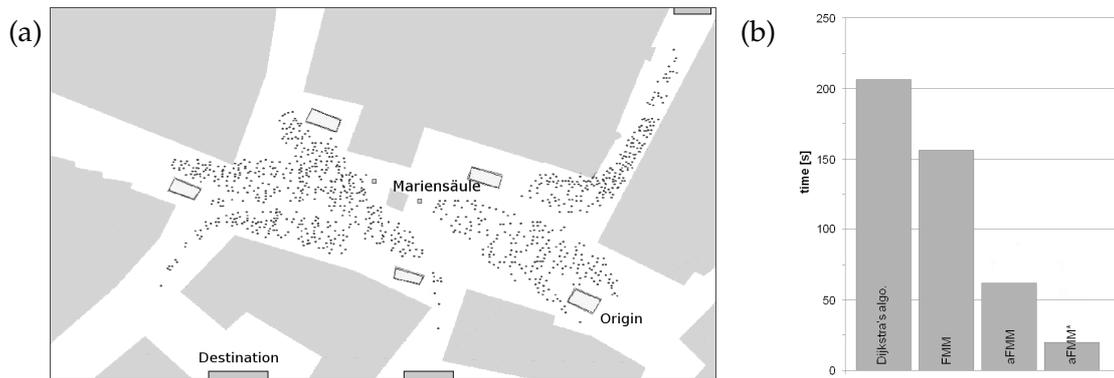


Figure 11: (a) Snapshot of a simulation of the Marienplatz in Munich. Pedestrians are entering the scenario from the underground station (yellow regions) and are steering towards the destinations (blue). (b) Overall computation times for the algorithms calculating floor fields outlined in this paper are compared.

## 6 Discussion and outlook

In this article, we have considered new algorithmic methods in the context of microscopic pedestrian simulations to increase the computational performance of the calculation of dynamic navigation floor fields. One of the most realistic navigation strategies in microscopic pedestrian simulators. Using adaptive strategies as well as a directed calculation of floor fields a performance of factor 10 can be gained without losing quality of the simulation for realistic scenarios. Here, the notion of quality is not understood in the mathematical sense of accuracy, i.e. convergence of results with the discretization getting finer and finer, but rather in a weaker notion whether the evolution of pedestrian densities agrees with observations in reality or not, e.g. agreement of evacuation times. The underlying PDEs are used only as a rule of thumb, formalizing the assumed crowd navigation logic. Thus that the considered mathematical model is very likely to be subject to modelling errors. Therefore, a highly accurate solution of the underlying PDEs would not improve the overall predictive capability of the model. Hence, we accept sacrifice of convergence in terms of computational efficiency as long as the results are sufficiently accurate for the simulation purposes, the prediction of crowd evolution.

Considering a dynamic approach to navigation, floor fields have to be recomputed every few steps. Actually, the overall computational time is mostly determined by the time spend for dynamically calculating floor fields. Thus also the overall computational time is decreased by a factor of 10. Furthermore, at the same time one could also increase the time between update of the floor field using the concept outlined in Appendix Appendix B. Thus in total a decrease of a factor 50 in overall computational time of microscopic pedestrian simulations for realistic scenarios is possible. Large realistic scenarios can now be simulated with a high degree of realism using dynamic floor field methods within acceptable times.

Within this paper we have considered a static adaptive grid. Coarsening the grid up to high levels, this might lead to unrealistic navigation of pedestrians due to the crude approximations in coarse cells. Therefore, we have restricted the coarsening to medium levels, where the local short range repulsion could still resolve unrealistic movement patterns. The extension of the concepts to a dynamic adaptation scheme taking pedestrian densities into account is an open issue. Furthermore, we have restricted our analysis to simple origin destination relations. Considering large scenarios one often encounters complex origin destination relations typically given with multiple intermediate destinations [52]. These scenarios can be efficiently modelled using a combination of graph based and field based navigation [33]. The algorithms introduced in this contribution can be extended straight forward to such a combination of graph and field based navigation. Since the recalculation of the floor fields only increases the realism of simulations if quite crowded regions are involved, a further lever to increase computational efficiency is additionally available: Only on crowded edges of the underlying graph a recalculation of floor fields is undertaken. The combination of both methods in the case of a soccer stadium is under current investigation [1].

The quality of the simulations, i.e. their realism, is more or less based on the eye metric, i.e. whether the evolution of pedestrians looks correct. As long as we do not have sufficient empirical data about the evolution of crowds the notion of accuracy has to be somewhat inexact in the field of pedestrian modelling. A more quantitative approach is therefore an open issue. But since experimental quantitative studies are more and more subject to research, e.g. [38], the next step would be to verify the simulation results with empirical studies.

The outlined algorithms in this paper will enable us to achieve realistic predictions of evolution of pedestrian densities also in the case of relative high pedestrian densities, typically occurring in evacuation studies for large scenarios, with acceptable computational effort (a factor 50 compared to existing approaches has been shown in this paper). The realism of the virtual pedestrian streams is of utmost importance in such applications, since predicted evacuation times and thus decisions of security staff depend on the prediction of these [44]. At the same time such approaches would be used only if computational times are in an acceptable range.

## Acknowledgments

The authors would like to thank the German Federal Ministry of Education and Research who funded our research through the priority program Schutz und Rettung von Menschen within the project REPKA – Regional Evacuation: Planning, Control and Adaptation [1]. Furthermore, the authors would like to thank Alex Vladimirovsky as well as the reviewers for their valuable comments on the manuscript.

## A Cellular automaton model

Although the concept of small to medium navigation is independent of a specific microscopic simulator, we rely within this article on the simple cellular automaton model introduced in [34]. However, generalizations to other microscopic models, e.g. including more complex social behaviour, should be straight forward.

The cellular automaton is based on a lattice graph

$$\mathcal{L}_{l,\vec{x}_0} = \{V_{l,\vec{x}_0}^{\mathcal{L}}, E_{l,\vec{x}_0}^{\mathcal{L}}\} \quad (\text{A.1})$$

with vertices  $V_{l,\vec{x}_0}^{\mathcal{L}} = \{\vec{x} \mid \vec{x} = \vec{x}_0 + \alpha l \vec{e}_\alpha + \beta l \vec{e}_\beta \text{ with } \alpha, \beta \in \mathbb{Z}\}$ , spacing  $l$  and a distinguished node  $\vec{x}_0$ . Here we restrict us to hexagonal geometries, i.e. we choose  $\vec{e}_\alpha = (1,0)$  and  $\vec{e}_\beta = (1/2, \sqrt{3}/2)$ . The edges are given by connection of nearest neighbours (Von-Neumann neighbourhood).

Choosing an appropriate subset  $V_{l,\vec{x}_0}^{\mathcal{A}} \subset V_{l,\vec{x}_0}^{\mathcal{L}}$  and  $E_{l,\vec{x}_0}^{\mathcal{A}} \subset E_{l,\vec{x}_0}^{\mathcal{L}}$  the spatial discretization of the cellular automaton is determined. Centers of hexagonal cells are given by  $V_{l,\vec{x}_0}^{\mathcal{A}}$  and neighbourhood relations are given by  $E_{l,\vec{x}_0}^{\mathcal{A}}$ , where edges from  $E_{l,\vec{x}_0}^{\mathcal{L}}$  cutting walls have been removed. In the following, we will use the notion  $\mathcal{A}_{l,\vec{x}_0} = \{V_{l,\vec{x}_0}^{\mathcal{A}}, E_{l,\vec{x}_0}^{\mathcal{A}}\} \subset \mathcal{L}_{l,\vec{x}_0}$  for the graph associated with the cellular automaton discretization. The size of a single hexagon is chosen such that it accommodates an average European male person having a size of  $0.185 \text{ m}^2$  [57], i.e.  $l = 0.46\text{m}$ . Each hexagon can be in two states, it is either empty or occupied by a single pedestrian or obstacle.

Similarly, the time is discretized into equidistant time steps  $\delta_t$ . Here, we typically use  $\delta_t = 0.17\text{s}$ , such that a person moving every time step can move at maximum with a velocity of  $2.7 \text{ m/s}$ . In each time step, a subset of the persons on the cellular automaton is chosen and allowed to move. We rely on a sequential update which ensures that pedestrians with higher free flow velocities are allowed to move more often according to their speed. Following [57] Gaussian distribution is assumed.

Each pedestrian allowed to move in a time step chooses one of its neighbour cells to move to. This decision is based on potentials borrowing ideas from electro-statics. To each intermediate destination an attractive potential is assigned in the simplest case given by the shortest Euclidean distance (for more details we refer to Section 2). This long range potential used for medium scale navigation (i.e. to the next intermediate destination) is often referred to as floor field or navigation field. Furthermore, to each pedestrian and obstacle a short range (1-2m) repulsive potential is assigned, i.e. a local interaction is considered. The pedestrian then moves to the unoccupied neighbour cell with the smallest summed potential value.

The potential based approach is somewhat equivalent to a force based one using conservative forces. The gradient always points to the direction of the neighbouring cell with the minimal potential. This simple field based approach sketched above is furthermore extended to keep spatial and temporal discretization artefacts to a minimum. For a detailed treatment of the complete concept we refer to the original work of [34].

## B Inertial navigation fields

Dynamic floor field methods, require a constant update of floor fields in order to dynamically take into account other pedestrians for medium range navigation decisions. Within our approach, we rely on fixed update intervals  $\Delta_t = u \cdot \delta_t$  with  $u \in \mathbb{N}$  and a time step length  $\delta_t$  of the microscopic simulator, i.e. floor fields are updated every  $u$  time steps. We have found  $u = 5$  to be a good choice ( $\delta t = 0.17s$ ) yielding the same pedestrian flow patterns (cf. Fig. 12(a)) as choosing an instant update, i.e.  $u = 1$ .

On the one hand increasing  $u$  implies a corresponding decrease in computation times, since the overall computation time of a microscopic simulation is mainly influenced by the time taken for calculating navigation fields. On the other hand, increasing  $u$  unfortunately also implies a decrease in realism since pedestrians do not respond "immediately" to evolving congestions. An example considering counter flows is shown in Fig. 12(b). Increasing  $u$  by a factor of 2 (from  $u = 5$  to  $u = 10$ ) leads to the formation of stripe patterns: Once a region of a higher density has formed, pedestrians avoid this region after an update of the floor field leading to a low pedestrian density. Since this region has now a lower density than surrounding regions, most pedestrian try to walk through this region after the next update of the floor field. This leads again to a higher density. Alternating stripe patterns of high and low pedestrian densities form. Similar artefacts are found in many other different scenarios.

An elegant way to cope with these unrealistic patterns is to include some inertia effects in the floor field, respectively the estimation of expected travel speeds  $V(\vec{x}, t)$ , i.e. to introduce some kind of memory. Instead of considering  $V(\vec{x}, t) = f(\rho(\vec{x}, t))$  in the Eikonal equation (2.1), we rather determine the speed  $V^{n+1}(\vec{x})$  at time step  $n+1$  as a combination of the speed in the previous time step and the current pedestrian density  $\rho^{n+1}(\vec{x})$ , i.e.

$$V^{n+1}(\vec{x}) = (1 - \Delta_t \alpha) V^n(\vec{x}) + \Delta_t \alpha f(\rho^{n+1}(\vec{x})), \quad (\text{B.1})$$

with  $\Delta_t = u \delta t$  and  $f(\rho) = 1/(1 + \omega \rho)$  as discussed in Section 2. Choosing  $\alpha = 1/\Delta_t$  again the original relation is recovered. We would like to remark, that an interpretation of (B.1) in terms of an evolution law discretized with an implicit Euler scheme and time step  $\Delta_t$  is possible.

Let us now compare the results of different  $\alpha$  and  $\Delta_t$ . The results for a counterflow at a narrowing is shown in Fig. 12. One clearly finds that using an inertial update scheme unnatural stripe patterns are not found, cf. Fig. 12(b) vs. (c). The update time can be even further increased without observing unnatural patterns, cf. Fig. 12(d).

Thus, we can conclude that introducing an inertial update a factor of 25 can be gained with respect to computational efficiency without losing realism of the microscopic pedestrian simulations – respectively a factor of 5 compared to  $u = 5$  where no artificial stripe patterns are seen considering standard calculation of floor fields.

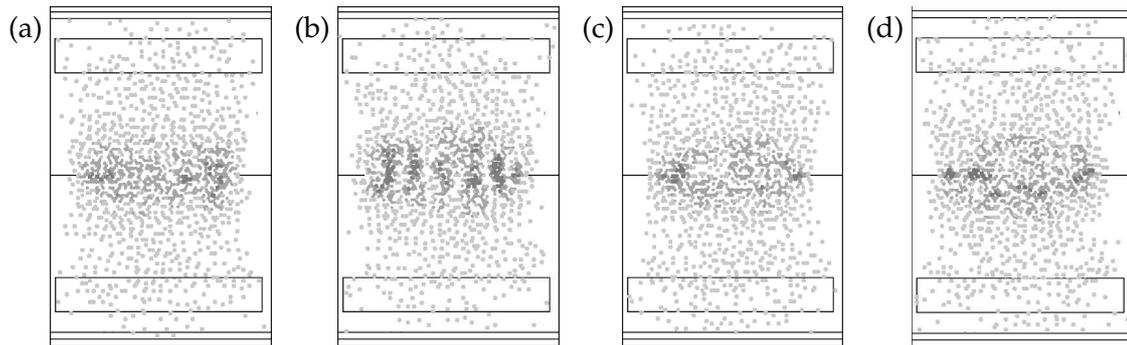


Figure 12: Pedestrian flow patterns of a counterflow with a narrowing due to walls. Pedestrians are moving from the top to bottom and vice versa. Floor fields are updated according to (B.1) with different values of  $\alpha$  and  $\Delta_t = u\delta_t$ . (a)  $u=5$ ,  $\Delta_t\alpha=1.0$ , (b)  $u=10$ ,  $\Delta_t\alpha=1.0$ , (c)  $u=10$ ,  $\Delta_t\alpha=0.5$ , (d)  $u=25$ ,  $\Delta_t\alpha=0.5$ .

## References

- [1] Repka – Regional Evacuation: Planning, Control, and Adaptation. <http://www.repka-evakuierung.de>. Sponsored by Federal Ministry of Education and Research, Germany.
- [2] N. Bellomo and A. Abdelghani Bellouquid. On the modeling of crowd dynamics: Looking at the beautiful shapes of swarms. *Netw. Heterog. Media*, 6:383–399, 2011.
- [3] N. Bellomo and C. Dodge. On the modeling of traffic and crowds: a survey of models, speculations, and perspectives. *SIAM Rev.*, 53:409–463, 2011.
- [4] V. Blue, M. Embrechts, and J. Adler. Cellular automata modeling of pedestrian movements. *IEEE Int. Conf. on Syst., Man and Cybern.*, page 2320, 1997.
- [5] C. Burstedde, K. Klauck, A. Schadschneider, and J. Tittartz. Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. *Physika A*, 295:507–525, 2001.
- [6] A. Chacon and A. Alexander Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM J. on Scientific Computing*, 34:547–578, 2012.
- [7] C. Chalons. Numerical approximation of a macroscopic model of pedestrian flows. *SIAM J. Sci. Comput.*, 29:539V555, 2007.
- [8] M. Chraïbi, U. Kemloh, A. Schadschneider, and A. Seyfried. Force-based models of pedestrian dynamics. *Netw. Heterog. Media*, 6:425–442, 2011.
- [9] M. Chraïbi, A. Seyfried, and A. Schadschneider. Generalized centrifugal force model for pedestrian dynamics. *Phys. Rev. E*, 82:046111, 2010.
- [10] E. Cristiani, B. Piccoli, and A. Tosin. Multiscale modeling of granular flows with applications to crowd dynamics. *Multiscale Model. Sim.*, 9(1):155–182, 2011.
- [11] Z. Clawson, A. Chacon, and A. Vladimirski. Casual domain restriction for Eikonal equations. *SIAM J. on Scientific Computing*, page submitted, 2013. <http://arxiv.org/abs/1309.2884>.
- [12] R. M. Colombo and M. D. Rosini. Pedestrian flows and nonclassical shocks. In *Mathematical Methods in the Applied Sciences*, page 13, 2005.
- [13] V. Coscia and C. Canavesio. First-order macroscopic modelling of human crowd dynamics. *Math. Mod. Meth. Appl. S.*, 18:1217–1247, 2008.
- [14] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271,

- 1959.
- [15] D. Ferguson and A. Stentz. Using interpolation to improve path planning: The field  $D^*$  algorithm. *Journal of Field Robotics*, 23:79–101, 2006.
  - [16] M. Fukui and Y. Ishibashi. Self-organized phase transitions in cellular automaton models for pedestrians. *J. Phys. Soc. Jap.*, 68:2861, 1999.
  - [17] S. Göttlich, S. Kühn, J. Ohst, S. Ruzika, and M. Thiemann. Evacuation dynamics influenced by spreading hazardous material. *Netw. Heterog. Media*, 6:443–464, 2011.
  - [18] H.W. Hamacher, S. Heller, and B. Rupp. Flow location (FlowLoc) problems: dynamic network flows and location models for evacuation planning. *Ann. Oper. Res.*, 207:161–180, 2013.
  - [19] H.W. Hamacher and Tjandra S. Mathematical modeling of evacuation problems: State of the art. In M. Schreckenberg and S.D. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 227–266. Springer, 2002.
  - [20] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100V107, 1968.
  - [21] D. Hartmann. Adaptive pedestrian dynamics based on geodesics. *New J. Phys.*, 12(4):043032, 2010.
  - [22] D. Helbing. A fluid-dynamic model for the movement of pedestrians. *Complex systems*, 6(6):391–415, 1992.
  - [23] D. Helbing and A. Johansson. Pedestrian, crowd and evacuation dynamics. *Encyclopedia of Complexity and Systems Science*, 16:6476–6495, 2010.
  - [24] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282V4286, 1995.
  - [25] L. F. Henderson. On the fluid mechanics of human crowd motion. *Transport. Res.*, 8:509–515, 1974.
  - [26] H.-J. Huang and R.-Y. Guo. Static floor field and exit choice for pedestrian evacuation in rooms with internal obstacles and multiple exits. *Phys. Rev.*, 78(2):021131, 2008.
  - [27] L. Huang, S.C. Wong, M. Zhang, C-W. Shu, and W.H.K. Lam. Revisiting hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm. *Transport. Res. B - Meth.*, 43:127–141, 2009.
  - [28] R. L. Hughes. A continuum theory for the flow of pedestrians. *Transport. Res. B - Meth.*, 36:507–535, 2002.
  - [29] R. L. Hughes. The flow of human crowds. *Annu. Rev. Fluid Mech.*, 35:169–182, 2003.
  - [30] P. Kachroo, S. J. Al-nasur, S. A. Wadoo, and A. Shende. *Pedestrian Dynamics - Feedback Control of Crowd Evacuation*. Springer, New York, 2008.
  - [31] R. Kimmel and J.A. J.A. Sethian. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA*, 95:8431V–8435, 1998.
  - [32] H. Klüpfel. *A Cellular Automaton Model for Crowd Movement and Egress Simulation*. PhD thesis, Universität Duisburg-Essen, Duisburg, 2003.
  - [33] A. Kneidl, D. Hartmann, and A. Borrmann. Using a multi-scale model for simulating pedestrian behavior. In *6th International Conference on Pedestrian and Evacuation Dynamics*, 2012.
  - [34] G. Köster, D. Hartmann, and W. Klein. Microscopic pedestrian simulations: From passenger exchange times to regional evacuation. In *Operations Research - Mastering complexity*, 2010.
  - [35] G. Köster, M. Seitz, F. Tremel, D. Hartmann, and W. Klein. On modelling the influence of group formations in a crowd. *Contemporary Social Science*, 6(3):397–414, 2011.

- [36] T. Kretz. Pedestrian traffic: on the quickest path. *J. Stat. Mech.*, page P03012, 2009.
- [37] T. I. Lakoba, D. J. Kaup, and N. M. Finkelstein. Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution. *Simulation*, 81:339–352, 2005.
- [38] J. Little, A. Seyfried, W. Klingsch, T. Rupperecht, A. Schadschneider, and A. Winkens. An experimental study of pedestrian congestions: Influence of bottleneck width and length. Preprint, 2013.
- [39] J. Little, A. Seyfried, and B. Steffen. Analysis of bottleneck motion using voronoi diagrams. In R.D. Peacock, E.D. Kuligowski, and J.D. Averill, editors, *Pedestrian and Evacuation Dynamics*, pages 833–836. Springer, 2011.
- [40] V. D. Liseikin. *Grid Generation Methods*. Springer, 2009.
- [41] R. Löhner. On the modelling of pedestrian motion. *Appl. Math. Model.*, 34:366–382, 2010.
- [42] D. Maury, A. Roudneff-Chupin, and F. Santambrogio. A macroscopic crowd motion model of the gradient-flow type. *Mathematical Models and Methods in Applied Sciences*, 20:1787–1821, 2010.
- [43] K. Nishinari, A. Kirchner, A. Namazi, and A. Schadschneider. Extended floor field CA model for evacuation dynamics. *IEICE Trans. Inf. Syst*, E87D:726–732, 2004.
- [44] M. Oppenhäuser. Realisierung und Potenzialanalyse von wissenschaftlichen Konzepten zur regionalen Evakuierung aus polizeilicher Sicht am Beispiel des Projektes REPKA. Master's thesis, German Police University, Münster, 2011. [http://opac.pfa-ms.de/onlinedokumente/masterarbeiten/2011/Oppenhaeuser\\_Markus.pdf](http://opac.pfa-ms.de/onlinedokumente/masterarbeiten/2011/Oppenhaeuser_Markus.pdf).
- [45] D. R. Parisi, M. Gilman, and H. Moldovan. A modification of the social force model can reproduce experimental data of pedestrian flows in normal conditions. *Physica A*, 388:3600V3608, 2009.
- [46] N. Pelechano, J. M. Allbeck, and N. Badler. *Virtual crowds: Methods, simulation, and control*. Morgan & Claypool Publishers, San Rafael, Calif., 2008.
- [47] C. Peteres. *Trajectory Planning for Autonomous Underwater Vehicles*. PhD thesis, Heriot-Watt University, Edinburgh, 2007.
- [48] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane. Path planning for autonomous underwater vehicles. *IEEE Transactions on Robotics*, 23:331–341, 2007.
- [49] G. Peyre and L.D. Cohen. Heuristically driven front propagation for geodesic paths extraction. In N. Paragios, O. D. Faugeras, T. Chan, and C. Schnrr, editors, *Proc. of VLSM '05*, pages 173–185. Springer, 2005.
- [50] G. Peyre and L.D. Cohen. Heuristically driven front propagation for fast geodesic path extraction. *International Journal for Computational Vision and Biometrics*, 1:55–67, 2008.
- [51] G. Peyre and L.D. L.D. Cohen. Landmark-based geodesic computation for heuristically driven path planning. In *Proc. of CVPR '06*, pages 2229–2236. IEEE Computer Society, 2006.
- [52] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation dynamics: Empirical results, modeling and applications. In Robert A. Meyers, editor, *Encyclopedia of Complexity and System Science*, volume 3, page 3142. Springer, 2009.
- [53] R. Sedgewick and K. Wayne. *Algorithms*. Addison Wesley, 4 edition, 2011.
- [54] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [55] J.A. Sethian and A.. Vladimirovsky. Fast methods for the eikonal and related hamilton-jacobi equations on unstructured meshes. *Proc. Natl. Acad. Sci. USA*, 97:5699–5703, 2000.
- [56] A. Varas, M. D. Cornejo, D. Mainemer, B. Toledo, J. Rogan, V. Munoz, and J. A. Valdivia. Cellular automaton model for evacuation process with obstacles. *Physica A*, 382(2):631–642, 2007.

- [57] U. Weidmann. *Transporttechnik für Fussgänger*. Schriftenreihe des IVT, 90, 1992.
- [58] Y. Xia, S.C. Wong, and C.W. Shu. Dynamic continuum pedestrian flow model with memory effect. *Phys. Rev. E*, 79:066113, 2009.
- [59] Y. Xia, S.C. Wong, M. Zhang, C.W. Shu, and W.H.K. Lam. An efficient discontinuous galerkin method on triangular meshes for a pedestrian flow model. *Int. J. Numer. Meth. Eng.*, 76:337V350, 2008.
- [60] K. Yamamoto, S. Kokubo, and K. Nishinari. Simulation for pedestrian dynamics by real-coded cellular automata. *Physica A*, 379(2):654, 2007.
- [61] D.S. Yershov and S.M. LaValle. Simplicial Dijkstra and A\* algorithms for optimal feedback planning. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [62] D.S. Yershov and S.M. LaValle. Simplicial Dijkstra and A\* algorithms: From graphs to continuous spaces. *Advanced Robotics*, 26:2065–2085, 2012.
- [63] W. J. Yu, L. Y. Chen, R. Dong, and S. Q. Dai. Centrifugal force model for pedestrian dynamics. *Phys. Rev. E*, 72:026112, 2005.