

A Simulation Tool for Modelling Pedestrian Dynamics during Evacuation of Large Areas

Ioakeim G. Georgoudas, Georgios Ch. Sirakoulis,
and Ioannis Th. Andreadis
Democritus University of Thrace, Department of Electrical and Computer
Engineering,
Laboratory of Electronics,
GR 67100 Xanthi, Greece
{igeorg, gsirak, iandread}@ee.duth.gr
<http://www.ee.duth.gr/people/frame.htm>

Abstract. The movement of large numbers of people is important in many situations, such as the evacuation of a building in an emergency. In this paper, pedestrian dynamics during the evacuation of large areas is simulated using a computational intelligent technique, based on Cellular Automata. The characteristic feature of the proposed model is that the crowd consists of independent parts rather than treated as homogeneous mass. The crowd behaviour is artificially formatted by the response of each of these parts to the rule according to which each pedestrian reaches one of the possible exits. Furthermore, an efficient graphical user interface has been developed, in order to study various hypotheses concerning the pedestrians' activity features. Collisions among pedestrians have been encountered while collective effects prominent at crowd behaviour have been also realised during simulation. Finally, the presence of fixed as well as user-defined moveable obstacles has been taken into account.

1 Introduction

When we are at a major sporting event or travelling on public transport or shopping around in shopping precincts, our safety and comfort depend crucially on our fellow crowd members and on the design and operation of the facility we are in. Thus it is unnerving to realize that the modelling currently used to design and operate these venues has more in common with the design of water-pipe networks than anything with a human dimension. As a result the need for a different and potentially far more realistic crowd behaviour modelling approach is of great importance.

Please use the following format when citing this chapter:

Georgoudas, Ioakeim, Sirakoulis, Georgios, Andreadis, Ioannis, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 618–626

During the last decade, it has been adopted the approach of modelling a crowd composed of discrete individuals rather than being faced as a homogeneous mass that behaves like a flowing fluid [1]. In the past, one of the main constraints for such an approach was the enormous number of calculations required to be solved for each crowd member. Fortunately, modern computer power has changed this situation. An additional step towards further simplification of modelling such processes can be the introduction of computational intelligent techniques such as Cellular Automata (CA) [2-3].

As far as it concerns particularly the simulation of pedestrian dynamics, CA models of generally two-dimensional (2-d) nature have been reported in literature [4-5]. Some of them treat pedestrians as particles subject to long-range forces [5] and others use walkers leaving a trace by modifying the underground on their paths [4]. More-over, generalisations of the city traffic model have been proposed mainly focusing on the occurrence of a jamming transition as the density of pedestrians is increased [4, 6]. The majority of the models reported, adopted as the most suitable update procedure the parallel update.

In this paper a user-friendly parameterized model attempts at further and more detailed investigation of pedestrian dynamics focusing on specific rather than generalised behaviour under certain conditions. To this direction, the user is provided with the ability to predefine distinguishing features of the evacuated area, incorporating both topological-oriented parameters and parameters that describe the crowd formation. In particular, the extent of the area, the presence as well as the exact location of obstacles, the number and the allocation of the exits are easily defined. Moreover, as far as it concerns crowd features, there are options regarding the population, crowd behaviour, velocity as well as the existence or not of different types of individuals (children or elderly and middle-aged). It should be reminded that all crowd characteristics are individual dependent. Consequently, the existence of several, externally defined parameters enhances the surveillance perspectives of a certain area. General characteristics are confined, indicating thus certain characteristics allowing the implementation of more effective monitoring techniques. Finally, data obtainable by video monitoring can be also supplied to the model in order to realize in what extent is the model capable of reproducing various observed phenomena under panic circumstances.

2 Mathematical definition of the proposed technique

In this section a more formal definition of a CA will be presented [7]. In general, a CA requires:

1. A regular lattice of cells covering a portion of a d-dimensional space;
2. A set $\mathbf{C}(\vec{r}, t) = \{C_1(\vec{r}, t), C_2(\vec{r}, t), \dots, C_m(\vec{r}, t)\}$ of variables attached to each site \vec{r} of the lattice giving the local state of each cell at the time $t = 0, 1, \dots$;
3. A rule $\mathbf{R} = \{R_1, R_2, \dots, R_m\}$ which specifies the time evolution of the states $\mathbf{C}(\vec{r}, t)$ in the following way:

$$C_j(\vec{r}, t+1) = R_j(C(\vec{r}, t), C(\vec{r} + \vec{\delta}_1, t), C(\vec{r} + \vec{\delta}_2, t), \dots, C(\vec{r} + \vec{\delta}_q, t)) \quad (1)$$

where $\vec{r} + \vec{\delta}_k$ designates the cells belonging to a given neighbourhood of cell \vec{r} .

In the above definition, the rule R is identical for all sites, and it is applied simultaneously to each of them, leading to a synchronous dynamics. It is important to notice that the rule is homogeneous, i.e. it does not depend explicitly on the cell position r . However, spatial (or even temporal) inhomogeneities can be introduced by ascribing definite and permanent values for some states $C_j(\vec{r})$ in some given locations of the lattice. In the above definition, the new state at time $t+1$ is only a function of the previous state at time t . It is sometimes necessary to have a longer memory and introduce a dependence on the states at time $t-1, t-2, \dots, t-k$. Such a situation is already included in the definition, if one keeps a copy of the previous state in the current state.

The neighbourhood of cell \vec{r} is the spatial region in which a cell needs to search in its vicinity. In principle, there is no restriction on the size of the neighbourhood, except that it is the same for all cells. However, in practice, it is often made up of adjacent cells only. For 2-d CA, two neighbourhoods are often considered [7]: The von Neumann, which consists of a central cell (the one which is to be updated) and its four geographical neighbours north, west, south and east. The Moore neighbourhood contains, in addition, second nearest neighbours northeast, northwest, southeast and southwest that is a total of nine cells. Extending the neighbourhood leads to various types of boundary conditions such as periodic (or cyclic), fixed, adiabatic or reflection [7].

CA can sufficiently represent phenomena of arbitrary complexity and at the same time can be simulated exactly by digital computers, because of their intrinsic discreteness, i.e. the topology of the simulated object is reproduced in the simulating device [8]. The CA approach is consistent with the modern notion of unified space-time. In computer science, space corresponds to memory and time to processing unit [9]. In CA, memory (CA cell state) and processing unit (CA local rule) are inseparably related to a CA cell [10]. In addition, algorithms based on CA run quickly on digital computers [9]. Models based on CA lead to algorithms which are fast when implemented on serial computers, because they exploit the inherent parallelism of the CA structure [8-10].

3 The Proposed Model

The proposed model is a 2-d CA and aims at the simulation of crowd dynamics during the evacuation of a large area. The whole area under test is divided into a 2-d matrix of identical square cells, with side length a , represented by a CA, assuming that each cell of the area is a CA cell. The grid of the CA is considered as homogeneous and isotropic, thus defining major model's features. Furthermore, CA cells are being able to exist in two possible states; either free or occupied by exactly one particle. Moreover, every cell covers an extent of approximately $40 \times 40 \text{ cm}^2$ [4], thus including even the situation of a dense crowd. In any case the value of a is user

defined and should be a compromise between accuracy and computer time and memory.

As far as it concerns the update process, it should be mentioned that it takes place in parallel for all particles. During each time step, an individual chooses to move in one of the eight possible directions of its neighbourhood. However in some cases it is not possible to move at all due to the occupation of cells by other particles. In any case, a particle cannot overcome more than one cell at a time step, meaning that in case that it moves, it does it with a maximum velocity of one cell per time step. Two more assumptions regarding velocity have been also considered. The first one defines that acceleration as well as braking time are negligible while the second one following the attributes of homogeneous and isotropic grid demands stable and common velocity for all particles of the same type. Consequently, a model of maximum velocity equal to unity is formed, meaning that movements are only performed towards the closest neighbours.

A brief description of the local CA rule follows. The general scheme is that each particle moves towards the direction which is closer to exit. In order to implement the appropriate movement, the transition matrix of each occupied cell, depending on the CA cell itself and its neighbours, is updated at every time step. Each matrix element represents a possible updated spatial state of the occupied cell, placed at the centre of the matrix, at the next time step, $t+1$. As far as it concerns the elements' values, these indicate the distances from the escape point of the occupied cell and its eight closest neighbours, respectively. The distance is defined as the minimum number of cells needed to be covered in order to reach the exit, moving strictly either in longitudinal (vertical) or in latitudinal (horizontal) direction. This way, the shortest path towards the exit is scouted.

In Fig. 1 is illustrated an example of a transition matrix. The occupied cell at time step t (dashed cell), placed at the centre of the array, is referred as (i,j) . Consequently, each cell is described by its relative position to the central one. The number found at the top of each cell represents the distance of the possible new position from the exit. For instance, a particle placed at cell $(i-1,j-1)$ needs 5 steps to reach the exit, i.e. 4 steps vertically and 1 horizontally. The minimum distant cell, e.g. cell $(i+1,j)$ placed just 2 steps away from the exit (bold line), becomes the choice of the new particle's position at the next time step, $t+1$.

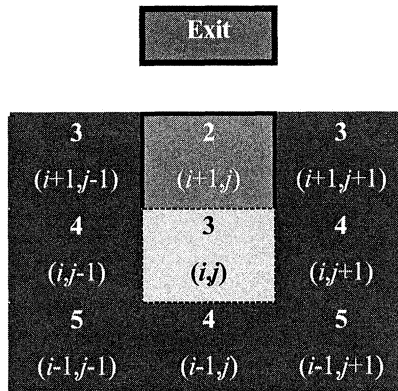


Fig. 1. An example of a transition matrix.

As soon as all possible routes have been detected, the shortest prevails and the particle moves this way at the next time step. This process takes place in parallel for all occupied cells. In case of multiple exits, the whole procedure is repeated for each one separately. In case of collisions the following scheme has been adopted; the particle does not move towards the target unless it is not occupied and it is free to move if no other particle targets at the same cell. In the latter, the cell that fronts the exit proceeds.

The flowchart of the model is illustrated in Fig. 2 while the aforementioned procedure can be summarized as follows:

When all the external parameters (population, noise percentage, type of individuals' percentages, area extent, number of obstacles and obstacles coordinates, number of exits and exits coordinates) have been defined and as a result the evacuation area has been thoroughly defined it is time for the CA to start its evolution. For each occupied cell in the CA grid the corresponding transition matrix is calculated. This process takes place in parallel for all particles. The location of the array's minimum value element represents the target cell of the central reference particle (i, j) for the next update step. If the target cell is occupied, the particle remains to its prior position after update step, while in the case that the target cell is free and no other particle targets it, the move takes place. Finally, provided that the target cell is free and more than one particles target it, then the one that fronts the exit proceeds.

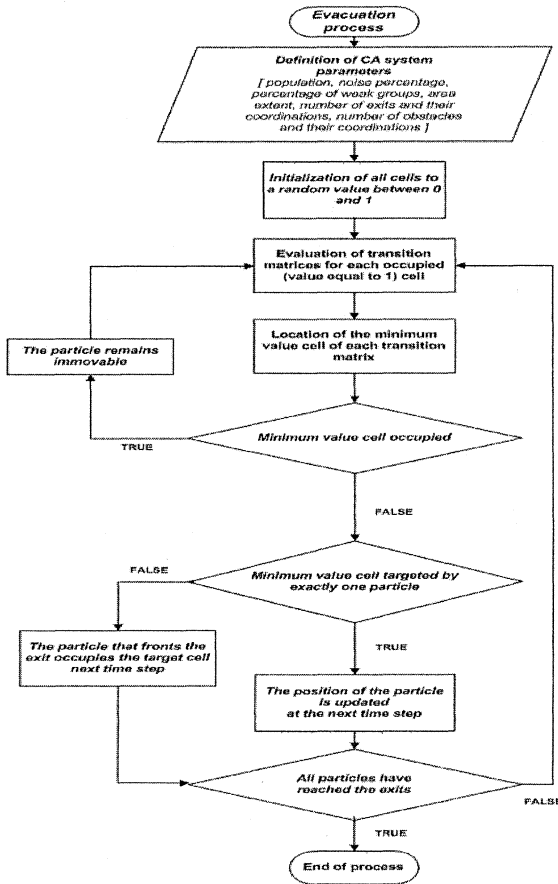


Fig. 2. Flowchart of the evacuation process algorithm.

4 Description of the Graphical User Interface

A graphical user interface (GUI) based on Matlab[®] has been developed. The simulator can be used as a computer tool for the study of the room evacuation process. No previous knowledge of CA or computer programming is necessary to use the simulator, because of the user-friendly graphical user interface that has been developed.

The GUI is equipped with various parameter options, in such an extent that it can provide an adequate level of prominent features of the area under test. It is oriented to elevate the space characteristics of the evacuated area as well as distinctive

properties of the crowd's synthesis. Therefore, the user can adjust the basic parameters of the system according to her/his requirements and to obtain an optical visualisation for a variety of system's characteristics. The evacuation process can be demonstrated for several exit locations defined by the user, for an area of a defined extent which can also be enriched with obstacles at various locations. Moreover, the density of the population as well as the motion of the particles can also be adjusted and the simulation process can be further enhanced including different types of individuals, e.g. the elderly that move slower.

Every system parameter corresponds to one of the various buttons and all of them are placed at the right side of the GUI. The 'Population' button allows the user to define the number of the individuals participating at the evacuation process. Each of them is presented by a coloured point which moves towards the closer exit location. The following one, called 'Weak groups percent', introduces a certain amount of individuals of different type that are also distinctively coloured for reasons of better monitoring.

During the demonstration, for each particle the transition matrix is evaluated and according to the aforementioned algorithm, they change their position in order to reach the escape point following the shortest route. In fact, all evaluations have been made for Moore neighbourhood, namely using 9 (including the central cell) active neighbours. As rationally expected, collisions delay the process without blocking it; already occupied positions are not available and multiple targeted cells are finally occupied by these particles which are preferentially placed relative to the exit.

Furthermore, the graphical interface provides the option to supply in some extent additional delay to the particles' motion using the 'Noise percent' edit button. It represents the movement's obstruction by undefined factors, such as panic or hysteria, resulting in a left-right drift parallel to the direction of the population motion.

The extent of the area under test can be parametrically defined using the 'Area Extent' edit buttons specifying in number of cells the length and the width of the area. The number of exits is defined in the field 'Max Exits' while the number of obstacles is set in the 'Obstacles' button. The obstacles' locations as well as the exits' exact locations are defined by setting their x , y – coordinates at the corresponding fields.

After setting all the simulation parameters, the simulator is activated by clicking the button 'Start' on the top-right of the interface. 'Stop' button freezes the evacuation process demonstration, allowing the user to capture certain phases of the procedure aiming at closer surveillance. 'Info' button provides a short description of the aforementioned algorithm as well as access to the code itself while the 'Close' button closes the demonstration window.

In Fig. 3 four different successive snapshots of the simulation process are presented for different time steps.

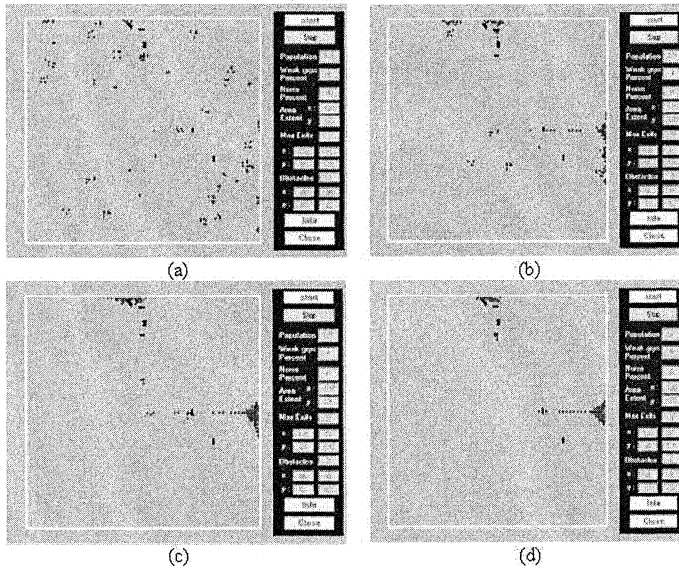


Fig. 3. Four different successive snapshots of the proposed model's GUI for different time steps. Middle aged individuals are coloured red, while the elderly individuals as well as children are coloured green. Black dots correspond to the obstacles' locations.

5 Conclusions

The presented pedestrian dynamics model is a computational technique based on the principles of a 2-d CA. Although a simplified version due to assumptions concerning velocity and negligible acceleration and braking time, certain attributes of crowd behaviour, such as collective effects, collisions and delaying factors have been successfully encountered during simulation process. Several important phenomena of crowd dynamics, meaning transition to incoordination (arching) due to clogging as well as mass behaviour, have become pronounced.

The user-friendly interface provides the ability for observing various situations of room evacuation process, thus drawing useful conclusions. Room arrangement, regarding the number and the location of exits as well as the obstacles' ordering can be defined in a user-friendly manner. The fact that the model is parameter-enriched in quite a remarkable extent offers the option of an easy, external adjustment of various significant features of the evacuated area, concerning both topological as well as crowd formation aspects. Consequently, the existence of several, externally defined parameters enhances the surveillance perspectives of a certain area. General

characteristics are confined, thus indicating certain characteristics allowing the implementation of more effective monitoring techniques.

Based on the flexibility of the proposed model, it could be possible that video sequences related to crowd escaping to be used in order to calibrate the CA model. Furthermore, this possibility could elucidate the response of the model to real data. In other words, data from video monitoring can be supplied to the model in order to realize in what extent is the model capable of reproducing various observed phenomena under panic circumstances. Such an upgrade would certainly increase its level of credibility, in order to be used to test gathering areas for their suitability to emergent situations.

Finally, due to the fact that in terms of circuit design and layout, ease of mask generation, silicon-area utilization and maximization of clock speed, CA are perhaps one of the most suitable computational structures for VLSI realization, the VLSI implementation of the proposed CA algorithm is straightforward with no silicon overhead. As a result it is feasible the perspective of an integrated surveillance system, with camera based monitoring algorithms that would be able to provide the necessary video data.

References

1. Helbing, D., Farkas, I. Vicsek, T.: Simulating dynamical features of escape panic. *Nature*. 407 (2000) 487–490
2. von Neumann, J.: *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana IL (1996)
3. Wolfram, S.: *Theory and Applications of Cellular Automata*. World Scientific, Singapore (1986)
4. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A*. 295 (2001) 507–525
5. Aubé, F., Shield, R.: Modeling the Effect of Leadership on Crowd Flow Dynamics. *Lecture Notes in Computer Science*. 3305 (2004) 601–611
6. Klüpfel, H., Meyer-König, T., Wahle, J., Schreckenberg, M., in: S. Bandini, T. Worsch (Eds.), *Theory and Practical Issues on Cellular Automata*, Springer, Berlin, 2000.
7. Chopard, B., Droz, M.: *Cellular Automata Modeling of Physical systems*. Cambridge University Press, Cambridge (1998)
8. Karafyllidis, I., Andreadis, I., Tzionas, P., Tsalides Ph., Thanailakis, A.: A Cellular Automaton for the Determination of the Mean Velocity of Moving Objects and its VLSI Implementation. *Pattern Recognition*. 29 (1996) 689–699
9. Sirakoulis, G.Ch.: A TCAD system for VLSI implementation of the CVD process using VHDL. *Integration, the VLSI Journal*. 37 (2004) 63–81
10. Sirakoulis, G.Ch., Karafyllidis, I., Thanailakis, A.: A Cellular Automaton for the propagation of circular fronts and its applications. *Engineering Applications in Artificial Intelligence*. 18 (2005) 731–744