A Study of Step Calculations in Traffic Cellular Automaton Models

Minjie Chen Günter Bärwolff Hartmut Schwandt

***Abstract*—We give some formal description for the cellular automaton (CA) models applied in the traffic simulation of vehicular and pedestrian dynamics in both one- and two-dimensional cases. In the two-dimensional case, we present a new solution for the step choice problem $v_{\max} > 1$ on the local operational level, with the aim of projecting the intended step exactly onto the underlying geometry. This method can be used in combination with the more advanced modeling techniques on higher levels for better simulation results.**

I. Introduction

The last two decades witnessed substantial progresses in the modeling and simulation of traffic and pedestrian dynamics. Cellular automaton (CA) is an important category of the modeling methods [6]. Its basic idea is that after mapping the physical geometry onto a set of grid cells, we can employ the state change of the cells to describe the dynamical aspects of the traffic/pedestrian flow.

In the single-lane traffic simulation, many CA models traced back to [5] (deterministic) and [11, 13] (probabilistic). The single-lane model was extended later by [4] to include the situation of multiple lanes. In contrast, pedestrian dynamics have been studied mostly as a two-dimensional problem, due to the lack of a unanimous flow direction in the general case. The ansatz proposed by [2, 9] successfully reproduced many human behavioral characteristics. However, these models do not provide us with clearly defined local transition rules for the CAs. A major purpose of our paper is to fill this gap and we will see that the so-called conflict solution (which is usually to be given attention separately) can be integrated into the transition rules. In comparison to the $v_{\max} = 1$ model [2, 9] (where the step size in each simulation cycle was prescribed to be of single-cells), we propose in the second part of this text a new method for the calculation of steps $v_{\max} \geq 1$ on the operational level. (Here we apply the common notation for velocity v which is counted as the number of grid cell length/width in the spatial setup, without unit.)

For a good overview of CA, we refer to [14] and [8]. For an overview of CAs in traffic modeling, please see [10].

II. Notation

An n-dimensional CA is a tuple (L^n, S, N, δ), where L^n is the index set of grid cells by which the spatial setup is addressed, S the set of cell states, $N = (\vec{x}_1 \ldots, \vec{x}_{|N|})$ is a vector of pairwise different elements in L^n called neighborhood (indicating position disparity in the spatial setup) and $\delta : S^{|N|} \to S$ the transition function (also called local transition rules). The elements of N will be called neighborhood patterns. When the CA is spatially infinite, L can be written as $\mathbb{Z}$ (or $\mathbb{N}$) explicitly.

A one-dimensional CA is called elementary when there are exactly two states and the neighborhood is of radius 1, we write this as $S = \{\mathbf{0}, \mathbf{1}\}$ and $N = (-1, 0, 1)$. (Since $n = 1$, the vector element $\vec{x} \in L^n$ degenerates into a scalar in L.) There are $|S|^8 = 2^8 = 256$ different elementary CAs in all, since $|N| = 3$ and there are $|S|^3 = 2^3 = 8$ different neighborhood patterns to be mapped into a new state in S by δ. Applying a notation similar to that of [14], the local transition rule $\delta : \{\mathbf{0}, \mathbf{1}\}^3 \to \{\mathbf{0}, \mathbf{1}\}$ can be expressed as

$$\begin{array}{ccc} \mathbf{l} & \mathbf{c} & \mathbf{r} \\ & \mathbf{s} & \end{array}$$

with **l**, **c**, **r** representing the current states in the three grid cells ("left", "center" and "right") which build up the neighborhood pattern. **s** denotes the new state after the transition (also called update) in the center cell.

Compiling the transition rules altogether, [14] suggested the following encoding schema for the elementary CA

1 1 1		**1 1 0**		**1 0 1**		**1 0 0**	
	$\mathbf{s_7}$		$\mathbf{s_6}$		$\mathbf{s_5}$		$\mathbf{s_4}$
0 1 1		**0 1 0**		**0 0 1**		**0 0 0**	
	$\mathbf{s_3}$		$\mathbf{s_2}$		$\mathbf{s_1}$		$\mathbf{s_0}$

Associated with the enumerated neighborhood patterns, the new states can be compressed into a binary string and further interpreted as a binary number $(\mathbf{s_7}\,\mathbf{s_6}\,\mathbf{s_5}\,\mathbf{s_4}\,\mathbf{s_3}\,\mathbf{s_2}\,\mathbf{s_1}\,\mathbf{s_0})_2$. Accordingly, each of the 256 elementary CAs can be identified by a distinct integer from 0 to 255.

III. One-Dimensional Case

In the context of one-dimensional traffic modeling, **1** can be used to indicate that a simulation object is present at a certain location, whereas **0** indicates the "free" state of a location ("unblocked" for the other objects). Without loss of generality, the moving direction of the simulation objects is assumed to be from left to right (i. e. in the positive order of L). In general, the one-dimensional models deal with the following problems.

Manuscript created April 19, 2010; revised July 9, 2010.

The authors are with the Institut für Mathematik, Technische Universität Berlin, Germany. Emails: {minjie.chen, baerwolf, schwandt}@math.tu-berlin.de

A. Single-Cell Step Choice

The single-cell step choice can be simulated by the following elementary CA

$$\begin{array}{cccc} \mathbf{1\ 1\ 1} & \mathbf{1\ 1\ 0} & \mathbf{1\ 0\ 1} & \mathbf{1\ 0\ 0} \\ \mathbf{\ \ 1} & \mathbf{\ \ 0} & \mathbf{\ \ 1} & \mathbf{\ \ 1} \\ \mathbf{0\ 1\ 1} & \mathbf{0\ 1\ 0} & \mathbf{0\ 0\ 1} & \mathbf{0\ 0\ 0} \\ \mathbf{\ \ 1} & \mathbf{\ \ 0} & \mathbf{\ \ 0} & \mathbf{\ \ 0} \end{array}$$

encoded as $(\mathbf{10111000})_2$ (named in [14] as Rule 184). The transition can be interpreted in the following way: a grid cell in current state $\mathbf{1}$ retains this state if and only if its neighboring cell on the right side has the state $\mathbf{1}$ (i. e. the forward step is impossible), and a cell in current state $\mathbf{0}$ retains this state if and only if its neighboring cell on the left side has the state $\mathbf{0}$ (i. e. no follower is present).

If the choice of the states $\mathbf{0}$ and $\mathbf{1}$ does not affect the transition rule, we may denote this by a meta-state $-$ (also called "don't care"). This enables us to rewrite the above CA as

$$\begin{array}{cccc} -\ \mathbf{1\ 1} & -\ \mathbf{1\ 0} & \mathbf{1\ 0}\ - & \mathbf{0\ 0}\ - \\ \mathbf{\ \ 1} & \mathbf{\ \ 0} & \mathbf{\ \ 1} & \mathbf{\ \ 0} \end{array}$$

B. Multiple-Cell Step Choice

To describe a multiple-cell step choice, we will need a larger neighborhood (in terms of the size of neighborhood patterns). Obviously, the transition rule of an r-cell step choice requests a neighborhood of radius no smaller than r ($r \in \mathbb{N}^+$). Let the neighborhood radius be r.

1) Variant 1: We first consider the case of strictly r cells, i. e. a step of r cells will be carried out if all the r cells concerned in the moving direction are in state $\mathbf{0}$.

For a compact notation, we use the superscript i to address a repetition of exactly i times of a certain state. A successful r-cell step at a location in state $\mathbf{1}$ takes place exactly when

$$\begin{array}{c} -^r\ \mathbf{1}\ \mathbf{0}^r \\ \quad\ \mathbf{0} \end{array}$$

otherwise we have

$$\begin{array}{c} -^r\ \mathbf{1} \\ \quad\ \mathbf{1} \end{array}$$

(The empty partial neighborhood pattern on the right side refers to the negation of the pattern $\mathbf{0}^r$, with $m+n=r-1$, $m,n \in \mathbb{N}$ this can be alternatively expressed as $-^m\ \mathbf{1}\ -^n$ which prescribes that at least one $\mathbf{1}$ comes up in the pattern.)

Similarly, at a location in state $\mathbf{0}$ a follower comes up exactly when

$$\begin{array}{c} \mathbf{1}0^{r-1}\mathbf{0}\ -^r \\ \quad\ \mathbf{1} \end{array}$$

i. e. the follower is in the correct position and there is no hindrance for this step, otherwise we have

$$\begin{array}{c} \mathbf{0}\ -^r \\ \mathbf{0} \end{array}$$

(The negation of the partial neighborhood pattern $\mathbf{1}\ \mathbf{0}^l$ on the left side is composed of $\mathbf{0}^r$ and $-^m\ \mathbf{1}\ -^n$ with $m+n=r-1$, $m \in \mathbb{N}^+$, $n \in \mathbb{N}$.)

For $r = 2$, after a somehow tedious enumeration, the local transition rules can be encoded as $(\mathbf{11100000111011111110000011100000})_2$ starting with the neighborhood pattern $\mathbf{1\ 1\ 1\ 1\ 1}$ as the highest bit and descending to $\mathbf{0\ 0\ 0\ 0\ 0}$ as the lowest bit. We notice there are 2^{2r+1} (here 32) neighborhood patterns to be defined in the transition function δ.

2) Variant 2: Another variant is that the multiple-cell step should be carried out as far as possible. This can be understood as conflict avoidance in the moving direction as well. The simulation for this is generally impossible for synchronous transition with $|S| = 2$. This is because sometimes the state change in the current neighborhood pattern depends on the information from outside, regardless of the neighborhood size. A simple example would be

$$-^r\ \mathbf{1}\ \mathbf{1}^r$$

In a synchronous transition, the actual move of the simulation object in the center depends on the moves of its neighbors on the right side, which are further dependent of their own neighbors on the right side. (On the other hand, applying a sequential update on the cells, for example, in the order from right to left, may serve the purpose.)

However, we can solve this problem by introducing a larger state set, see the discussion in III-E.

C. Probabilistic Step Choice

Step choice of this kind prescribes rules in a way like "perform a certain action with a probability $p \in (0,1)$" (we exclude the trivial cases $p = 0, 1$). We notice that rules of this kind refer to simulation objects only, since a certain action is to be performed probabilistically; in other words, this concerns the locations in state $\mathbf{1}$. We wish to point that it is generally inappropriate to implement a probabilistic step by applying the local transition rules independently on the cells with the given probability. This can result in collision and undefined situations.

On the other hand, a probabilistic action should always be associated with a non-free state (which further differentiates from the usual $\mathbf{1}$ and $\mathbf{0}$ states) and only in this state the local transition occurs with the prescribed probability. Our suggestion is to introduce a new state $\mathbf{q}$ and disintegrate the transition into three substeps. First, all the cells in state $\mathbf{1}$ undergo a probabilistic transition

$$\mathbf{1} \to \begin{cases} \mathbf{1} & \text{with probability } p, \\ \mathbf{q} & \text{with probability } 1-p. \end{cases} \tag{1}$$

Next, the usual local transition rules are to be applied on the neighborhood patterns. For the normal states $\mathbf{1}$ and $\mathbf{0}$, no further attention will be needed. For neighborhood patterns containing the new state $\mathbf{q}$, we consider the following two situations. If the center cell is in state $\mathbf{q}$, the transition should

be understood as if this location were in the negative state $\mathbf{0}$, so that the possible action should not be carried out. Otherwise (on the left or the right side of the neighborhood), it should be replaced by the positive state $\mathbf{1}$, since the probabilistic nature of the action in question would disturb other simulation objects in the synchronous update procedure. In formal expression,

$$-^r\ \mathbf{q}\ -^r \quad \longrightarrow \quad -^r\ \mathbf{0}\ -^r \tag{2}$$

$$-^m\ \mathbf{q}\ -^n\ \mathbf{0}\ -^r \quad \longrightarrow \quad -^m\ \mathbf{1}\ -^n\ \mathbf{0}\ -^r \tag{3}$$

$$-^r\ \mathbf{0}\ -^m\ \mathbf{q}\ -^n \quad \longrightarrow \quad -^r\ \mathbf{0}\ -^m\ \mathbf{1}\ -^n \tag{4}$$

with $m+n=r-1$, $m,n\in\mathbb{N}$. The "don't care" state $-$ is now adapted to include the new state $\mathbf{q}$ in addition. Finally, the simulated free state $\mathbf{0}$ in (2) should be recovered into the normal state $\mathbf{1}$

$$-^r\ \mathbf{0}\ -^r \quad \longrightarrow \quad -^r\ \mathbf{1}\ -^r \tag{5}$$

D. Acceleration and Deceleration

Another frequently encountered problem is acceleration and deceleration. In the context of CA modeling, it is to be understood as integer velocity changes. Let $v_{\max}\in\mathbb{N}^+$ denote the maximal attainable velocity, then the possible velocities can be $0,1,\ldots,v_{\max}$.

To this end, we need a larger set of $v_{\max}+2$ cell states. The new set of states includes a new free state $\boldsymbol{\epsilon}$ which corresponds to the state $\mathbf{0}$ in the previous paragraphs and a series of $v_{\max}+1$ states $\mathbf{0},\ldots,\mathbf{v}^*$ which correspond to the velocities from 0 to $v_{\max}$ respectively. ($\mathbf{0}$ now refers to the state in velocity 0 instead of being free.) Obviously, a location in free state $\boldsymbol{\epsilon}$ retains its state and once the highest velocity is reached, the velocity cannot be raised,

$$\begin{array}{ccc} & \boldsymbol{\epsilon} & \\ -^r & & -^r \\ & \boldsymbol{\epsilon} & \end{array} \qquad \begin{array}{ccc} & \mathbf{v}^* & \\ -^r & & -^r \\ & \mathbf{v}^* & \end{array}$$

In between, for a state $\mathbf{v}$ associated with velocity $v<v_{\max}$, the increment of one unit in velocity results in a new state $\mathbf{v}^+$ associated with velocity $v+1$,

$$\begin{array}{ccc} & \mathbf{v} & \\ -^r & & -^r \\ & \mathbf{v}^+ & \end{array}$$

Again, $-$ is the new "don't care" state including all the possible states $\boldsymbol{\epsilon},\mathbf{0},\ldots,\mathbf{v}^*$.

Deceleration can be treated in an analogous way. For a location in free state or stationary state (i. e. with velocity 0)

$$\begin{array}{ccc} & \boldsymbol{\epsilon} & \\ -^r & & -^r \\ & \boldsymbol{\epsilon} & \end{array} \qquad \begin{array}{ccc} & \mathbf{0} & \\ -^r & & -^r \\ & \mathbf{0} & \end{array}$$

Otherwise, for a state $\mathbf{v}\in\{\mathbf{1},\ldots,\mathbf{v}^*\}$,

$$\begin{array}{ccc} & \mathbf{v} & \\ -^r & & -^r \\ & \mathbf{v}^- & \end{array}$$

with $\mathbf{v}^-$ denoting the state corresponding to the velocity $v-1$.

E. Multiple-Cell Step Choice Revisited

We notice that the velocity change can be further interpreted as the change of step lengths in the simulation cycle. Thus, the second variant of the multiple-cell step choice can be implemented in a similar way. The implementation takes two substeps. The first substep is a synchronous update to determine the actual possible length for every potential move and the second is the execution of the new step choice.

Similar to the situation of acceleration or deceleration, let $\boldsymbol{\epsilon}$ denote the free state and $\mathbf{0},\ldots,\mathbf{r}$ the states of actual steps of the lengths $0,\ldots,r$, respectively. Let $-$ be the new "don't care" meta-state including all states and $\mathbf{l}$ the meta-state including all non-free states $\mathbf{0},\ldots,\mathbf{r}$, then the new actual step can be determined by

$$\begin{array}{ccc} & \boldsymbol{\epsilon} & \\ -^r & & -^r \\ & \boldsymbol{\epsilon} & \end{array}$$

concerning the free state and

$$\begin{array}{ccc} & \mathbf{l} & \\ -^r & & \boldsymbol{\epsilon}^m\ \mathbf{l}\ -^n \\ & \mathbf{m} & \end{array}$$

with $m+n=r-1$, $m,n\in\mathbb{N}$. m is the maximum number of successive free states starting from the right side of the current position in the neighborhood pattern, which is also the possible step length at the current position. The result is a new state $\mathbf{m}$ associated with the length m. In the second substep, the state $\mathbf{m}$ prescribes the actual step size in the current simulation cycle.

IV. Two-Dimensional Case

Unlike vehicular traffic where a dominant flow direction is present, pedestrian dynamics are usually modeled in two dimensions. In microscopic pedestrian simulation models, [7] suggested the separation of pedestrians' behaviors on three different levels. These are a strategic level (choice of activity pattern), a tactical level (for activity scheduling and route choice) and an operational level (for local and temporary actions). Once the pedestrian activity decision is made on the other two levels, step choices are computed and executed on the operational level.

Typical models like [2, 9] dealt with the case $v_{\max}=1$, i. e. the move is limited to the immediate neighbors in the simulation cycle. Sometimes diagonal moves (of one grid cell in both dimensions) are allowed. Despite its simplicity and clarity, this method introduces a much too large difference between the intended and actual step lengths. (A move to the relative position $(1,1)$, which is of physical length $\sqrt{2}$, is of length 2, if the diagonal move is not allowed, otherwise, 1.)

On the other hand, it appears that we should not confine the step choice to be single-cell ones. A plausible computing model for multiple-cell step choices should, in return, allow us not to limit the grid cell size to be fixed in the spatial setup a priori (which has been commonly assumed to be $0.4\mathrm{m}\cdot 0.4\mathrm{m}$), if necessary. The necessity of a flexible (of course, within a

reasonable range) cell size becomes self-evident, if we wish to describe situations of (extremely) high and low object densities in the models. For further discussion, we refer to our earlier work [3].

For multiple-cell steps, i. e. $v_{\max} \geq 1$, it is possible to apply J. E. Bresenham's algorithm of line rastering [1]. This algorithm approximates the intended step in the two-dimensional spatial setup (in most cases, a regular grid) and the actual step is executed as a sequence of substeps. The intended step is considered as a vector $(\Delta x, \Delta y) \in \mathbb{N}^2$, representing the position differences in the two dimensions. The step calculation is carried out in the dimension with the larger difference. By looking for the closest position to the vector $(\Delta x, \Delta y)$, we decide whether a position change in the other dimension is needed. Obviously, in nontrivial cases we encounter diagonal substeps. This method is, however, not exactly what we want in two-dimensional simulation of pedestrian dynamics, since a diagonal substep is considered to be of equal length with a usual substep in just one of the dimensions. In other words, a series of substeps including diagonal moves may represent a much larger real physical step length than a series of substeps in exactly one dimension of the same number. Our concern is to minimize the aforementioned diagonal length discrepancy.

Now in the context of pedestrian dynamics, we consider a state set $S = \{\boldsymbol{\epsilon}, \mathbf{0}, \mathbf{1}\}$ composed of a free state $\boldsymbol{\epsilon}$, a blocking state $\mathbf{0}$ (induced by stationary hindrances) and a dynamical blocking state $\mathbf{1}$ (manifesting the presence of a mobile simulation object). Analogous to the discussion in the previous paragraphs, if we split a potential step of $v_{\max} > 1$ into single-cell substeps (or substeps of a smaller size), we only have to consider the cell states in the (relatively) small neighborhood in the cell state transitions. The Moore neighborhood of radius 1 has the form

	1	

On the other hand, we notice that on the operational level, certain route plan information (computed on the other higher levels, according to [7]) is assigned to the moving objects. For such an object, in fact only the information in its partial neighborhood, e. g.

1	

is substantial. Now we would like to construct a series of single-cell substeps which should rebuild the intended step choice in good approximation.

Let x and y denote the horizontal and vertical dimension respectively. We consider an intended step of $(\Delta x, \Delta y)$. Without loss of generality, we request $\Delta x \in \mathbb{N}^+$, $\Delta y \in \mathbb{N}$. We disintegrate this multiple-cell step into a series of substeps.

A possible substep can be $(1, 0)$ (move in the x-direction), $(0, 1)$ (move in the y-direction) or $(1, 1)$ (move in both directions). These possible substep choices are given the probabilities $p_x, p_y, p_{xy} \in [0, 1]$. Naturally, there holds

$$p_x + p_y + p_{xy} = 1,$$

in addition, we request

$$p_x \cdot \Delta y = p_y \cdot \Delta x,$$

since we can assume that the choice to take a substep in exactly one dimension is also dependent of the components Δx and Δy of the intended step. We can imagine that the larger the step in one dimension is, the more likely a substep should be undertaken in that direction.

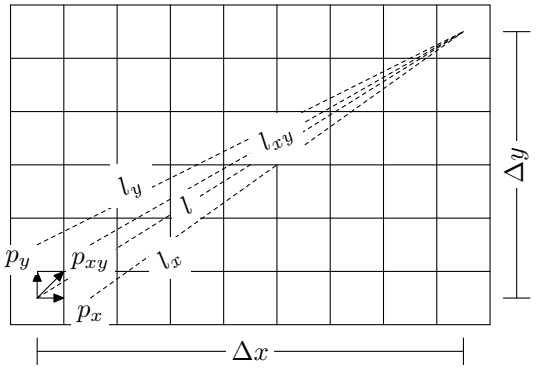

Fig. 1. Substep choices. The step $(\Delta x, \Delta y)$ has a length of l; the remaining lengths after the three possible substeps are shown as l_x, l_y and l_{xy} respectively.

Next, we establish a further relationship among p_x, p_y and p_{xy}. Obviously, a diagonal substep covers considerable more spatial distance than the other two choices. We consider the physical lengths from the position $(\Delta x, \Delta y)$ to the current position (l), to the position after a substep in the x-direction (l_x), to the position after a substep in the y-direction (l_y) and to the position after a substep in both directions (l_{xy}), which can be written explicitly as

$$\begin{aligned} l &= \sqrt{(\Delta x)^2 + (\Delta y)^2}, \\ l_x &= \sqrt{(\Delta x - 1)^2 + (\Delta y)^2}, \\ l_y &= \sqrt{(\Delta x)^2 + (\Delta y - 1)^2}, \\ l_{xy} &= \sqrt{(\Delta x - 1)^2 + (\Delta y - 1)^2}. \end{aligned} \tag{6}$$

Since we wish to make the number of the required substeps as close as possible to the physical length of the intended step choice, we may request

$$p_x(1 + l_x) + p_y(1 + l_y) + p_{xy}(1 + l_{xy}) = l. \tag{7}$$

See Fig. 1. The solution of (7) is

$$\begin{aligned} p_x &= \frac{l - l_{xy} - 1}{l_x + \frac{\Delta y}{\Delta x} l_y - \left(1 + \frac{\Delta y}{\Delta x}\right) l_{xy}}, \\ p_y &= \frac{\Delta y}{\Delta x} p_x, \end{aligned}$$

and if $\Delta y > 0$,

$$p_y = \frac{l - l_{xy} - 1}{l_y + \frac{\Delta x}{\Delta y} l_x - \left(1 + \frac{\Delta x}{\Delta y}\right) l_{xy}}, \tag{8}$$
$$p_{xy} = 1 - p_x - p_y.$$

In the next substep, we update Δx and Δy according to the last substep execution, calculate the new probabilities p_x, p_y and p_{xy} and make a decision according to these. This procedure will be repeated until the destination is reached ($\Delta x = \Delta y = 0$).

In the trivial case $\Delta x = \Delta y = 1$, (8) renders $p_x = p_y = \frac{\sqrt{2}-1}{2}$, $p_{xy} = 2-\sqrt{2}$. In the other trivial case $\Delta x = 1$, $\Delta y = 0$, the solution is $p_x = 1$, $p_y = p_{xy} = 0$, which meets the expectation of a simple step in the x-direction exactly. Table I lists a set of solutions of p_x in float numbers for the most usual combinations of Δx and Δy. In software implementation, a similar lookup table can be employed in the program to enhance efficiency.

TABLE I
LOOKUP TABLE OF THE SOLUTION p_x FOR SOME COMMON STEP CHOICE CONFIGURATIONS.

	$\Delta x = 1$	$\Delta x = 2$	$\Delta x = 3$	$\Delta x = 4$	$\Delta x = 5$
$\Delta y = 0$	1	1	1	1	1
$\Delta y = 1$	0.207	0.258	0.285	0.299	0.306
$\Delta y = 2$	0.129	0.252	0.305	0.335	0.354
$\Delta y = 3$	0.095	0.204	0.267	0.306	0.334
$\Delta y = 4$	0.075	0.168	0.230	0.273	0.305
$\Delta y = 5$	0.061	0.142	0.200	0.244	0.278

Technical Note 1: Obviously, (7) assumes that the substep choice is indeed possible, i. e. the relevant cell positions are in the free state ϵ. If this requirement is only partially met, the relationship (7) should be modified according to the available substep options to deliver solutions similar to (8). In fact, all combinations of the cell states in positions $(1, 0)$, $(0, 1)$ and $(1, 1)$ (respecting positive Δx and Δy) should be investigated. For example, in case the vertical substep move is impossible, we have

$$p_x + p_{xy} = 1,$$
$$p_x(1 + l_x) + p_{xy}(1 + l_{xy}) = l.$$

The solution (8) becomes

$$p_x = \frac{l - l_{xy} - 1}{l_x - l_{xy}},$$

and since $p_{xy} = 1 - p_x$,

$$p_{xy} = \frac{l_x - l + 1}{l_x - l_{xy}}.$$

This also solves the conflict problem on a local basis.

Technical Note 2: In the previous section of the one-dimensional case we have discussed the implementation (see (1)–(5)) of probabilistic step choices. A similar strict implementation in the two-dimensional case is also possible. However, in view of notational complexity, it suffices to give a well-defined schema for the calculation of the substep probabilities.

Experimental Results

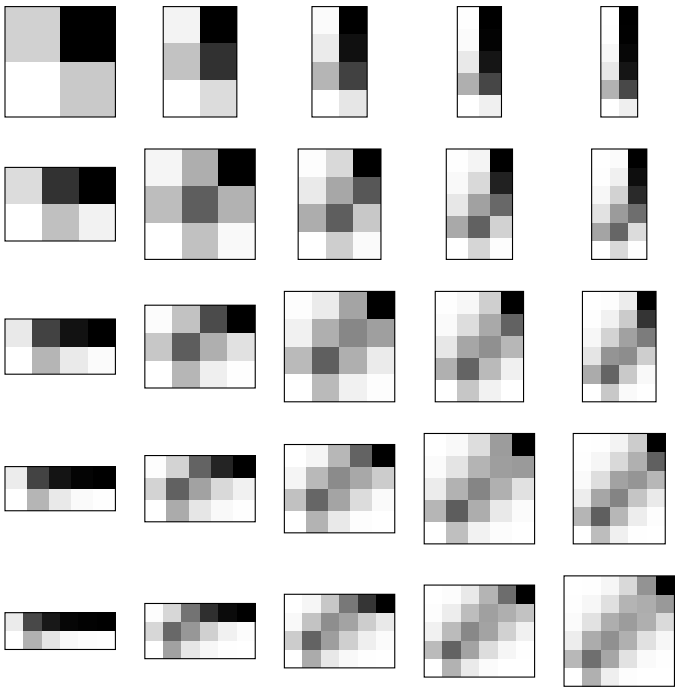

Fig. 2. The cell position's occurrences in the substep choices for different combinations of $\Delta x, \Delta y = 1, \ldots, 5$. The cell with a relative position $(\Delta x, \Delta y)$ is always drawn in black which indicates its contribution as the final substep in all configurations.

In Fig. 2 some of the experimental results are listed. For a combination of Δx and Δy, the empirical occurrences of the grid cell positions in the relevant substep choice for $(\Delta x, \Delta y)$ are shown in a linear grayscale. The white and black color refers to no occurrence and definite occurrence (i. e. with a probability of 1) respectively. The start position $(0, 0)$ is left empty, since it is not to be counted as a substep in the execution. In each of the configurations, the overall appearance (in color) of the cells forms a virtual trail of the intended step.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we considered the step choice problem in one-dimensional and two-dimensional CA models for the simulation of vehicular and pedestrian dynamics. Especially in the two-dimensional case, we are of the opinion that a step size larger than one grid cell in the simulation cycle should generally be allowed in the CA modeling. Due to the nature of the spatial setup of the CAs, larger step sizes inevitably introduce diagonal moves which are of significantly larger physical lengths than steps in the horizontal or vertical

directions exclusively. By applying some simple techniques of probabilistic substep calculations, we presented a new constructive method for the solution of step choice problem with good approximation results. Although the proposed method is merely aimed at the computations on the lowest simulation level, it can be used in combination with arbitrary modules on the higher levels (e. g. with route choice, user-defined pedestrian characteristics/behavioral patterns).

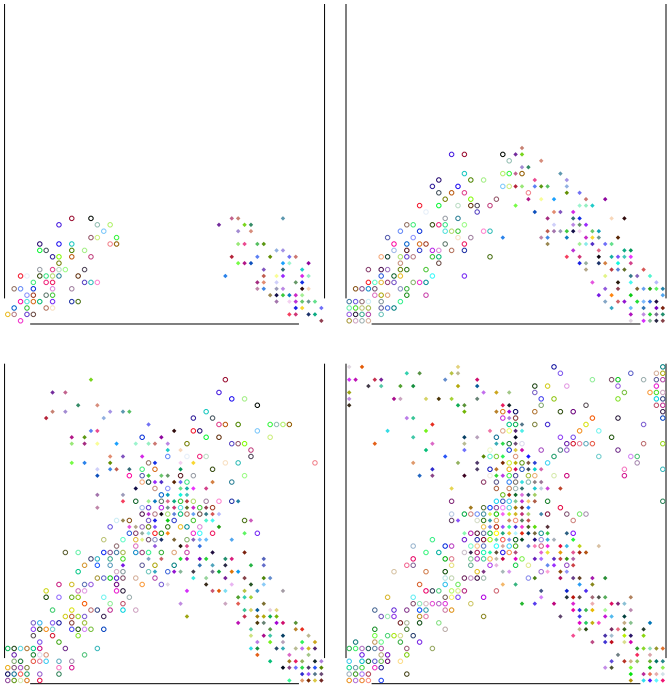

Fig. 3. Simulation example showing two intersecting pedestrian streams, recorded at different time points.

In Fig. 3 we provide a very rudimentary simulation example in demonstration of our method. The spatial setup is a rectangular area. At the two bottom corners of this area we produce two groups of pedestrians (drawn as unfilled circles and filled rhombuses in different colors) on a random basis. The individual pedestrians are given very simple instructions as route strategy namely a fixed moving direction and the pedestrians will be dismissed once they reach the other side of the rectangular region. In the step executions, local conflicts are to be avoided (see Technical Note 1). For repeated unsuccessful step executions, the pedestrians in question are given the freedom to change or adjust their moving directions temporarily to bypass the obstacles. The simulation result is roughly satisfying for this simple test case. With additional computation modules on the higher simulation levels, we believe that more convincing results in complicated situations can be achieved, this will be our future work too. Another focus in our future work is the high density situations, we wish to understand and describe the typical behavioral patterns of pedestrians of large number and possibly with different activity patterns, by means of empirical data (e. g. video recordings of crowds), and use them to construct computing modules on higher levels for the simulation of more complicated scenarios, especially when pedestrians form multiple intersecting flows and the interacting aspects of the flows that disturb one another need to be investigated.

VI. Acknowledgment

The authors gratefully acknowledge the support of Deutsche Forschungsgemeinschaft (German Research Foundation) for the project SCHW548/5-1+BA1189/4-1.

References

[1] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[2] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A*, 295:507–525, 2001.

[3] M.-J. Chen, G. Bärwolff, and H. Schwandt. Automaton model with variable cell size for the simulation of pedestrian flow, 2008. An electronic version can be retrieved at: http://www.math.tu-berlin.de/~chenmin/pub/cbs080331.pdf (accessed July 9, 2010).

[4] J. Esser and M. Schreckenberg. Microscopic simulation of urban traffic based on cellular automata. *Int. J. of Mod. Phys. C*, 8(5):1025–1036, 1997.

[5] M. Fukui and Y. Ishibash. Traffic flow in 1D cellular automaton model including cars moving with high speed. *Journal of the Physical Society of Japan*, 65(6):1868–1870, 1996.

[6] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, 73:1067–1141, 2001.

[7] S. P. Hoogendoorn and P. H. L. Bovy. Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B*, 38:169–190, 2004.

[8] J. Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334:3–33, 2005.

[9] A. Keßel, H. Klüpfel, J. Wahle, and M. Schreckenberg. *Microscopic simulation of pedestrian crowd motion*, pages 193–200. In Schreckenberg and Sharma [12], 2002. ISBN 978-3-540-42690-5.

[10] S. Maerivoet and B. De Moor. Cellular automata models of road traffic. *Physics Reports*, 419:1–64, 2005.

[11] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *J. Phys. I France*, 2:2221–2229, 1992.

[12] M. Schreckenberg and S. D. Sharma, editors. *Pedestrian and Evacuation Dynamics*. Springer-Verlag Berlin Heidelberg, 2002. ISBN 978-3-540-42690-5.

[13] M. Schreckenberg, A. Schadschneider, K. Nagel, and N. Ito. Discrete stochastic models for traffic flow. *Physical Review E*, 51(4):2939–2949, 1995.

[14] S. Wolfram. *A New Kind of Science*. Wolfram Media, Inc., 2002. ISBN 1-57955-008-8.