A More Realistic Simulation of Pedestrian based on Cellular Automata

Pengyuan, Shao
School of Information Science and Technology
Xiamen University
Xiamen, China
barretshao@gmail.com

***Abstract*—the simulation of pedestrian has been studied for a long time from various points; however, most of them have not considered the psychological and terrain factors for pedestrians. Firstly, we develop the fundamental model of pedestrian simulation based on Cellular Automata. To consider the terrain factors, this paper embeds neural network into these agents to ensure intelligent and realistic. The data of training neural network are collected from real pedestrians, which make agents more realistic. This model can help us find the best path of pedestrians' simulation in both single and many pedestrians. This method can be widely used in the business strategy and security control. These implementations are based on the open-sourced toolboxes in Scilab, including Neural Network Toolbox and Cellular Automata Toolbox.**

Keywords-Cellular Automata, pedestrian simulation, Scilab, BP-Neural Network

I. INTRODUCTION

Currently, planners and business executors need to make suitable and accurate decisions, concerning business location selection, emergency equipment implementations etc. Macroscopically analysis, such as optimization, makes a relatively inaccurate decision. The reason is that these methods can not grasp the details and swarming phenomena. Because of the nonlinear behaviors, it is necessary to use computational simulation of pedestrian on graph.

Cellular Automata (CA) model, proposed by Von Neumann in the late 1940s, has been applied in the complex system simulation, such as fluid dynamics and urban engineering [1, 6, 7]. It is a discrete model studied in computability theory, mathematics, theoretical biology and microstructure modeling frequently. CA model mainly consists of the CA grids and agents, while grids define the uniform lattices of agents with local states [2]. Agents need to follow some particular rules which lead the swarming behavior of pedestrians. Besides, according to CA, it is possible to choose the suitable neighborhood model such as Fig.1.

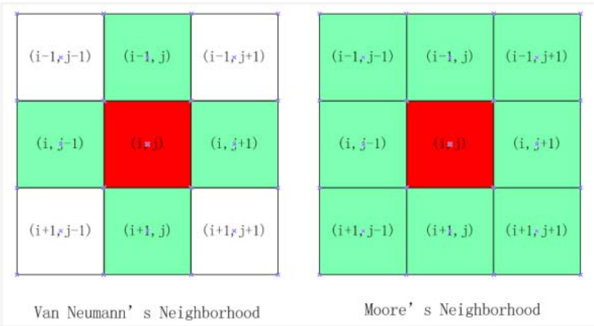

Figure 1. Neighborhood Structure

Moreover, range of cellular automata is important in making decision, shown in Fig 2. Different ranges of cellular automata will affect the behaviors of agents, especially in pedestrian simulation. Range would affect the personal space [3], which will contribute a comfortable area and influence the action of pedestrian.

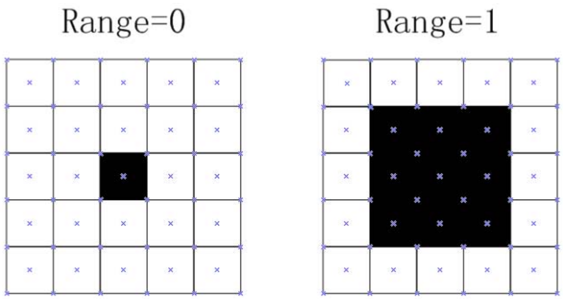

Figure 2. Range of Cellular Automata

In previous work of applying CA to pedestrian simulation, there are lots of works, focusing on mechanical movement of pedestrian and emergency escaping etc. Pedestrians were regarded as the agents and rules were embedded in agents as intelligence. Different pedestrians can communicate and interact through the neighbor and grid, and then they can make proper rules according to the rules. Most works of the pedestrian simulation on CA focus on the rule, which represents agents' intelligence.

Most previous papers have not considered the terrain and other variables except the interactions among agents [2]. However, terrain always plays an important role in decision making by pedestrians. It is hard to model these nonlinear conditions by usual optimized theory. In this work, BP Neural Network is applied to make the rules based on the previous knowledge of human-beings. BP neural network can solve these nonlinear problems much better than traditional computational methods. Some useful attributes or factors are selected as input variables, and the decisions of pedestrians are represented as the output. After the training on previous knowledge, BP neural network can play the function of brain of these agents in CA. Then they can make proper choices even when facing the unpredicted situations. Through this more realistic simulation of pedestrian, we can find the difference between whether considering terrain or not.

Scilab has developed both Neural Network Toolbox (published) and Cellular Automata Toolbox. Both of them are open-sourced and friendly to apply.

II. Fundamental Simulation and Frame

A. Basic Assumptions

Following assumptions of grids and agents are made to realize the fundamental model of pedestrian simulation.

Assume that the space of simulation is $Width \times Width$ and set $\{i, j|0 \leq i, j \leq \mathrm{Width}\}$ is used to represent the different locations of cells [4], which can be regarded as the uniform grids, while t represents the discrete time point.

There are four situation types of cells and variable S is used to represent them.

$$S \in \{\mathrm{Object}, \mathrm{Available}, \mathrm{Decision}, \mathrm{Held}\}$$

Where "Object" refers to that this cell is held by the objects and can not be gone through, such as barrier, shop and canyon; "Available" refers to that this cell is available to get across; "Decision" refers to that this cell will be held by pedestrians next time, because it has been "decided" by other agents; "Held" refers to that this cell is held by one pedestrian.

Hence $A\{i, j, S, t\}$ represents required information of cell.

Here we note the two variables for agents and grids by using the signals as follows:

$$S = \{k, x_k, y_k, x_k^*, y_k^*, \mathrm{next}(k)\}$$

Where k represents k^{th} pedestrian of all; (x_k, y_k) represents this pedestrian's current location in two-dimensional area by the grid location; (x_k^*, y_k^*) represents this pedestrian's destination. The important character of S is $\mathrm{next}(k)$, which refers to the choice this pedestrian will make. $\mathrm{next}(k)$ is decided on both rules and current grid situation. Simply assume that

$$\mathrm{next}(k) = \{\mathrm{up}(k), \mathrm{down}(k), \mathrm{left}(k), \mathrm{right}(k), \mathrm{stop}(k)\}$$

The other variable for the cell's description is

$$C = \{i, j, \mathrm{cell}_{\mathrm{situation}}\}$$

C is easier to describe than S , where (i, j) represents the location of this cell. $\mathrm{cell}_{\mathrm{situation}}$ represents the current situation of this cell and $\mathrm{cell}_{\mathrm{situation}} \in \{0,1,2\}$, where they represent available, held by pedestrian and held by block respectively.

B. Rules without considering terrain

For k^{th} person (agent), it needs to do the following steps:

Step1: Find the current location of pedestrian (i, j) and check the shortest way to the (x_k^*, y_k^*). Then check $\mathrm{cell}_{\mathrm{situation}}$ of this required cell situation. If $\mathrm{cell}_{\mathrm{situation}}$ is 0, then (x_k, y_k) is changed as this cell location. Meanwhile, this cell's $\mathrm{cell}_{\mathrm{situation}}$ will be 1. If not, then do the next step;

Step2: Check the secondary cell's situation by the pedestrian's behaviors, if the situation is 0, then (x_k, y_k) is changed as this cell location. Accordingly, this cell's $\mathrm{cell}_{\mathrm{situation}}$ will be 1. If not, then do this step again until check all the adjacent cells' situation, then do the next step;

Step3: This pedestrian will not change his position and stay at that position at this time.

C. Simulation and Analysis of this fundamenta simulation

This graph is one state of the simulation process, where the red point represents pedestrian current position and blue area represents the destination of these pedestrians.

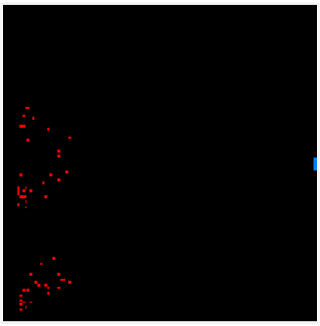

Figure 3. Simulation of this model

This fundamental simulation model can show the basic movement rules of pedestrian when there are obstructs and changing neighborhood situation [4]. The simulation has illustrated the interaction and communication among different agents, even though it is simplified.

However, this basic model is based on the assumption that terrain factors are not considered. Next part we will discuss the further model, considering terrain factors based on this basic model.

III. Model with Terrain

To consider terrain factors, we need to add some variables to record altitude of these grids.

A. Altered Assumptions

By adding the attributes of altitude of grids, the cell's variable description will be changed as:

$$C = \{i, j, \mathrm{cell}_{\mathrm{situation}}, \mathrm{cell}_{\mathrm{altitude}}\}$$

Where $\mathrm{cell}_{\mathrm{altitude}}$ represents altitude of $C(i, j)$, which is similar to $\mathrm{cell}_{\mathrm{situation}}$.

However for pedestrian, agent, we need to build a variable which combines current altitude of pedestrian and neighborhood. Importing the grads concept to analyze it, this refers to the perception of pedestrian itself rather than designers.

$$S = \{k, x_k, y_k, x_k^*, y_k^*, next(k), grad(x_k, y_k)\}$$

Where $grad(x_k, y_k) = \begin{bmatrix} grad_{forward} & grad_{back} \\ grad_{left} & grad_{right} \end{bmatrix}$

We define $grad_{foward} = artan\left(\frac{\Delta altitude_{forward}}{\Delta movement}\right)$

Besides, to simplify the description of pedestrian, this model combines (x_k, y_k) and (x_k^*, y_k^*) together as the relative position of pedestrian, shown as follows:

$$relative_{position} = \arctan\left(\frac{y_k^* - y_k}{x_k^* - x_k}\right)$$

In this way, the angular matrix can be used to analyze current position, goal position and the altitude.

B. Design and Implementation of BP Neural Network

This BP Neural Network consists of one input layer, one output layer and one or more hidden layers. Each layer consists of multiple neurons that are connected to the neurons in the next layers. Compared with the normal algorithm, neural network can simulate the interactive action of human-beings, such as pedestrians, more realistically.

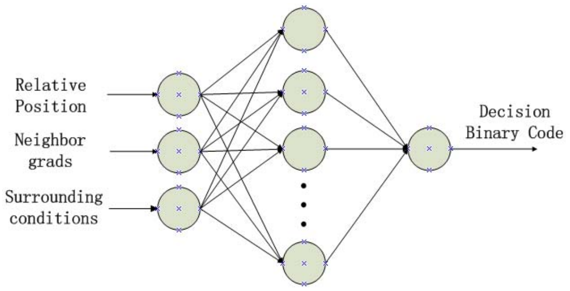

Figure 4. BP Neural Network Structure

The inputs of neural network are consisted of the following variables.

$$Input = \{relative_{position}, grad(x_k, y_k), surr(x_k, y_k)\}$$

Where $grad(x_k, y_k)$ and $surr(x_k, y_k)$ represent grads and situation of surrounding neighborhood as above-mentioned.

As for the output, since we require this neural network can make $next(k)$ according to the immediate information acquired by agents. Being the same with the structure of $next(k)$, output is bounded by the five situations as follows:

$$output = \{stop, up, back, left, right\}$$

To simplify it, binary coding is used to translate them into 000, 001, 010, 011, and 100 respectively. Hence the number of neurons in output layers is 3. Furthermore, since all of the output values are 0-1, it is proper to use logsig() as the transfer function in the output layer.

Considering the training data, this work collects suitable data from a particular survey on people among Xiamen City. To collect these data, virtual reality is used to simulate the different terrains respondents. The training data have property of unique and limitation, which means that these data are not suitable to model other places pedestrians. So simulations of different areas' pedestrian require various training data.

Before training the neural network, it is necessary to do data preprocessing based on the collected data. $relative_{position}$ and $grad(x_k, y_k)$ are angular matrixes, which are different from the $surr(x_k, y_k)$. Hence it is necessary to make a normalization of $relative_{position}$ and $grad(x_k, y_k)$.

To illustrate the structure of neural network better, there is a random example of collecting data:

$$Input = [-0.25, 0, 0, 0.3, 065, 1, 0, 0, 0] \quad Output = [0, 0, 0]$$

Which means that when an agent is confronted with the situation, that $relative_{position} = -0.25$, $grad_{foward} = grad_{back} = 0$, $grad_{left} = 0.3$ and $grad_{right} = 0.65$ which means that there are upgrades on both sides. $[1, 0, 0, 0]$ represents that neighborhood cells are available except the forward one. Finally according to the data, this agent chooses to stop rather than to move left or right.

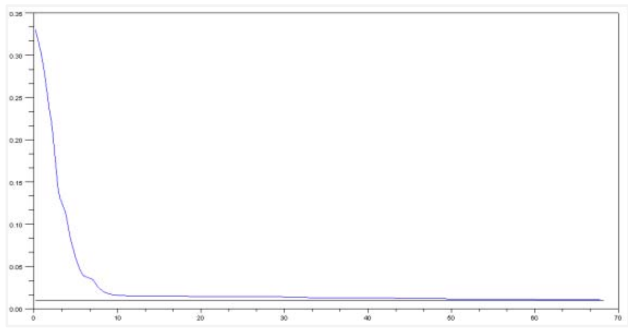

Figure 5. Training result

Fig 5 shows the training result of the above-mentioned neural network model, where blue line is training line and black line is the goal. Since the goal error is 1%, finally the performance is 0.919% after 27 epochs.

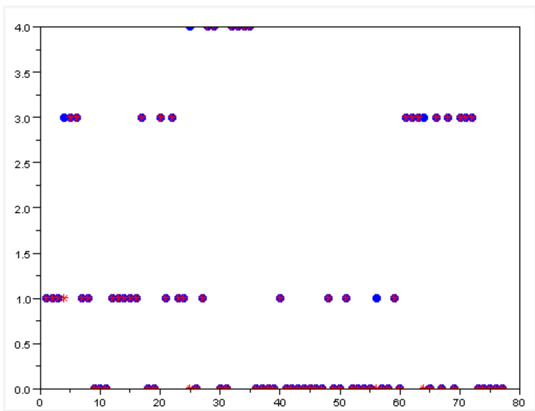

Figure 6. Test of Training Data

Fig 6 shows the original data and output of the neural network model. Where red points represent the original data and blue ones represent output of model. The precision is 94.8% out of 78 samples.

Furthermore, to test the efficiency of this neural network model, we use some additional data, which sample size is 23, to test the output.

From the Fig 7, we can find that when the input of neural network is dealing with new data, they mostly act like the human-beings. The precision is 91.3% out of 23 samples.

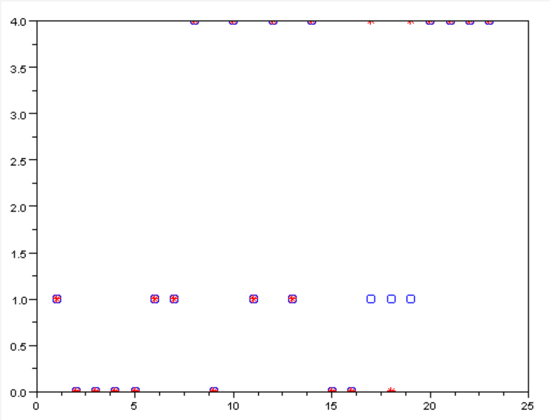

Figure 7. Test of New Data

C. Embed BP Neural Network in Cellular Automata

After the BP neural network is well trained, it can help agents make the decision concerning different terrain conditions, we need to embed this trained neural network into all of the cells in order to make them more intelligent. Compared with previous works on simulation of pedestrians, it is no longer simply taking into account "shortest path" [5].

In this part, a model of cellular automata, embedded by BP neural network, is used to simulate one pedestrian simulation. Through this part we can see the obvious difference between this model and other previous works. As for the communication among different agents in same area, we will show it in the later part.

First we select a random terrain condition which differs from the training data. Fig 8 shows the geographical condition in Scilab.

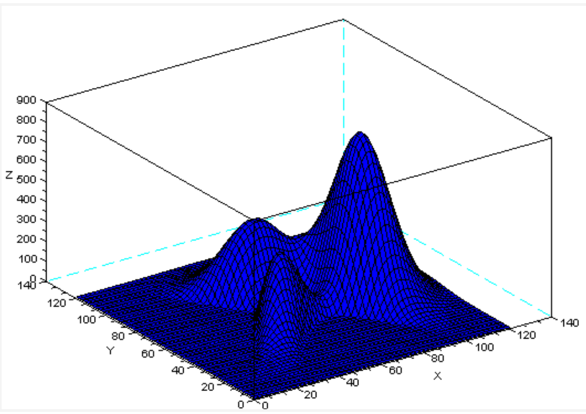

Figure 8. Geographical Condition

Since the goal of this paper is to find the realistic path of pedestrian, it is not necessary to use the 3D geographical representation which is used in previous works [3]. Besides, in the above-mentioned model assumptions, one of inputs of neural network requires the altitude, such as

$$grad_{foward} = \operatorname{artan}\left(\frac{\Delta altitude_{forward}}{\Delta movement}\right)$$

So it is better to use contour line to represent terrain which is more convenient to deal with altitude. Fig 9 is the contour line graph of the previous geographical conditions, where the digits denote the various numbers of altitudes.

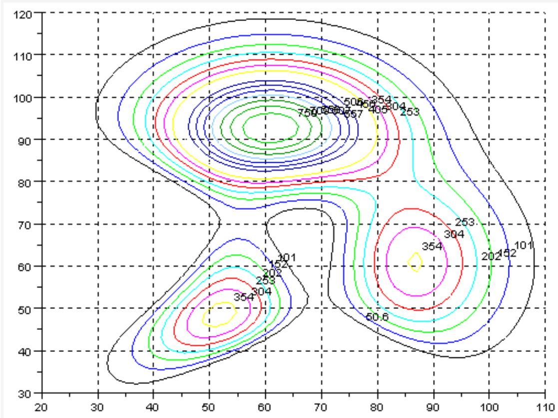

Figure 9. Contour Line

After denoting the terrain, assume that the only agent starts from point (70,50), whose destination is point (70,120). Without considering the terrain, the agent will move forward directly to the goal, since it is the shortest path. However, if do so, agents need to be confronted with the huge $grad_{foward}$, which is obvious not preferred by real pedestrian.

Cellular automata embedded with neural network are used to simulate this pedestrian. To record the path of pedestrian, an array is used to record agent's decisions: $P(i) \in \{0,1,2,3,4\}$, where 0,1,2,3,4 represent stop, forward, left and right respectively. In this way, path of pedestrian can be recorded.

Fig 10 shows the recording of agent's paths, which are pointed in red asterisks.

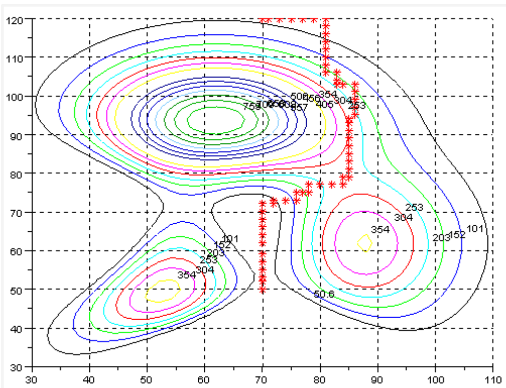

Figure 10. Recording of Pedestrian Path

From the single agent recording in a complicated terrain, cellular automata embed with neural network simulates pedestrians much more realistically. Even though there are a few decisions which might be different from real pedestrian, it can be amended by the improvement of sample data.

D. Swarming effect of Simulation

The previous part is analyzing the simulation of only one pedestrian. When the number of agents increases, agents have to communicate with different agents and make decisions based on the neighborhood occupied conditions. Swarming effect or group behavior of pedestrian is very important in nowadays business optimization and civil engineering, because it is necessary to acquire more general information concerning the pedestrians. [3]

In this part, the basic model is the same with the former one, which also includes the surrounding cells situations. To analyze the swarming effect of pedestrian, especially on the different terrains, we enlarge the number of pedestrians in this area and provide them a same destination.

Because too many paths are not convenient and intuitive to analyze, it is necessary to make some additional criteria. An important criterion is the average waiting time $\overline{T}$ [6, 8]. In a geographical condition which is full of upgrades, agents need to make a tradeoff between waiting and upgrades. In previous works concerning only shortest path, pedestrians will choose other directions instead of waiting. However, since the training data for neural network are collected from human-beings who are averse to the upgrades, it is possible that agents in this model will wait for the agents who have occupied their desired cells.

We put 50 agents in the area and run this simulation among agents' communications and interactions. Assume N is equal to the number of periods for simulation. Since the space of simulation is 120×120, we can properly set $N = 100$ to calculate $\overline{T}$. Besides, the periods of stop and waiting, need to be recorded for every agent. Use $W(i,j), 1 \le i \le 50, 1 \le j \le N$, to define whether the i^{th} agent is waiting in the j^{th} period.

$$W(i,j) = \begin{cases} 1, & \text{if output of } i^{th} \text{ agent is equal to "000"} \\ 0, & \text{otherwise} \end{cases}$$

Hence $\overline{T}(i)$ can be defined as follows:

$$\overline{T}(i) = \frac{\sum_{j=1}^{N} W(i,j)}{N}$$

To build a control with our model, the same number of agents and space will be applied to the basic CA model, which does not consider the terrain factor.

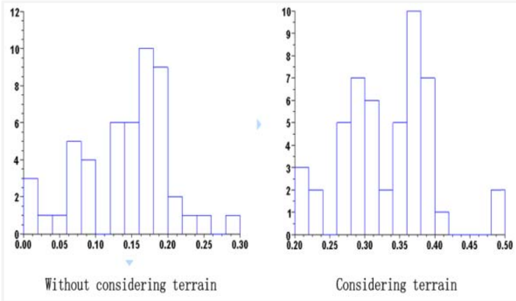

Figure 11. Histogram of $\overline{T}(i)$

Fig 11 is the histogram of $\overline{T}(i)$, where the left one represents the basic CA model without concerning terrain, the right one represents the CA model embedded with neural network. More accurately, the characteristics of statistics are as follows:

Table 1

Model	Statistics Characteristics	
	Mean	***Standard Deviation***
Basic CA Model	0.1454	0.0629
CA model with neural network	0.3356	0.0644

From Table 1, we can easily find that the average waiting time $\overline{T}(i)$ of CA model with neural network is much longer than the basic CA model's. In this point, the CA model with neural network works more realistically considering the terrain factor. Agents' goals in this model are to make a suitable tradeoff between upgrades and waiting time, conversely, basic model's goal is only the shortest path.

IV. CONCLUSIONS

This paper presents a new method for pedestrian simulation based on open-sourced toolbox for Scilab. To make up the shortage of basic CA model for pedestrian simulation, we embed a BP neural network in agents. The gradients of agent's neighborhood terrains can represent various terrains. After the training of neural network on the collected data from real pedestrians, agents under this model become more intelligent. The simulations of one pedestrian show that this model can match reality well and that agent will seek the path with fewer upgrades. Then we make a swarming effect of this pedestrian simulation and build a criterion to value this model.

The concept of embedding trained neural network in agents makes the simulation more realistic. However, the training data will influence the effect of simulation. To make the agents more intelligent and realistic, other factors can also be included, such as psychological and visional factors. Hence future work can be intended to these based on the concepts.

REFERENCES

[1] Kazuhiro Yamamoto1, Satoshi Kokubo1, and Katsuhiro Nishinari2, "New Approach for Pedestrian Dynamics by Real-Coded Cellular Automata (RCA)" ACRI 2006, LNCS 4173, pp. 728 – 731, 2006.© Springer-Verlag Berlin Heidelberg 2006J. Clerk Maxwell, A Treatise on

Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Dijkstra, J. , H.J.P. Timmermans and A.J. Jessurun (2000). “A Multi-Agent Cellular Automata System for Visualising Simulated pedestrian Activity”; Theoreteical and Practical Issues on Cellular Automata - Proceedings on the 4th International Conference on Cellular Automata for research and Industry, S. Bandini and T. Worsch (eds.), Springer Verlag, held at Karlsruhe, Germany, October 4-6, 2000. pp. 29-36.

[3] Shohei SHIMOHARA , Tsukasa WATANABE , Toshikazu SHIMAZAKI , “Cellular Automata Model of Pedestrian’s Group Behaviour” Proceedings of the Eastern Asia Society for Transportation Studies,Vol.6,2007

[4] Dijkstra J, Timmermans HJP. “Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance.” In: Ataman O, Bermúdez (eds) ACADIA99 – Media and design process. Acadia 1999: 226-237

[5] C. Burstedde, K. Klauck, A. Schadschneider, J. Zittartz, “Simulation of pedestrian dynamics using a two-dimensional cellular automaton.”, Physica A, vol.295 (2001) 507-525.

[6] Boris Goldengorin, Alexander Makarenko, and Natalia Smelyanec, “Some Applications and Prospects of Cellular Automata in Traffic Problems” ACRI 2006, LNCS 4173, pp. 532 – 537, 2006. © Springer-Verlag Berlin Heidelberg 2006

[7] Xin-Gang Lia, Bin Jiaa, Zi-You Gaoa, Rui Jiangb, “A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle”2005 Elsevier B.V. All rights reserved. doi:10.1016/j.physa.2005.11.016。

[8] Nicole Ronald and Michael Kirley , “Pedestrian Modelling: A Comparative StudyUsing Agent-Based Cellular Automata” ICCS 2006, Part III, LNCS 3993, pp. 248–255, 2006.Springer-Verlag Berlin Heidelberg 2006。