A pedestrian simulation for hiking in the alps

Christian Gloor, Dept. of Computer Science, ETH Zürich
Laurent Mauron, Dept. of Computer Science, ETH Zürich
Kai Nagel, Dept. of Computer Science, ETH Zürich

STRC 03 Conference paper
Session Model & Statistics

STRC 3rd Swiss Transport Research Conference
Monte Verità / Ascona, March 19-21, 2003

A pedestrian simulation for hiking in the alps

Christian Gloor, Laurent Mauron
Dept. of Computer Science
ETH Zürich
Zürich, Switzerland

Phone: 01-632 04 32
Fax: 01-632 13 74
eMail: chgloor@inf.ethz.ch

Kai Nagel
Dept. of Computer Science
ETH Zürich
Zürichr, Switzerland

Phone: 01-632 54 27
Fax: 01-632 13 74
eMail: nagel@inf.ethz.ch

Abstract

One goal of the project "Planning with Virtual Alpine Landscapes and Autonomous Agents" is to build a multi-agent simulation model of tourists hiking in the Alps. Such a simulation generically consists of two components: The physical mobility simulation, which moves the hikers through the system and computes their interactions; and the strategy generation module(s), which compute(s) strategic decisions of the agents such as destination or route choice. This paper concentrates on the mobility simulation. This paper discusses which kind of simulation is suitable and what model was finally selected. The model is modified for our particular purpose, i.e. for hiking in the Alps rather than crowd or panic simulations in small enclosed spaces. The modified model is then calibrated with real-world data.

Keywords

Pedestrian Dynamics – Multi Agent Simulation – Parallel Computing – 3rd Swiss Transport Research Conference – STRC 03 – Monte Verità

1. Introduction

The project “Planning with Virtual Alpine Landscapes and Autonomous Agents” [1], which is part of Swiss National Science Foundation’s NFP 48 “Landscapes and Habitats of the Alps”, uses a multi agent simulation to model the activities of tourists (primarily hikers). The eventual goal is to have these agents live in a virtual world where they should be able to evaluate different development scenarios for tourism.

Such scenarios include, for example, the question of re-forestation of meadows, or the summer use of chair lifts and the like. Left to themselves, many areas in the Swiss Alps would be covered by dense forest; it seems however that most hikers would prefer a more variable landscape. This is confounded by legal regulations, which essentially allow landowners to prevent forest from growing on meadows, but once a forest is there, it is not allowed to get rid of it again. Similarly, areas which are used for skiing during winters normally make the landscape visually unattractive; on the other hand, many people, in particular families with children or people with health limitations, like mechanical aids to bring them nearer to the top of mountains.

In this situation, it seems that it would be helpful to have models which evaluate these aspects. As so often, it seems that a model that starts from “first principles” (i.e. from modeling the individual people including their decisions directly) offers conceptual and methodological advantages. Also, many questions will be difficult to answer in more aggregated models; for example, it is impossible to think about judging the overall appeal of a tourist area to people with certain demographic characteristics if those demographic characteristics are decoupled from, say, the hiking streams.

All this points to the result that multi-agent simulations for such problems should be tried. Multi-agent here means that indeed each tourist is individually represented, and that his or her full day or full week in the vacation area is simulated. Such an approach is in principle feasible, as similar projects in the area of traffic simulations have shown (e.g. Raney *et al.*, 2003a), but it is unclear if the corresponding behavioral rules and parameters can be calibrated well enough to allow meaningful predictions. – Note that in this paper, the words “pedestrian”, “hiker”, and “agent” will be used interchangeably.

The aim of this project is to implement such a simulation in order to investigate the achievable level of realism. At the same time, the project is used to explore general computational implementations of mobility simulations. As it turns out, mobility simulation systems generically consist of at least two components: the simulation of the physical system, and the simulation of the strategic decisions of the agents (e.g. Ferber, 1999, Chap. 4). While the former is concerned with physical aspects, such as limits on acceleration or speed, or the interaction with other agents, the latter is concerned with the strategic or mental decisions of the agents. The fact that both of them, plus their interplay, are important, has often been neglected in the past – either people coming from physics or similar areas were concerned about the physical simulation but neglected strategies, or people coming from artificial intelligence or similar areas concentrated on simulating “intelligent” agents but neglected to implement a realistic representation of the

[1] see http://www.sim.inf.ethz.ch/projects/alpsim/

physical system (e.g. Ferber, 1999, Chap. 4).

This paper concentrates on the physical simulation, also called mobility simulation. This paper first discusses the type of basic model that was selected and the reasons for this choice (Sec. 2). All of these models are in some way driven by a "desired velocity" of the agent. Sec. 3 thus discusses options of how to obtain that desired velocity in a scenario with hiking paths. Sec. 4 then discusses field experiments that were used to calibrate the model for our purposes. The challenge is to simulate hikers in large geographical areas, of, say, $50\,km \times 50\,km$. Since the simulation uses full two-dimensional space, and needs high spatial resolution in order to be realistic, this poses considerable demands on the computation and memory resources. Sec. 5 discusses computational methods of how to achieve this. The paper is concluded by a quick look on a very preliminary application result, and a summary.

2. Mobility Simulation

As mentioned above, the mobility simulation takes care of the physical aspects of the system, such as interaction of the agents with the environment or with each other. Typical simulation techniques for such problems are:

- In **microscopic simulations**, each particle is represented individually. There are three versions of this:
  - The agents' movements can be given by coupled differential equations. For computer implementations, the differential equations need to be discretized with a time step h.[2] The original model is recovered for $h \to 0$. This is the same technique as applied in molecular dynamics simulations (e.g. Beazley *et al.*, 1995).
  - Sometimes, it makes sense to define the agents' dynamics directly in coarse-grained time. This makes for example sense for traffic (Krauß, 1997), where the reaction time plays an important role. In these models, the time step h needs to be selected with care, and the limit $h \to 0$ is *not* meaningful for such models. Such models are sometimes called **coupled map lattices**.
  - Analogous to coarse-grained time, it is also possible to coarse-grain space. That is, again one does not consider the limit $\xi \to 0$ of the discretization constant, but the model is explicitely formulated with a specific spatial resolution in mind. These models are typically called **cellular automata** (CA, e.g. Wolfram, 1986).
- In macroscopic or **field-based simulations**, particles are aggregated into fields. The corresponding mathematical models are partial differential equations, which can be discretized for computer implementations.
- It is possible to combine microscopic and field-based methods, which is sometimes called **smooth particle hydrodynamics** (SPH, Gingold and Monaghan, 1977). In SPH, the individuality of each particle is maintained. During each time step, particles are aggregated to field quantities such as density, then velocities are computed from those densities, and then each individual particle is moved according to those macroscopic velocities.
- As a fourth method, somewhat on the side, exist the queuing simulations from operations research. Here, particles move in a networks of queues, where each queue has a service rate. Once a particle is served, it moves into the next queue. Most queuing models do not model spillback, i.e. the fact that queues can extend upstream.

For our simulations, we need to maintain individual particles, since they need to be able to make individual decisions, such as route choices, throughout the simulation. This immediately rules out field-based methods. We also need a realistic representation of inter-pedestrian interactions, which rules out both the queue models and the SPH models with their too simplified dynamics. This leaves the microscopic models. We decided against using a CA technique for the following

[2] Event-driven simulations are possible, but not considered here.

reason: Hikers make movements into arbitrary directions. CA models however are best at representing movement along the main directions, which could be four or six depending on the choice of quadratic or hexagonal grid. Off-axis movements in CA models is typically represented by using different probabilities for different directions. This however results in erratic movements for off-axis movement, and in different directional variabilities for on-axis vs. off-axis movement. For crowd simulations, such as for evacuation (e.g. Meyer-König *et al.*, 2001), this is not of major importance. But for our application, where it may even be necessary to follow the eye movements of an individual agent, this makes CA simulations awkward.

This leaves the simulations based on continuous space. Here, models derived from coupled differential equations seem to be much better understood than coupled maps (for which we are not even aware of an example), which is why we decided to use the former. A generic coupled differential equation model for pedestrian movement is (Helbing *et al.*, 2000)

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{\mathbf{v}_i^0 - \mathbf{v}_i}{\tau_i} + \sum_{j \neq i} \mathbf{f}_{ij} + \sum_{W} \mathbf{f}_{iW} \tag{1}$$

where m_i is the mass of the pedestrian and $\mathbf{v}_i$ its velocity. $\mathbf{v}_i^0$ is its desired velocity; in consequence, the first term on the RHS models exponential approach to that desired velocity, with a time constant τ_i. The second term on the RHS models pedestrian interaction, and the third models interaction of the pedestrian with the environment.

Modeling the desired speed $\mathbf{v}_i^0$ is critical for hikers since they need to be able to follow arbitrarily complicated paths through variable terrain. In the simplest model, one assumes that the hiking path is defined by a large enough number of waypoints, and the agent simply heads towards a waypoint (marked “w” in Fig. 1). That is, if the agent is at $\mathbf{r}_i$ and the waypoint is at $\mathbf{R}$, then the desired velocity of the agent is given as

$$\mathbf{v}_i^0 = v_i^0 \, \frac{\mathbf{R} - \mathbf{r}_i}{|\mathbf{R} - \mathbf{r}_i|} \ ,$$

where v_i^0 is the magnitude of the desired velocity. Once a waypoint is reached, $\mathbf{R}$ is moved to the next waypoint. – A disadvantage of this method is that pedestrians are artificially pulled towards that waypoint even in situations where this is not plausible. This will be discussed in more detail in Sec. 3.

The specific mathematical form of the interaction term does not seem to be critical for our applications as long as it decays fast enough. Fast decay is important in order to cut off the interaction at a relatively short distance. This is important for efficient computing, but it is also plausible with respect to the real world: Other pedestrians at, say, a distance of several hundred meters will not affect a pedestrian, even if those other pedestrians are at a very high density. We use an exponential force decay of

$$\mathbf{f}_{ij} = \exp\left(\frac{|\mathbf{r}_i - \mathbf{r}_j|}{B_p} \right) \, \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|} \ ,$$

which seems to work well in practice. $\mathbf{f}_{ij}$ is the force contribution of agent j to agent i; $\mathbf{r}_i$ is the position of agent i. Alternative more sophisticated formations are descibed by Helbing *et al.* (2000).

This leaves the environmental forces, $\mathbf{f}_{iW}$. The most important environmental influence is giving by the hiking path. The waypoints, as described above, pull the pedestrian towards a point on the path, but they do in general not pull the pedestrian onto the path. One option to achieve the latter is to give locations $\mathbf{x}$ outside a path a force towards the nearest path (see Fig. 1 for a cell-based example which is discussed in Sec. 5). As already said, improvements of this are discussed in Sec. 3. As the only other aspect of environmental forces, point-like obstacles are modeled as non-moving pedestrians.

In the implementation, a time-step of $h = 0.5s$ is used. Pedestrians are considered one at a time. Velocity is updated according to

$$\mathbf{v}_{i,t+1} = \mathbf{v}_{i,t} + h\left(\frac{\mathbf{v}_i^0 - \mathbf{v}_{i,t}}{\tau_i} + \frac{1}{m_i}\sum_{j\neq i}\mathbf{f}_{ij} + \frac{1}{m_i}\sum_{W}\mathbf{f}_{iW}\right) \; ; \tag{2}$$

position is updated in parallel according to

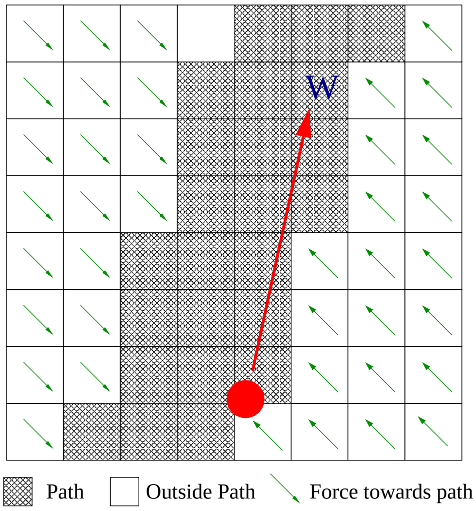
$$\mathbf{r}_{i,t+1} = \mathbf{r}_{i,t} + \mathbf{v}_{i,t+1} \; .$$

As of now, the update is sequential; for our problems, no large differences to parallel update were observed.

In fact, it may happen that the velocity according to Eq. (2) is much larger than is plausible. For that reason, if the magnitude of the velocity is larger than v_i^0, it is artificially reduced to v_i^0.

An additional element is given by a so-called **walkability** parameter $w(\mathbf{r}_i)$. This number, between 0 (obstacle) and 1 (flat street), parametrizes the walkability of the terrain. This number depends on the position $\mathbf{r}_i$ of the agent. This, instead of reducing the velocity to v_i^0 as indicated in the last paragraph, it is actually reduced to $v_i^0\, w(\mathbf{r}_i)$.

Figure 1: Example of the force towards the waypoint. It is at the same time an illustration of the use of cells for local force information.



3. Improved version of the path force ("model B")

The pedestrian model described in the previous section ("model A") suffers from some shortcomings:

- Agents move in the direction of a waypoint. This is for example a problem if they are on a wide straight street with a waypoint in the middle, because they will all move towards that waypoint instead of remaining on their side of the street (see Fig.2).
- If two paths are close together and a pedestrian was pushed away, say by other pedestrians, too far from its own path, the environmental forces which pull the pedestrian to the nearest path may pull the pedestrian to the wrong path. This will eventually be corrected when the simulation continues since the pedestrian is still pulled towards the correct waypoint, but it looks implausible. The conceptual reason behind this is that in model A the path forces depend on the location rather than on the agent's own intentions.

In this section, a model will be presented that allows agents to follow a path without those two artifacts.

The idea is to move to a force system that follows the path. The path is given by a line, which in turn is given by our input data. This path line may for example be a piecewise linear object, or a Bezier curve, or a spline.[3] Each pedestrian now has, besides its true location, a **shadow tag**

[3] In fact, standard B-splines do not work well; instead, one can use Akima splines (Akima, 1972).

on that line. The shadow tag is always computed such that the connection between the shadow tag and the pedestrian is orthogonal to the path line. The desired speed $\mathbf{v}_i^0$ is now computed such that it is in the direction of the path line (light arrows in Fig. 3); and there will be a weak environmental force component towards the center of the path (dark arrows in Fig. 3). In consequence, agents now essentially move parallel to the path direction and do *not* have a tendency any more to go to the center of the path near waypoints. The success of this is documented in Fig. 4.

Note that the shadow tag is used only for computing the path force. Once the agent has moved, the position of the shadow tag needs to be computed again. In general, this is a non-linear problem and thus needs to be solved by an iterative algorithm, for example the one by Brent. Even with this algorithm, it is clear that it can become stuck in a local minimum, and care needs to be taken to prevent that situation. The situation gets confounded by the fact that shadow tags need to be passed on across waypoints, and the best position for a shadow tag could be already on the next segment. Some details of this are discussed by Mauron (2002).

In our case, it turns out that the path data is good enough that a piecewise linear interpolation is sufficient. For this case, the position of the shadow tag can be computed directly, and the only thing that can happen is that is has moved on to the next segment. What we do is to calculate the distance d_{i+1} of the pedestrian also to the next segment P_{i+1} and compare it with d_i. If d_{i+1} is smaller or equal than d_i, then P_{i+1} becomes the current segment; that is, if there is a choice, then the next segment is chosen. A resulting artifact of this is that at connections with angles larger than 90 degrees, agents will move towards the inner side of the curve at an implausible location. For our purposes, this artifact can be accepted, since such situations are rare with the given data.

Technically, this means that the environmental force contribution, $\sum_W \mathbf{f}_{iW}$, is now separated into two terms,

$$\sum_W \mathbf{f}_{iW} = \mathbf{f}_{obstacles} + \mathbf{f}_{path} \ . \tag{3}$$

Obstacles are still simply treated as non-moving pedestrians. The path force $\mathbf{f}_{path}$ (denoted by dark arrows in Fig. 3) keeps the pedestrian on the path and is perpendicular to the path line. It is given by

$$\mathbf{f}_{path,i} = A_{path,i} \left[\exp\left(\frac{h_1 - h}{B_{1,i}} \right) - \exp\left(\frac{h - h_2}{B_{2,i}} \right) \right] \mathbf{n}(s) \ , \tag{4}$$

where $\mathbf{n}(s)$ is the vector normal to the path, h_1 and h_2 characterize the path width, and the constants $B_{1,i}, B_{2,i}, A_{path,i}$ characterize resp. the range and strength of the pedestrian-path interaction, individually for each pedestrian i. Some plots of $|\mathbf{f}_{path,i}|$ can be found in Fig. 6. The force Eq. (4) can be derived from a potential as $\mathbf{f}_{path,i} = -\partial_h V_i$ with

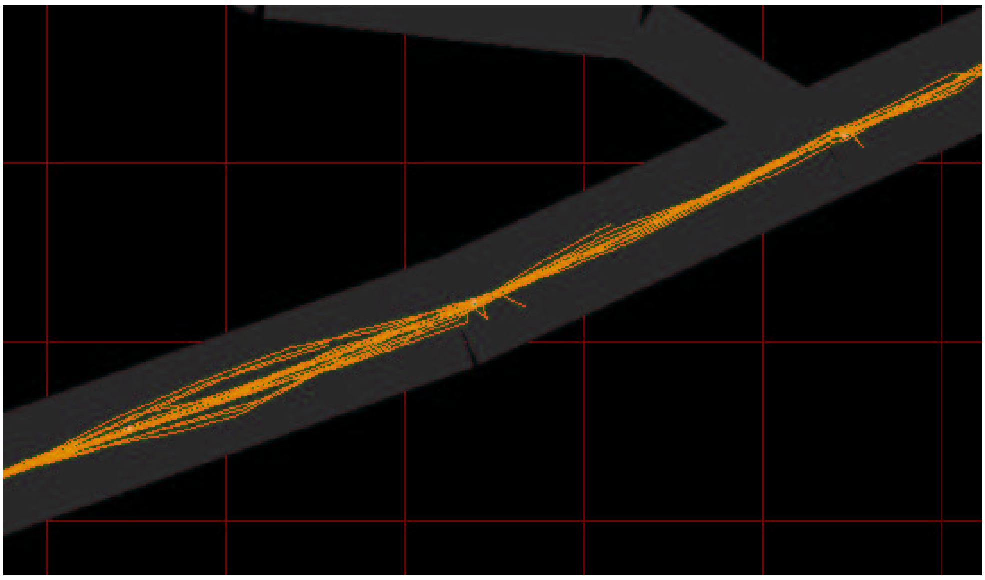
$$V_i = A_{path,i} \left[B_{1,i} \ \exp\left(\frac{h_1 - h}{B_{1,i}} \right) + B_{2,i} \ \exp\left(\frac{h - h_2}{B_{2,i}} \right) \right] \ .$$

Since the potential is more intuitive than the force plot, it is given in Fig. 7.

Note that an equilibrium position can be found by setting the force equal to zero; this results in

$$h_{0,i} = \frac{B_{2,i} \, h_1 + B_{1,i} \, h_2}{B_1 + B_2} \ . \tag{5}$$

Figure 2: Traces of hikers in “model A”, where they are all pulled towards the same waypoint. Note how the trajectories focus near the waypoint, and diverge before and after. The width of the path remains unchanged.



Sec. 4 will describe an asymmetric walkway where all these parameters are indeed used. In general, however, we will use $h_1 = -h_2$, $B_{1,i} = B_{2,i}$, and a uniform A_{path}.

Figure 3: Path-oriented coordinate system for the computation of the desired velocity and the path forces. The light arrows show the desired velocity, which drives the agent forward along the path. The dark arrows show the path force, which pull the agent towards the middle of the path.

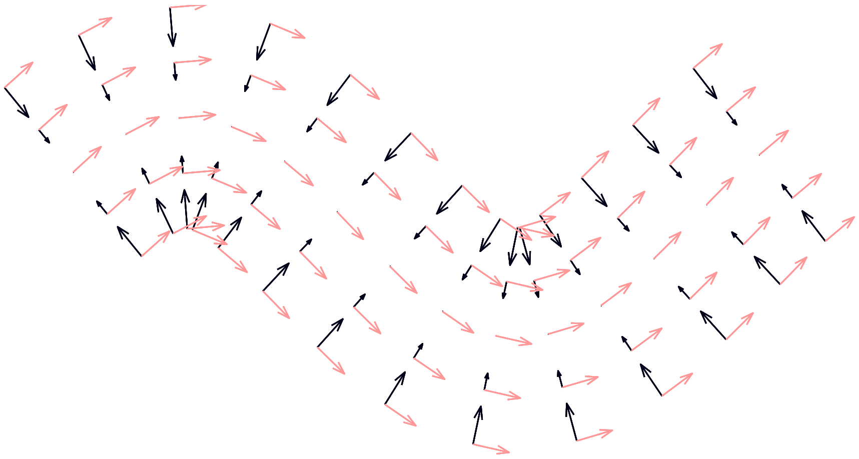


Figure 4: Traces of pedestrians walking in the same direction according to “model B”. Note that they stay on their side of the path, even at a bend.

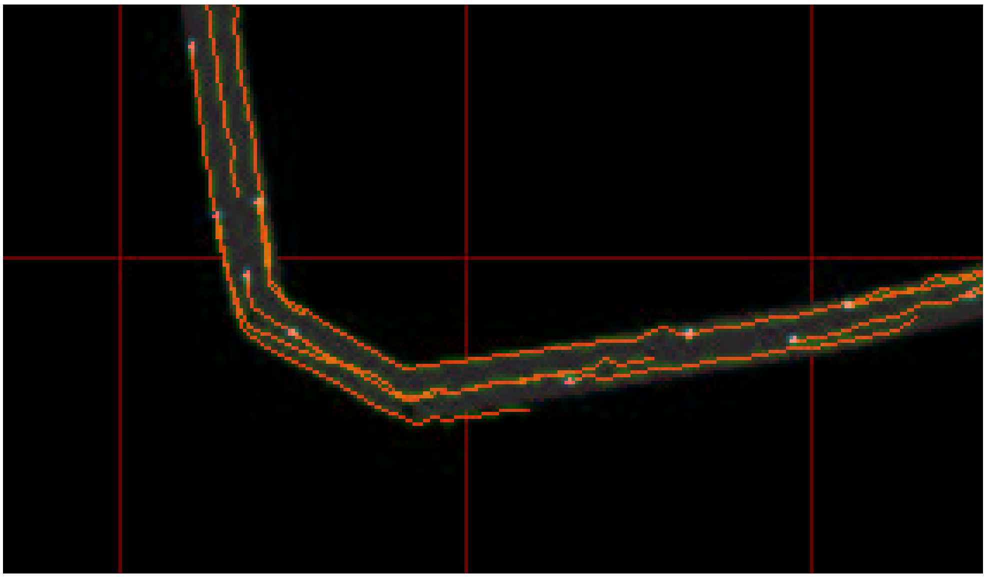
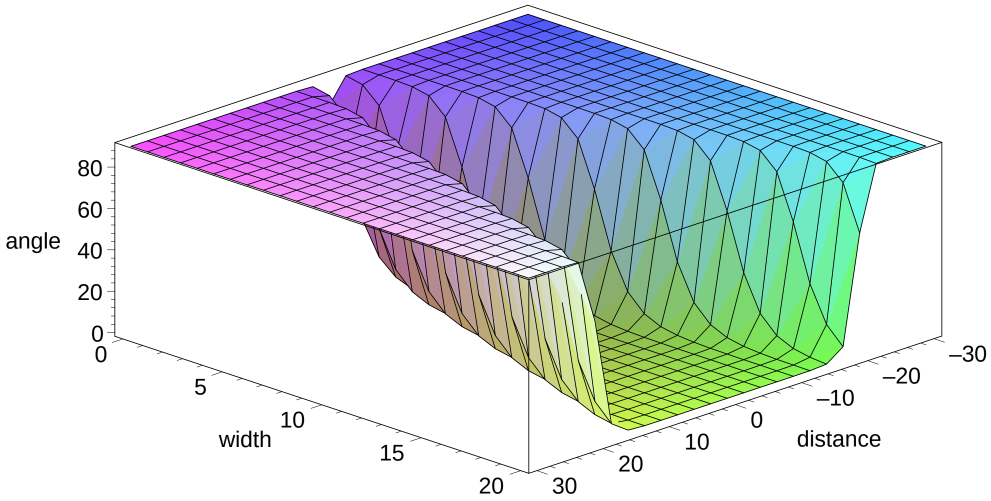


Figure 5: Angle between the "force" caused by the desired velocity, and the path force. This is both a function of the path width, and the distance from the center of the path. At large distances from the path, the agent is driven towards the path with no component in its desired walking direction. On wide paths, the lateral component of the force is nearly zero. For the plot, the velocity of the agent, $\mathbf{v}_i$, is assumed to be zero.



4. Model Calibration

In this section we present field measurements on pedestrian flow we have performed using a video camera. We took video footage of pedestrians walking on a sidewalk and measured their distance to the borders. We used these observations to select an appropriate path force model and calibrate various model parameters.

The calibration of the model parameters like the force constants A and B in Eq. (4) can be difficult. A common method in microscopic modeling is to use observed macroscopic properties of the flow, like speed-density curves or flow-density and adapt the simulation parameters to match them (Hoogendorn *et al.*, 2002). A major difficulty in the case of pedestrian flow is that the speed-density curves, the so-called *fundamental diagrams* are scarce. The usual reference curve, the Weidmann curve (Weidmann, 1992), although generally acknowledged as realistic , was not tested on the field. An additional problem with speed-density curve calibration is the lack of specificity since one can only calibrate the whole set of parameters and not only single parameters. A more precise fine grained calibration would be obtained by measuring specific properties of the traffic flow but here again, the lack of proper studies causes great difficulties, although several efforts are currently conducted to obtain quantitative data from pedestrian flow.

Figure 6: Force $|\mathbf{f}_{path}(h)|$ in a 2 meters wide sidewalk ($B_1 = 0.5$). Note that the plot shows the absolute value $|.|$ of the force; to the right of where the force becomes zero, the force is actually negative.

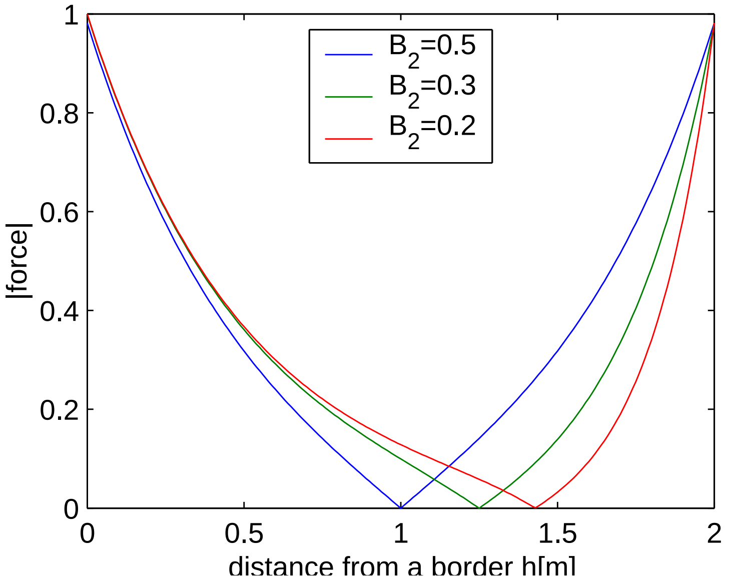
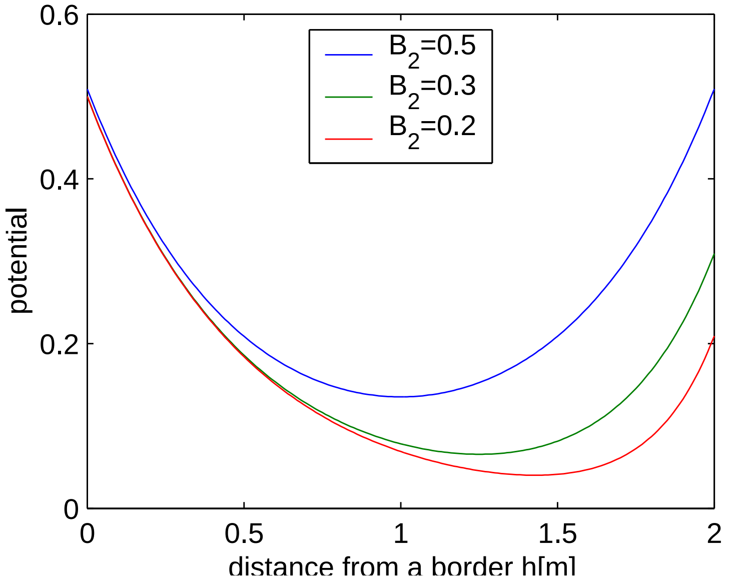


Figure 7: Potential in a 2 meters wide sidewalk ($B_1 = 0.5$)



4.1 Experimental setup

Pedestrian movements were observed at a sidewalk on Tannenstrasse, between the CLA building and the main building of ETH in Zurich (Fig. 8). Advantages of that location included: it was possible to place the camera high above the observed area; different observation times resulted in different flow characteristics; the area is devoid of inhomogeneities such as additional entrances/exits, shopping windows, etc. A disadvantage is the slight uphill grade of the sidewalk.

After the video footage was taken, pedestrian movements had to be translated into a Cartesian field coordinate system. This was achieved by a half-automatic image analysis and coordinate conversion software written explicitely for this purpose (Mauron, 2002). The system was calibrated by four control points that were marked with bright duct tape on the pedestrian side walk. The field coordinates of those points were determined using standard measuring tape. A 1-meter reference stick, randomly placed on the sidewalk and analysed via the system resulted in a length error of about $5cm$, which is reasonable for the purpose here. The width of the sidewalk was $2.5\ m$, resulting in

$$h_1 = 0.0\, m \text{ and} \tag{6}$$

$$h_2 = 2.5\, m\,. \tag{7}$$

After this, movements of real pedestrians were tracked. For this, second-by-second video images were read into the software, and their estimated projection of their center of gravity to the sidewalk was manually determined. That position was then converted by the software into field coordinates. Two situations were distinguished:

- **Single pedestrian.** No other pedestrian was within a 10 meter radius.
- **Crossing pedestrians.** Two pedestrians of opposing direction pass each other, without any other pedestrian within a 10 meter radius.

As will be explained below, the first scenario was used for the calibration of the B_i, while the second scenario was used to calibrate pedestrian-pedestrian interaction. 475 non-interacting pedestrians and 150 crossing events were used for the analysis. The resulting distributions are shown in Fig. 9.

4.2 Calibration of the path force

The observation of *non-interacting pedestrians* on a straight sidewalk have shown that they tend walk in straight line keeping a constant distance h from the road border. The radial pedestrian distribution obtained with the field measurements reveal that the h's are statistically distributed around a mean value $\overline{h}_0 \approx 1.6\ m$. The pedestrians tend to keep a larger distance from the street than from the wall which is understandable since the street is potentially more dangerous.

For a calibration of the path force constants, it was assumed that the path force constants $B_{1,i}$ and $B_{2,i}$ are normally distributed with a mean value $\overline{B}_1$, $\overline{B}_2$ and the same standard deviation ΔB.

This results, via Eq. (5), in a distribtion for the equilibrium values $h_{0,i}$. The constants $\overline{B}_1$, $\overline{B}_2$, and ΔB were now varied so that the resulting distribution of the $h_{0,i}$ matches as well as possible the field measurement distribution; note that the value of A does not matter here. The method converged to the following values,

$$\begin{aligned} \overline{B}_1 &= 3.3\,m, && (8)\\ \overline{B}_2 &= 1.1\,m, && (9)\\ \Delta B &= 0.8\,m. && (10) \end{aligned}$$

The corresponding pedestrian distribution is shown in figure 10. Although the simulated distribution peaks roughly at the same value h_o, the field distribution is significantly higher at h_o and decreases faster approaching the wall.

4.3 Calibration of pedestrian interaction

Given the above calibration of the path force, the field measurements of the two-pedestrian-encounters can be used to calibrate the pedestrian-pedestrian interaction. The basic assumption here is that this interaction follows the same functional form as the path force, i.e.

$$\mathbf{f}_{pp}(\mathbf{x}_{ab}) = A_{pp}\,\exp\left(\frac{|\mathbf{x}_{ab}|}{B_{pp}}\right)\frac{\mathbf{x}_{ab}}{|\mathbf{x}_{ab}|}\ ,$$

where $\mathbf{x}_{ab}$ is the vector from the position of b to the position of a. The strength of the interaction force, A_{pp}, is assumed to be the same as the strength of the path force, A_{path}. Finally, it is assumed that the path force, with $B_{1,i} = \overline{B}_1$ and $B_{2,i} = \overline{B}_2$, should be exactly in equilibrium with the interaction force when the pedestrians are side-by-side; the measured average positions when the pedestrians are side-by-side are

$$\begin{aligned} \overline{h}_a &= 0.85 \pm 0.25\,m, && (11)\\ \overline{h}_b &= 1.88 \pm 0.20\,m. && (12) \end{aligned}$$

For the forces to cancel out, this results in two equations, one for each pedestrian:

$$\begin{aligned} \mathbf{f}_{path}(\overline{h}_a) + \mathbf{f}_{pp}(\overline{h}_{ab}) &= \mathbf{0} && (13)\\ \mathbf{f}_{path}(\overline{h}_b) + \mathbf{f}_{pp}(\overline{h}_{ba}) &= \mathbf{0}\ . && (14) \end{aligned}$$

$\overline{h}_{ij} = \overline{h}_i - \overline{h}_j$ is the distance between the two average values.

Inserting the force equations for the first equation one obtains

$$\exp\left(\frac{h_1 - \overline{h}_a}{\overline{B}_1}\right) - \exp\left(\frac{\overline{h}_a - h_2}{\overline{B}_2}\right) - \exp\left(\frac{\overline{h}_a - \overline{h}_b}{B_{pp}}\right) = 0\ , \qquad (15)$$

where already the assumption was made that all force strengths A are the same. Solving this equation with respect to B_{pp} results in

$$\begin{aligned} B_{pp} &= \frac{\overline{h}_a - \overline{h}_b}{\log\left(\exp\left(\frac{h_1 - \overline{h}_a}{\overline{B}_1}\right) - \exp\left(\frac{\overline{h}_a - h_2}{\overline{B}_2}\right)\right)} && (16)\\ &\approx 1.71\,m. && (17) \end{aligned}$$

Given our previous values for $\overline{B}_1$ and $\overline{B}_2$, this value seems plausible.

Doing the same procedure for the second equation results in a much lower estimate for B_{pp} of $0.18\,m$. This is due to the fact that for a pedestrian close to the wall the presence of a second pedestrian does not shift the average value for h a lot – from $\overline{h}_0 \approx 1.6\,m$ to $\overline{h}_b \approx 1.88\,m$ – and therefore the other pedestrian does not have to "push" a lot to achieve this. We conclude that for our purposes an interaction range of $B = 1\,m$ is plausible, both for the path force and for the interaction force.

4.4 Fundamental diagrams, and the interaction strength A

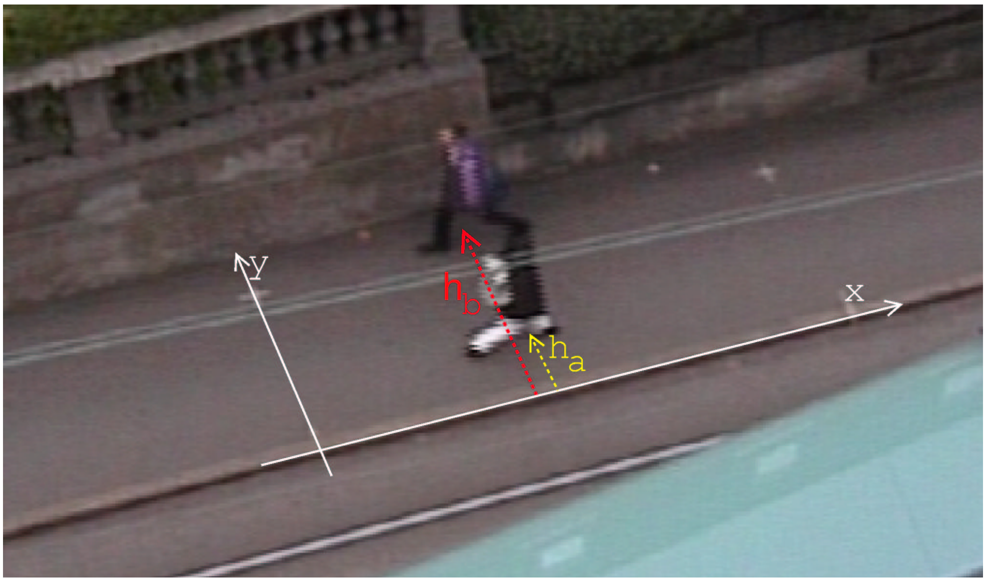
We measured the behaviour of simulated pedestrians on a path with width $2\,m$, i.e. $h_1 = 0\,m$ and $h_2 = 2\,m$. The length of the path segment was $10\,m$. For this section, a constant and large path force strength of $A_{path} = 6000$ was selected, while the pedestrian interaction strength, A_{pp}, was varied as indicated. The large path force models solid walls. We were interested in the change of mean walking speed $\overline{v}$ of the agents if we change their density ρ. A curious result when doing this with periodic boundary conditions is that the velocity does not go down at all, even with very high density. The reason is that the pedestrians move as a single block even at very high densities; this is caused by a combination of two properties of the equations: The pedestrian interaction is uniform in all directions, that is, a pedestrian from behind pushes as much as a pedestrian in front. Second, nothing in the formulation says that their velocity should depend on some distance from each other as it is the fact with most car following equations.

This is clearly unrealistic. It however turns out that with "randomized" boundary conditions, the problem goes away and reasonable fundamental diagrams result (Fig. 11). With randomized boundary conditions, a pedestrian reaching the end of the segment will reappear at the start of the other end with a randomized lateral position. Since now the boundary conditions are not periodic, the pedestrian about to leave the corridor at one end does not push the pedestrians on the other end. This can cause unrealistic overlapping pedestrians. In order to avoid that, the simulation checks if there is space available at the randomized location (for the precise algorithm see Mauron (2002)). If the location is already occupied, the pedestrian is stuck and the procedure is repeated at every simulation time step till there is space. – This essentially means that with randomized boundary conditions the "pushing from the back" is not done at the entry to the path segment.

From Fig. 11, one observes that an interaction strength of $A_{pp} = 100$ is too weak. Values of $A_{pp} \approx 300$ are plausible.

Finally, Fig. 12 shows fundamental diagrams when a share of pedestrians, C, is moving in the opposing direction. Clearly, with C increasing towards 0.5, the average velocity goes down. As expected, the effect is more pronounced for high than for low densities. We are not aware of any systematic field measurements to compare this data against.

Figure 8: Measured quantities



5. Computational aspects

It was argued earlier that a cellular automata representation of space did not seem appropriate for our purposes. Instead, we use a continuous representation of space. However, some aspects of our simulation, such as walkability or obstacle forces, depend on the spatial location of the agent. The same is true for the path forces in “model A” (Sec. 2), which we still intend to use for simulations with complicated geometries such as inside buildings. These forces are relatively expensive to calculate, since one needs to enumerate through all possible objects that could influence a given location.

However, since those forces do not depend on time, they can be pre-computed before the simulation starts. In order for this to be successful, some coarse-graining of space is necessary. For this, we use cells of size $25cm \times 25cm$, and assume that all time-independent forces are constant inside a cell. The resulting force field (Fig. 1) becomes non-continuous in space, but this is not a problem in practice since this only influences the acceleration of pedestrians. That is, the acceleration contribution from the environmental forces can jump from one time step to the next, but since time is not continuous, this is not noticeable.

Unfortunately, precomputing the values for all cells in a hiking region of, say, $50km \times 50km$, does not fit into regular computer memory. To avoid this problem, we implemented two methods: lazy initialization, and disk caching (Fig. 13). By **lazy initialization**, we mean that the values are computed only when an agent really needs them. In practice, the simulation area is divided into blocks of size $200m \times 200m$. Every time an agent enters one of these blocks, the values for

Figure 9: Real-world pedestrian distribution for non-interacting and interacting pedestrians

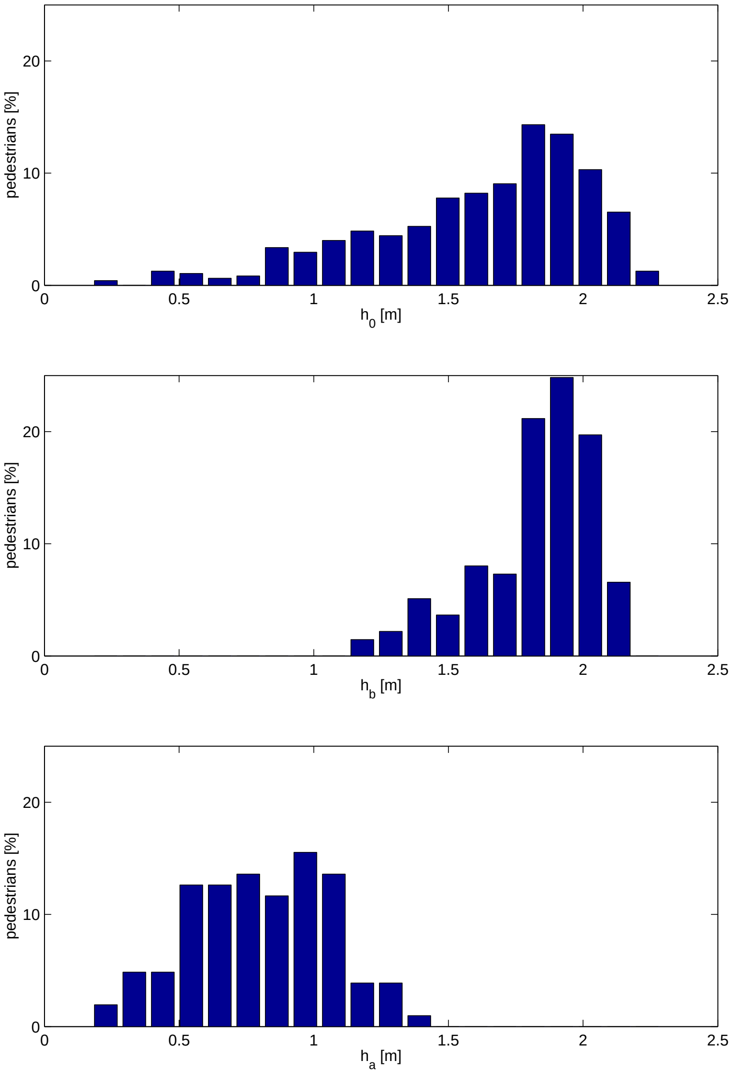


Figure 10: Validation: Real-world and simulated single pedestrian distribution

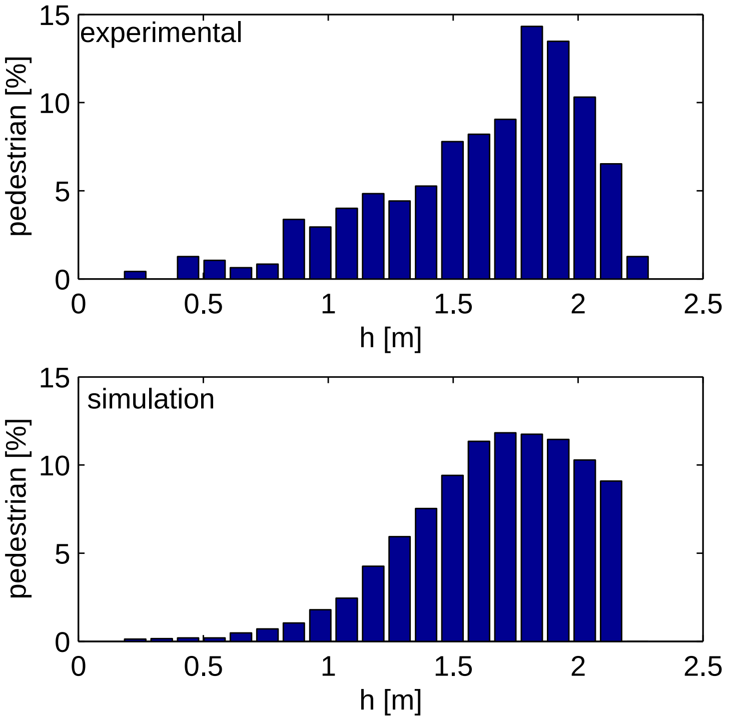
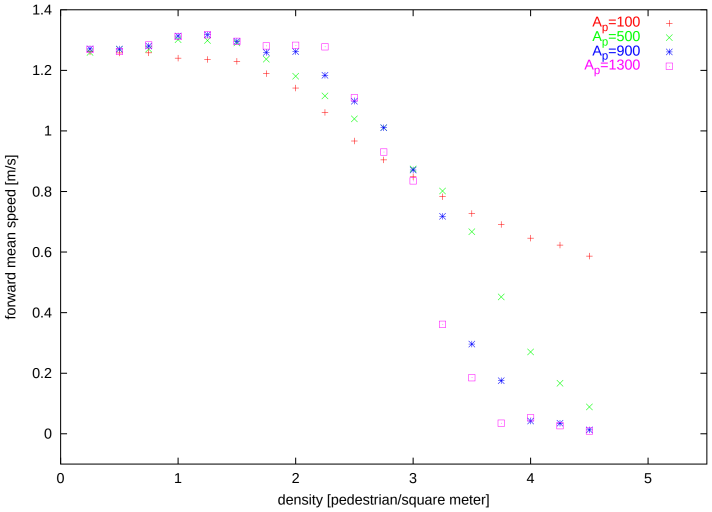


Figure 11: Average speed $\overline{v}$ vs. pedestrian density for uni-directional flow and randomized boundary conditions ($B_{pp} = 1$, $L = 10$[m], $W = 2$[m])



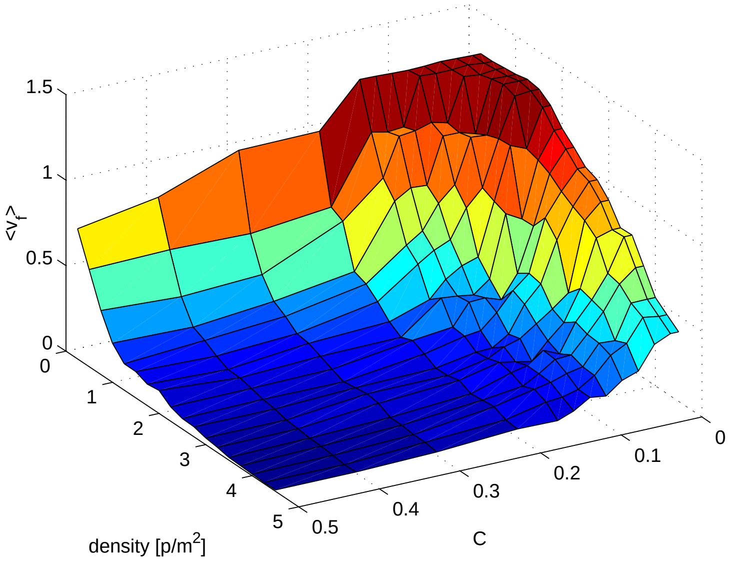
all cells inside that block are computed. Since hiking paths cross only a small fraction of those blocks, the cell values for many blocks in our hiking area will never be calculated.

In addition, the cell values, once computed, are stored on disk (**disk caching**). Every time when an agent encounters a block for which the cell values are not in memory, the simulation first checks if they are maybe on disk. Computation of the cell values is only started when those values are not found on disk. In consequence, a simulation started for the first time will run slower, because the disk cache is not filled yet.

If the simulation runs out of memory, then blocks which are no longer needed (i.e. which have not been crossed by an agent for a long time) are unloaded from memory. If they are needed again, they are just re-loaded from disk. It would also be possible to allocate main memory for all the blocks and rely in the paging mechanism of the operating system. However, since we know more about the simulation, we are able do optimize the parameters for this special purpose.

An additional advantage of the blocks, well known from molecular dynamics simulations, is that one can use them to cut off the short-range interaction between the pedestrians. Agents which are not in the same or one of the eight adjacent blocks are ignored (Fig. 14). This implies that there needs to be some data structure where agents are registered to the block. Agents that move from one block to another need to unregister in the first block and register in the second one. In this way, an agent searching for its neighbors only needs to go through the registered agents in the relevant blocks. This brings the computation complexity from O(N^2) down to O(NM), where M is the number of agents in a single block. M is a reasonably small number when compared to the number N of all agents in a real-world scenario.

Figure 12: The maximal velocity of walking pedestrians decreases if their density rises, and when oncoming traffice increases. $A_{pp} = 300$, $B_{pp} = 1$, $A_{path} = 6000$, random boundary conditions.



For a testing scenario, of size $12\,km \times 15\,km$, we would need approx. 2.9×10^9 cells or 4500 blocks, resulting in 9 GByte memory requirement. The result of the lazy initialization together with the caching mechanism is that 50 MByte are enough for the scenario shown in Fig. 16. The computational speed for that simulation, with 300 hikers, was about 40 times faster than real time.

Figure 13: Since block consume a lot of memory, they are loaded into memory as soon as an agent walks over, and are deleted after a while.

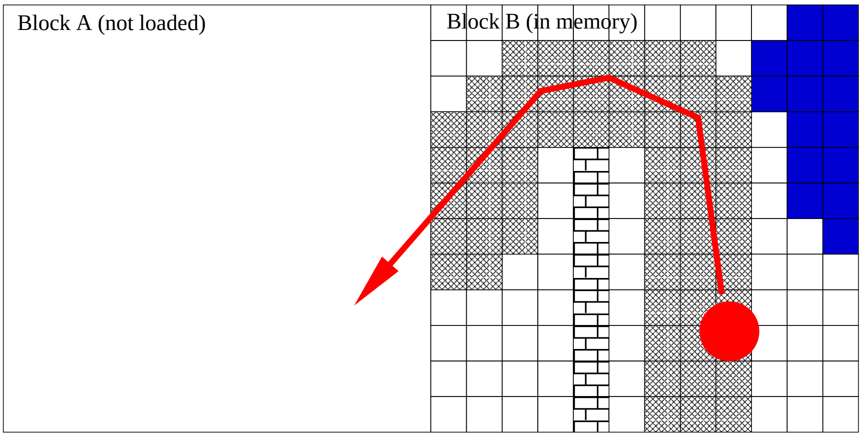


Figure 14: The scene is divided into multiple “blocks”. Only forces from adjacent blocks have an influence to an agent.

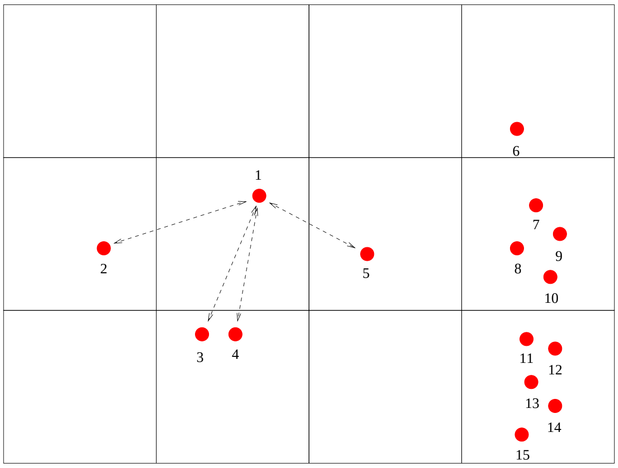
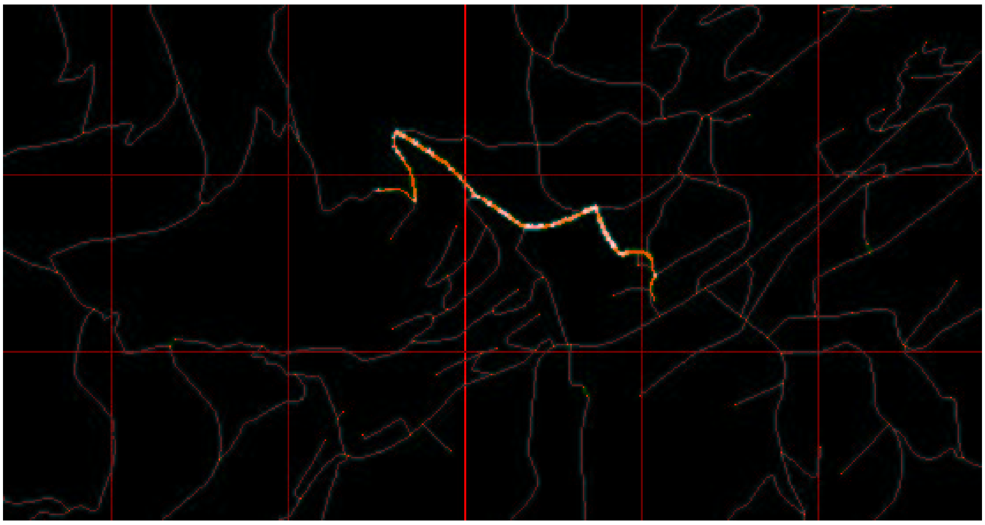


Figure 15: First run. All agents have the same destination, and at this stage, the same route.



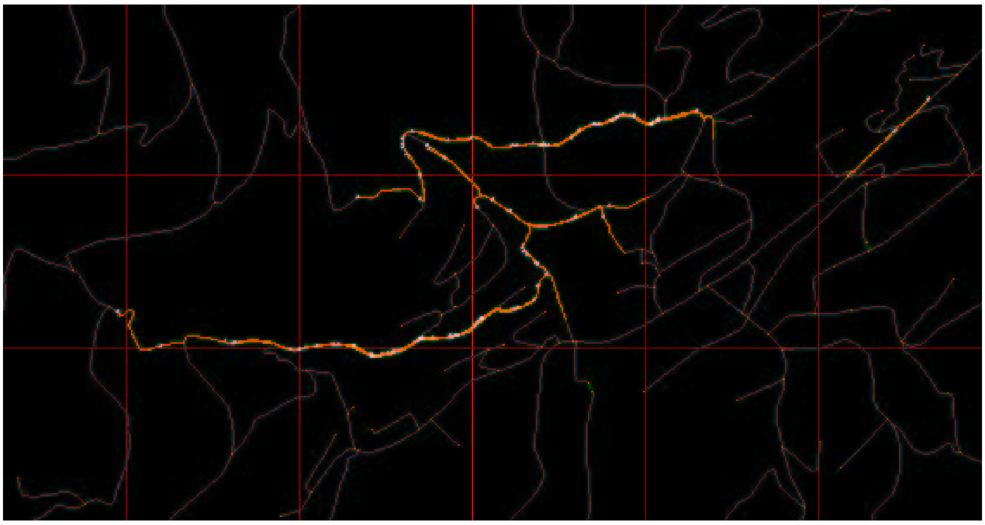
6. A preliminary application

The eventual goal of this project is to simulate hikers in a tourist area in the Swiss Alps, and make them react to visual stimuli. It would go beyond the scope of this paper to discuss the whole idea of that project, and its planned implementation. In general, the approach is similar to an activity-based microsimulation approach to transportation simulation (e.g. Raney *et al.*, 2003b). That is, there will be a **synthetic population**, presumably consisting of tourists staying at hotels or visiting for a day. For each tourist, a daily or possibly weekly **activity plan** will be generated. These activities will include, say, to hike to a certain peak, to have a coffee at a mountain restaurant, or to go shopping in the village. These activities are then connected by **routes**. Routes will be computed according to generalized cost functions, which depend, for example, on scenical beauty or on the number of other hikers that are encountered. A **learning algorithm** on top of this will make the tourists adapt, for example in order to avoid other hikers.

A small proof-of-principle run is documented in Figs. 15, 16, and 17. It is assumed that all agents leave in the morning from the hotel and hike to the same mountain peak. They however want to avoid each other because they want to hike in solitude. Fig. 15 shows the first run, where no hiker knew about the other hikers' intentions. Fig. 16 shows the situation after 50 iterations, where hikers have learned to spread out and avoid each other.

Considerably more work will be necessary to fill these elements with true real-world meaning. This aspect goes, as said before, beyond the scope of this paper.

Figure 16: After 50 iterations



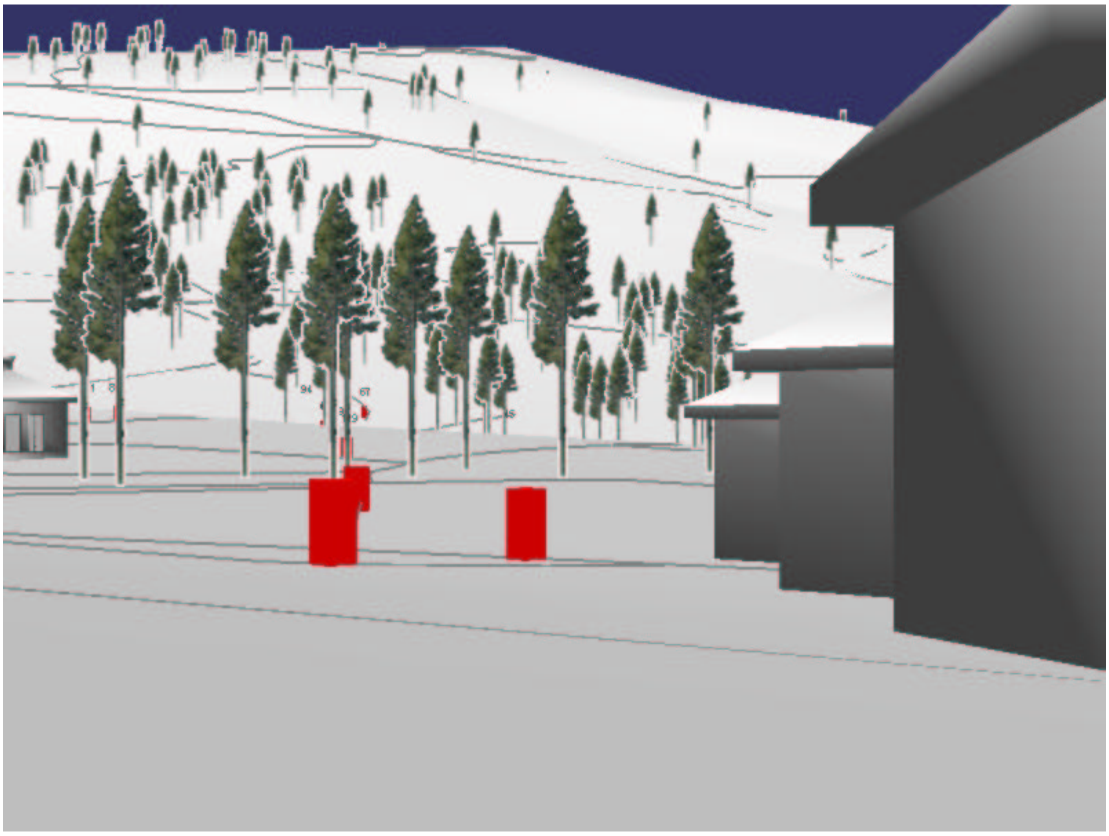
7. Summary

Within the research project "Planning with Virtual Alpine Landscapes and Autonomous Agents", a simulation of hikers in the Alps is implemented. The ultimate goal is to have these hikers make realistic tours, where they react to visual stimuli including the presence of other hikers. This paper presents the underlying pedestrian simulation for the project. Because of the need to allow for realistic arbitrary movement, a simulation based on continuous space was selected. The dynamic model is taken from the literature, but thoroughly tested for our purposes.

One important aspect is the computation of the forces which keep the agent on the path and make it follow the possibly winding path. Two approaches were tested: one based on a combination of location-based forces, which push the agent towards the nearest path, and attractive waypoints, which pull the agent along its route. The other model ("model B") uses a path-oriented coordinate system, with a strong force along the path, and a weak force towards the middle of the path. For the hiking application, model B seems to be easier to handle and to generate fewer artifacts than model A.

In contrast to other pedestrian simulations, which concentrate on crowd or even panic behavior but have the advantage of relatively small spatial scenario sizes, in this project crowd behavior is less important but large scenarios need to be handled. The implementation therefore uses lazy initialization plus caching for location-based data. This means that the simulation area is segmented into blocks. Location-based data is only computed when an agent enters a block for the very first time. It is then simultaneously kept in memory and written to disk. If the simulation runs out of memory, blocks which have not been used for a long time are removed from meory but kept on disk. Such blocks can be re-loaded if another agent enters the block at some later

Figure 17: Some agents hiking from their hotel to the top of a mountain.



time, or if the simulation is run a second time with the same parameters. – Since the hiking paths cover only a relatively small fraction of the whole geographical area, this mechanism makes the handling of large scenarios possible on normal desktop PCs.

Future work will concentrate on “strategic” modules, which compute destinations and routes for the hikers. The intention is to do this in a more general distributed computational architecture, which will allow the simulation of arbitrary large scale mobility scenarios.

Acknowledgments

We thank Duncan Cavens for Fig. 17, and Bryan Raney for the learning mechanism which computed the transition from Fig. 15 to Fig. 16. This project is funded by Swiss National Science Foundation.

References

Akima, H. (1972) A433, interpolation and smooth curve fitting based on local procedures, *Communications of the ACM*, **15** (10) 914–918.

Beazley, D., P. Lomdahl, N. Gronbech-Jensen, R. Giles and P. Tamayo (1995) Parallel algorithms for short-range molecular dynamics, in D. Stauffer (Ed.), *Annual reviews of computational physics III*, 119–176, World Scientific.

Ferber, J. (1999) *Multi-agent systems. An Introduction to distributed artificial intelligence*, Addison-Wesley.

Gingold, R. and J. Monaghan (1977) Smoothed particle hydrodynamics - theory and application to non-spherical stars, *Royal Astronomical Society, Monthly Notices*, **181**.

Helbing, D., I. Farkas and T. Vicsek (2000) Simulating dynamical features of escape panic, *Nature*.

Hoogendorn, S., P. Bovy and W. Daamen (2002) Microscopic pedestrian wayfinding and dynamic modelling, in M. Schreckenberg and S. Sharma (Eds.), *Pedestrian and Evacuation Dynamics*, 123–154.

Krauß, S. (1997) Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics, Ph.D. thesis, University of Cologne, Germany. See www.zpr.uni-koeln.de.

Mauron, L. (2002) Pedestrians simulation methods, Diploma thesis, Swiss Federal Institute of Technology ETHZ.

Meyer-König, T., H. Klüpfel and M. Schreckenberg (2001) Assessment and analysis of evacuation processes on passenger ships by microscopic simulation, in M. S. et al (Ed.), *Pedestrian and Evacuation Dynamics*, 297–302, Springer.

Raney, B., N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen and K. Nagel (2003a) An agent-based microsimulation model of Swiss travel: First results, *Networks and Spatial Economics*, **3** (1) 23–41.

Raney, B., N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen and K. Nagel (2003b) An agent-based microsimulation model of Swiss travel: First results, *Networks and Spatial Economics*, **3** (1) 23–41.

Weidmann, U. (1992) Transporttechnik der Fussgänger, ***Schriftenreihe des IVT***, IVT ETH Zürich.

Wolfram, S. (1986) *Theory and Applications of Cellular Automata*, World Scientific, Singapore.