Crowd-Z: the user-friendly framework for crowd simulation on an architectural floor plan

Machi Zawidzki* (E: zawidzki@gmail.com T/F: +81 3 5452 5286)

Research Center for Advanced Science and Technology, the University of Tokyo, Japan

Mohcine Chraibi

Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany

Katsuhiro Nishinari

Research Center for Advanced Science and Technology, the University of Tokyo, Japan

Abstract

This paper introduces Crowd-Z (CZ): a framework that provides a user-friendly platform where architects can perform simple crowd simulations on floor plans. A simple but robust and flexible agent-based system is used for modeling of the crowd dynamics. Such simulations can be performed at any stage of design - from rough sketches to the final blueprints. CZ allows acquiring the layouts for the simulations in a number of ways: freehand sketches, importing already prepared images and appropriating preprocessed images from commercially available Computer Aided Design programs. These three methods are illustrated with practical examples, followed by a number of simulations compared with the literature or other commercially available programs.

Keywords: Pedestrian dynamics, agent based modeling, design support, digitized floor plan.

[1]

1. Introduction

In the past, several models for pedestrian dynamics were developed and used for various purposes, e.g. planning of buildings, organization of mass events etc. In general, these models can be classified in two major categories: space continuous models, which are based on the Newtonian approach [1, 2, 3, 4] and rule-based models discrete in space that describe the dynamics of crowd on a regular grid by means of transition probabilities [5, 6, 7, 8]. Particularly relevant to the concept presented in this work are agent-based systems [9, 10, 11].

Most of these models describe rather well pedestrian dynamics quantitatively and qualitatively. Based on these models, several commercial tools are available, for example: VISSIM [12], ASERI, [15], PedGo [17]. However, in the common architectural and urban planning practice they are rather rarely used. One of the reasons is that designers often do not feel necessity for such experimentation, but also they may get discouraged by the price, complexity or "unfriendliness" of the available software. The motivation for this work is to introduce an intuitive environment for designers, in which they can comfortably perform crowd simulations (CS). The holistic approach presented here also contributes to the academia so the researchers can have an overview of what designers expect from such tools and models. For the same reason - it is also addressed to the software engineering community. Crowd-Z (CZ): the user-friendly framework brings the idea of CS to broader audience than before, and hopefully will encourage the architects and planners to use it for designing safer and more convenient spaces. The main objectives of CZ are:

1. To simulate the crowd dynamics (CD) on a given floor plan, in particular - to find the problematic areas where the traffic does not flow properly or conversely, the areas that are visited too sparsely.
2. To allow for experimentation by alterations to the geometry of the environment, in particular by introducing or erasing elements or their parts, changing the locations of the start terminals, exits etc., and to immediately observe the influence of these changes on the overall CD.
3. To study the emergent behavior of agents determined by different behavioral scenarios.

The next section briefly describes the model used for CSs, followed by three practical applications of CZ in architectural design and a number of simulations validating the CZ model.

2. Crowd Dynamics Component

This work focuses on the geometrical processing rather than the mathematical modeling of pedestrian dynamics. Although modern models produce more realistic results in terms of CD, for testing purposes a very simple and robust model is introduced. Since the Crowd Dynamics Component (CDC) is an autonomous module of CZ, it can be easily equipped with more sophisticated models. The one presented here contains only a few parameters, which makes it intuitive and easy to understand, particularly for the users who are not familiar with crowd modeling.

Various strategies describing how pedestrians move towards exits are investigated in [13, 14]. In the presented work a simple strategy based on the shortest path is used. It is defined at each cell of the environment by the smallest distance to the given exits. Hereby, the distance is defined by taxicab metric [21]. The exits and walls are assigned the values of zero and infinity, respectively. A matrix containing this information for entire environment is called distance field (DF). In CZ, DF is static and does not depend on the dynamics of the agents. The value of DF_{ij} is proportional to the distance of from a cell (i, j) to the given exit. The definition of the distance is determined by the type of the neighborhood (e.g.: Moore, von Neumann).

In cases with multiple exits, DFs are calculated for each exit k and combined into DF^*_{ij} in a way so that for each cell (i, j) the lowest value in the corresponding position in all DFs is selected:

$$DF^*_{ij} = \min_{k=1,\cdots,n} DF^k_{ij}, \tag{1}$$

with n the number of exits.

As a result, each agent proceeds to the closest exit. The movement of agents is determined entirely by the local neighborhood defined by DF, and can be expressed as the following rules:

1. If all of the neighboring cells are occupied, either by other agents or walls - wait.
2. Check the values of DF in the available neighboring cells and act according to the "perkiness".

The "perkiness" describes the willingness of an agent to move. In this model there are three possibilities: Lazy, Conservative and Perky. Lazy will only move to a neighboring cell if its value is lower than the current one, Conservative will move also to an equipotent cell and Perky will move to a neighboring cell of the lowest DF value regardless of its current value. This means that a Perky agent may occasionally move away from the goal, which may be particularly beneficial in the congested areas.

Depending of the actual speed, the perkiness can be adjusted dynamically. For example, starting from the Lazy tactics, if the effective speed of an agent decreases due to the congestion change to Conservative and finally to Perky. Such an adaptation can be implemented globally to all the agents according to the overall performance of the crowd, or individually to each agent according to its speed. In the case of equivalent choices, the destination cell is selected randomly.

In this paper, in order to avoid the conflicts among agents who choose the same location - sequential update is used. For the study of the influence of the update scheme on CD, see [7]. In the presented work CDC also supports the speed of agents larger than one. This is implemented in the way so that a given agent can update its position more than once at one time step. This means that agents can effectively advance at multiple steps in a single move. By assigning different maximal velocities to the agents, it is possible to analyze, to a certain extent, the effects of inhomogeneities [19]. Studies have shown that the best agreement with empirical observations has been achieved with 5% slow and 5% fast walkers [20]. Since in CZ the effective local speeds of all the agents are recorded, it is also possible to adjust the preferred speed, that is the number of steps at a time step of the agents. The following section illustrates the simulations of different crowd scenarios on three examples of various scales and complexity: from playing with a sketch, through a residential building to a large church.

3. Three practical examples of CS with Crowd-Z

Digital imaging is a common and well established method of managing visual information. Since digital images are based on discrete, regular grid, they can be appropriated for agent-based simulations rather easily. A digital image can be provided to CZ in three ways: directly by a user, acquired (e.g. downloaded from the Internet) or imported from Computer Aided Design (CAD) program, as described in the corresponding subsections below.

3.1. Sketch and Scan

Probably the most intuitive way of providing an environment for simulation is by simply drawing it on a computer screen with a mouse, on a touch-pad or on a sheet of paper and scanning it. An example of a sketch at the early stage of design of an exhibition space is shown in figure 1a.

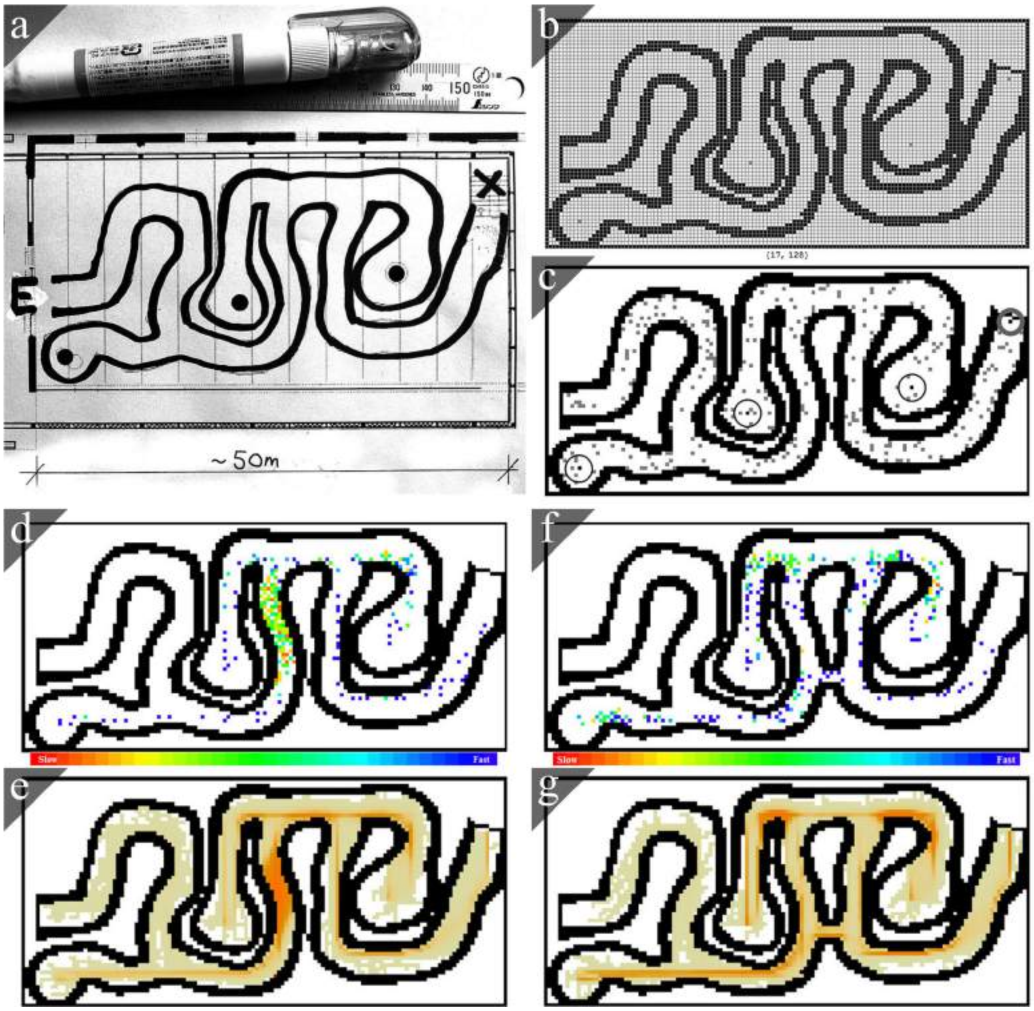

Figure 1: a). A free-form exhibition space with three exhibits marked with black dots. The entrance and exit are marked with E and X, respectively. b). In the CZ interactive panel, clicking with a pointer on a cell reverses its value (wall to void and vice versa). The exhibits and exit are placed manually and shown in gray. The coordinates of the last clicked position are shown below (row from the top, column from the left). c). 250 randomly distributed agents ready for the simulation are shown in gray. Three exhibits are shown in black and marked with black circles. The exit is marked with a gray circle. d). A screen-cast of the initial simulation. e). The corresponding heat map which confirms the congestion area. f). A screen-cast showing that a small alteration to the layout can substantially improve the agents' flow. g). The corresponding heat map.

The image has been scanned and imported to CZ. Since the size of the environment must relate to the size of the agents, the scan has been also resized. In the literature, the empirical maximal density of human crowd is given as 6.25 Persons/ m^2 [22], which leads to assumption that the minimal space requirement for an agent is 0.16 m^2, which corresponds to a $40 \times 40\,\mathrm{cm}^2$ cell on a square grid. Therefore in human pedestrian simulation with discrete models, the size of an agent is usually assumed to be $40 \times 40\,\mathrm{cm}^2$ [19]. Accordingly, the 50 meters wide exhibition space was rasterized into 125 pixels wide grid as shown in figure 1b. The following scenario has been simulated: each agent visits one, two or three exhibits and then proceeds to exit. The sequences of visits are random, and the initial positions of 250 agents are randomly distributed as shown in the bottom right of figure 1c. It takes approximately one minute for all the agents to clear the space. The local velocities are dynamically displayed as the simulation runs, and the problematic areas can be instantly identified as shown in figures 1 d&e. CZ allows to make intuitive, manual modifications to the layout. For example, creating a small connection between the corridors is expected to alleviate the congestion. The new simulation and the corresponding heat map confirming this improvement are shown on the right in figures 1 f&g.

3.2. Large layout: St Peter's Basilica

As an example for a simple evacuation study, St Peter's Basilica (SPB) is used for the following reasons: it is a well-known, truly large building; the documentation including digital images of the plan are easily acquirable from the Internet, there are multiple exits and it has already been used for validation by other researchers, e.g.: [18]. A digital image has been downloaded and re-sampled so that each pixel corresponds to a $40 \times 40\,\mathrm{cm}^2$ square area, as shown in figure 2a. Next, the global DF^* based on DFs for all 15 exits was calculated according to formula (1) as visualized in figure 2b.

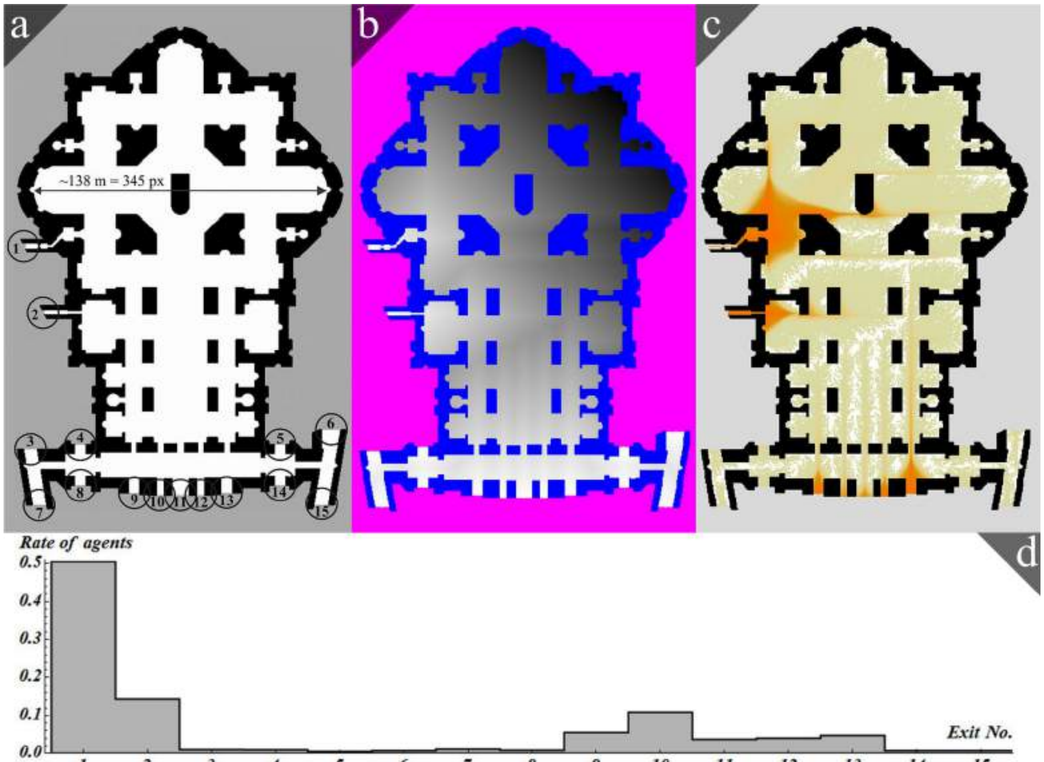

Figure 2: a). The plan of SPB. The width at the transept is approximately 138 m. 15 assumed exits are indicated accordingly. b). The gray gradient visualize DF^*: the darker, the further from the exits. c). The heat map after approximately half of the 10^4 agents have left the building. d). The agent distribution among the exits. The results are averaged from 100 trials of 1000 Conservative agents at OSU and $eR = 0$.

For a simple evacuation simulation, 10^4 agents have been randomly distributed on the plan. After approximately 8 hours of simulation on an Intel Xeon W5590 desktop PC all the agents have left the space. It took so long, for the following reasons: the positions of agents where displayed continuously, the number of agents was exaggerated (almost three times more than in [18]), each exit is only the size of a single cell, and the agents proceeded to the closest exit. This setup was chosen to examine the computational robustness of CZ. An intermediate screen-cast of the heat map is shown in figure 2c.

3.3. Import from CAD

In architectural practice, blueprints of floor plans are usually in a form of vector CAD drawings as shown in figure 3a. In the case of usability for CS, such drawings contain much unnecessary information and often do not articulate the most important spatial features. However, it is easy for architects to prepare them for CS by: filling the walls and other inaccessible areas such as kitchen furniture, toilet fixtures and fixed furniture with uniform, solid hatch; removing the door and window markings, etc. Such pre-processed drawing can be easily exported as a high quality raster image from any commercially available CAD program, as shown in figure 3b. The width of this layout is approximately 20 meters, which corresponds to 50 pixels on a grid. Unlike the case of SPB, here the elements of the plan are very fine in relation to the size of the grid. Therefore in order to produce the best results, a semi-automatic down-sampling in CZ is done under user's supervision. At first, the original layout is normalized to the size of an agent, that is divided into $40 \times 40\,\mathrm{cm}^2$ areas, as shown in figure 3c, and next, each sub-image is averaged as shown in figure 3d.

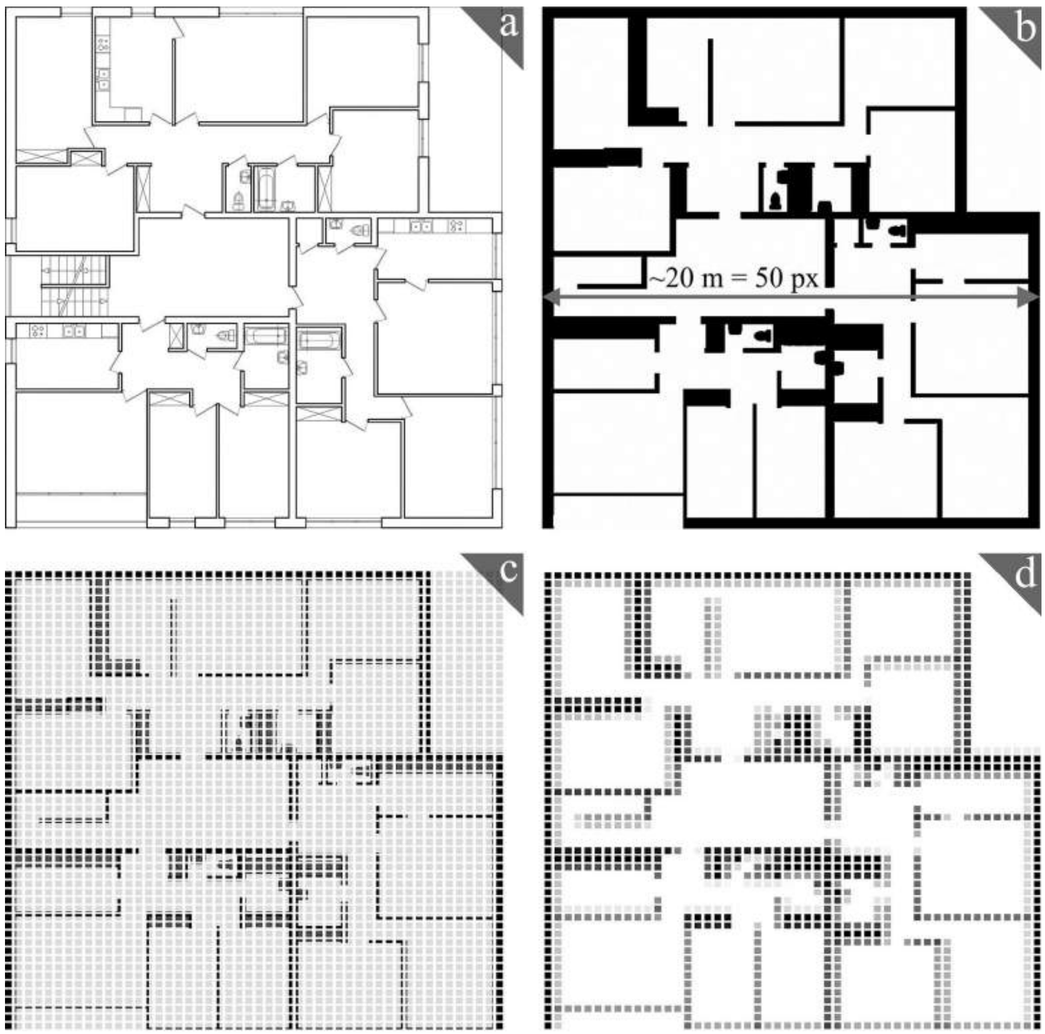

Figure 3: a). An example of an architectural CAD drawing of a three-apartment building. b). A preprocessed CAD drawing exported as a digital image. All inaccessible areas are shown in black. c). The original raster image has been normalized to the size of an agent. d). The values in each cell have been averaged.

Next, the layout has been binarized at the threshold controlled by the user, as shown in figure 4a.

Figure 4: a). Different binarization thresholds produce layouts of substantially different resemblance to the original. b). The binarization at 0.73 threshold compared to the original layout (shown in black in the background). Since the grid is very coarse, certain openings and corridors become too narrow, as indicated by the dotted circles. d). Manually corrected layout, so the widths of the corridors and openings are proper. The positions of exits and agents are shown in red and green, respectively. e). The heat map shows that although the exit area is perhaps a little narrow, the layout is rather safe.

In this case, the binarization at 0.73 threshold produced fairly good result, as shown in figure 4b: all the rooms are accessible and the walls are continuous. However, since some of the corridors and openings became substantially narrower than another, some further corrections were necessary. CZ allows to manually adjust the layout geometry as well as to place the exits and the agents, as shown in figure 4c. As in SPB, the number of agents was also exaggerated, but the exit size is realistic. A simple evacuation simulation has been performed and the resulting heat map is shown figure 4d.

4. Validation of the Crowd Dynamics Component

The purpose of CZ is to provide an intuitive and interactive environment particularly for designers to experiment with layouts at any stage of the design process. This section validates CDC by presenting several simulations and comparing them with models described in literature, commercially available software and empirical data.

4.1. Evacuation from a bottleneck

In CZ two types of update schemes are separately implemented in CDC: ordered sequential update (front to back - OSU) and random shuffled update (RSU). To examine the influence of these update schemes on the evacuation time the experiment described in [25] has been reproduced: 200 agents to leave a 25×25 cell room through a 3-cell wide exit. Figure 5a shows the positions of the agents after 60 and 100 time steps.

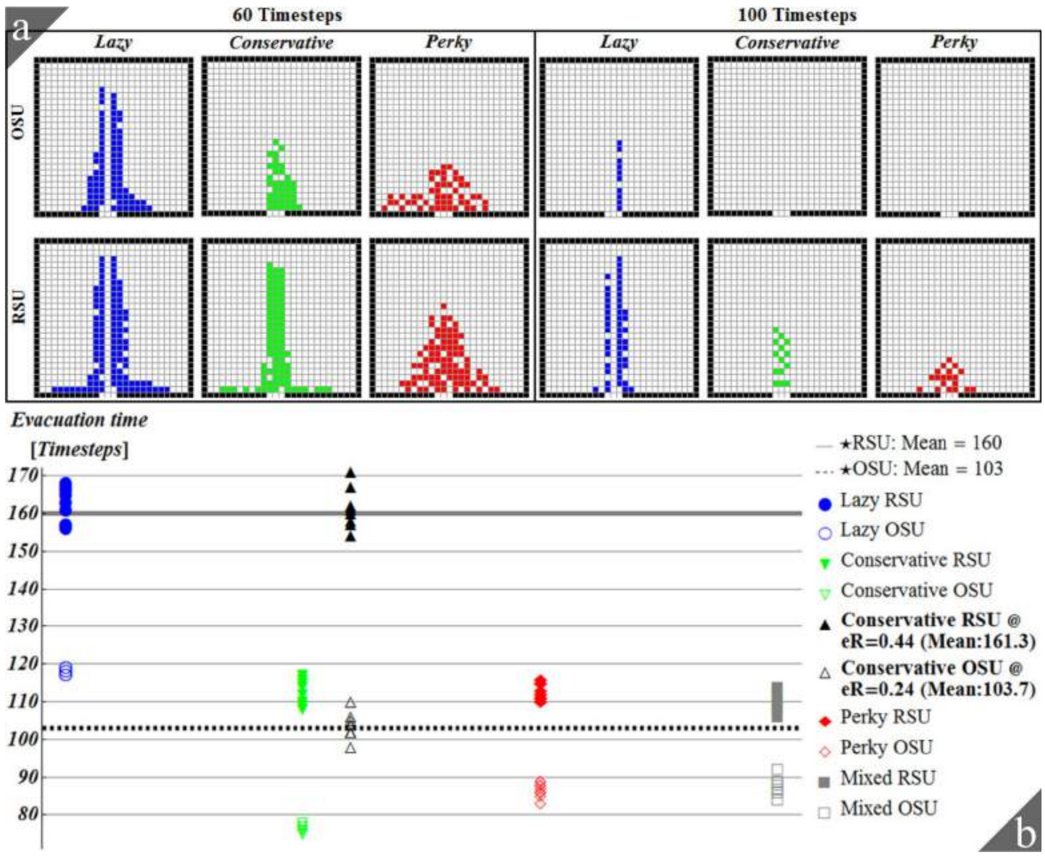

Figure 5: a). The positions of three types of agents after 60 and 100 time steps at OSU and RSU. Lazy, Conservative and Perky agents are shown in blue, green and red, respectively. b). Ten trials for each combination of the update scheme and perkiness. The starred items in the legend represent the results from [25]. Additionally the rightmost marks in the plot show the results for the equinumerous mixed crowd of Lazy, Conservative and Perky agents.

As figure 5 indicates, in the case of homogeneous crowd, Conservative agents at OSU evacuate from the room faster than the Perky ones. Lazy agents are the slowest and demonstrate the same queuing behavior as in [25], which is a consequence of their unwillingness to make a move which will not bring them immediately closer to the exit, in other words "they would rather wait in line". The evacuation time, however, can be prolonged by adding stochasticity to the agents' movements. This is implemented by error rate (eR). For $eR = 0$ there is no additional randomness, whereas for $eR = 1$ the agents perform a random walk. By adjusting this parameter it is possible to extend the evacuation time so it matches the results from [25], as shown in figure 5b. The same figure also conforms with the intuition, that disciplined agents (here Conservative at OSU) leave the space at the fastest rate. The same agents at RSU (not as ordered) are a little slower than mixed agents and comparable with the Perky ones.

To investigate the influence of eR on the agents' behavior, the same evacuation scenario at different values of eR has been performed, as shown in figure 6.

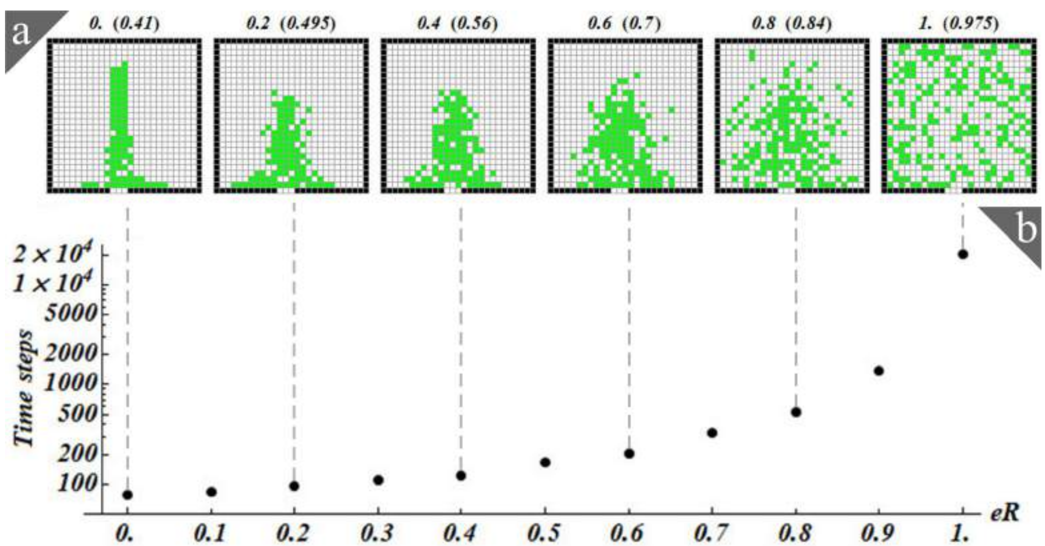

Figure 6: a). The positions in the room of the Conservative agents at RSU after 60 time steps at different eRs. The values of eR are shown above each sub-figure; the rates of remaining agents are shown in parentheses. b). The log plot of the time steps of evacuation of all the agents from the room as a function of eR (one trial for each value of eR). Although at $eR = 1$ the agents perform a random walk, they also finally leave the room, however, at approximately 250 times slower rate.

As figure 6a indicates, the rate of evacuation decreases with eR. In fact for approximately $eR > 0.5$ the evacuation time increases enormously, as shown in figure 6b.

4.2. Evacuation from a school: the comparison with PedGo

In this section, an experiment on a larger layout (School) imported from CAD is described. The results are compared with PedGo - a commercial software for simulating evacuations of pedestrians [16]. The size and character of the building are similar to the empirical case described in [17]. Here three groups of 200 agents are initially distributed in different sections of the School and are to leave the space through two exits at the shortest path as depicted in figure 7a. The actual positions, local speeds of agents and the heat map can be simultaneously monitored in CZ as shown in figure 7 b–g.

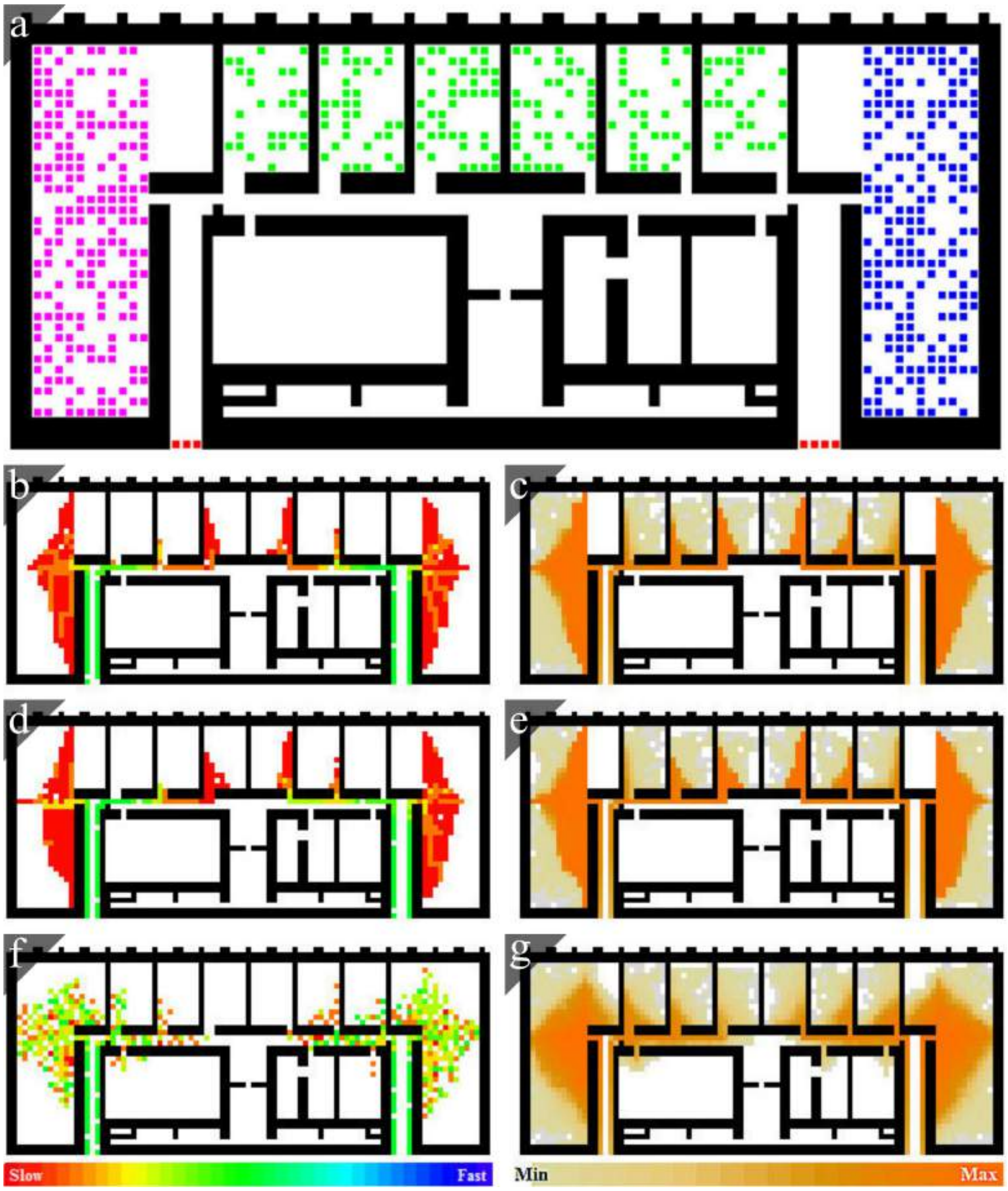

Figure 7: a). 600 agents are initially located in 3 sections. Agents in sections A, B and C are shown in magenta, green and blue, respectively. Walls and exits are shown in black and red, respectively. In the left column below: the positions with local speeds at the 60th time-step of the Lazy (b), Conservative (d) and Perky (f) crowds. In the right column: the corresponding heat maps (c - Lazy, e - Conservative, g - Perky).

500 trials were performed for the following 6 schemes: Lazy, Conservative and Perky crowds at OSU with $eR = 0$ and $eR = 0.25$. Since the standard velocity of a pedestrian is assumed after [19] to be 1.2 m/s, the maximal velocity in CZ is one cell per step, and each cell is 0.4 m, therefore 3 time steps last one second. Figure 8 shows the evacuation times compared with the results produced by PedGo.

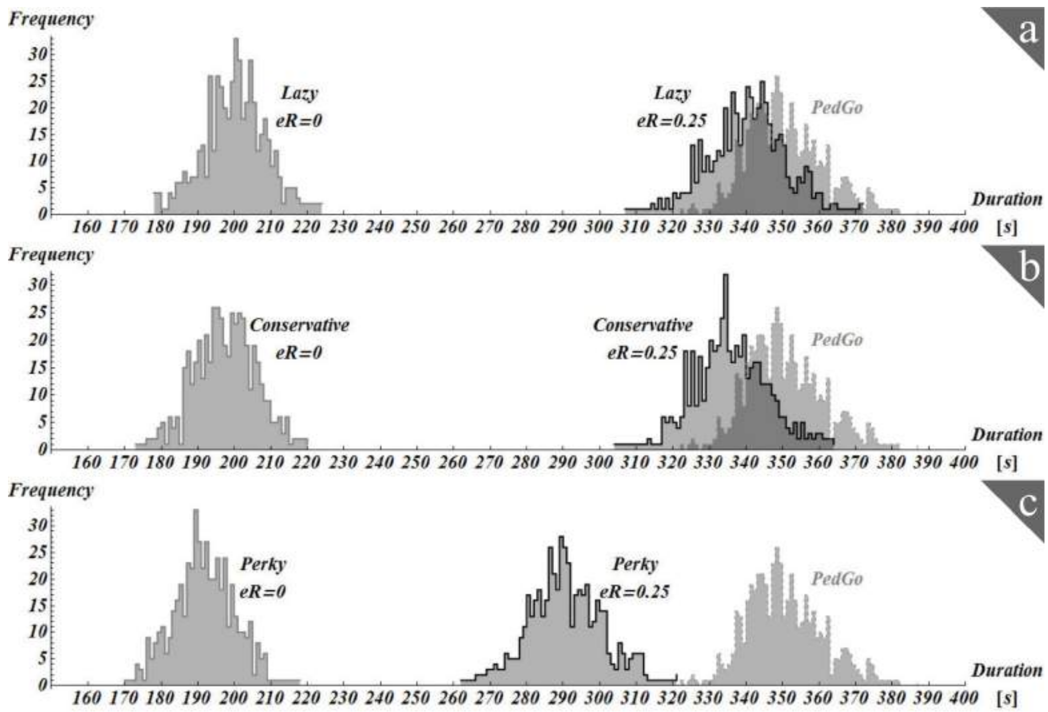

Figure 8: Histograms of evacuation times for 500 simulations performed with PedGo and CZ. From the top: a). Lazy, b). Conservative and c). Perky crowds.

As figure 8 indicates, the CZ agents at $eR = 0$ leave the School nearly twice as fast as the PedGo agents. Interestingly, in the case described in [17], the actual human pedestrians (schoolchildren) also "outperformed" the "standard simulation agents" by a similar factor. Nevertheless, the evacuation time of agents in CZ can be controlled quite precisely by the selection of agents' perkiness combined with the adjustment of eR. As the same figure indicates, it is fairly straightforward to set these parameters, so the CZ agents leave the given space in a similar time as the CA-model implemented in PedGo. It seems that not only Lazy, but also Conservative and Perky agents can, by increasing eR, to a certain degree emulate PedGo in this respect. However, the distribution of the results can have different characteristics. Moreover, the influence of eR is different for the three types of crowds. It is quite intuitive that Perky agents are the least affected by eR, since their behavior already has the highest degree of randomness of those three. Conversely, Lazy agents, whose movement in principle is the most stringent, is also the most sensitive to eR.

4.3. Evacuation from St Peter's Basilica: systematical experiments

Finally, more systematical simulations of the evacuation from SPB described in section 3.2 are performed and compared with the results produced by the floor field cellular automaton model (FFCA) [26, 27]. Interestingly, in the preliminary trial, the Lazy tactics was more efficient than Perky unlike in the case of the School evacuation. However, since the results are very close and there was only one trial for each type of agents, this does not have to be conclusive. Nevertheless, the Conservative crowd also here left the space at the lowest number of time steps (and therefore the simulation was the shortest) - thus this type was used in the actual simulations. In this experiment, the evacuation time was analyzed as a function of agents' density (ρ). Figure 9 shows the results from $\rho = 0.045$ (50 agents) to $\rho = 0.965$ (9750 agents).

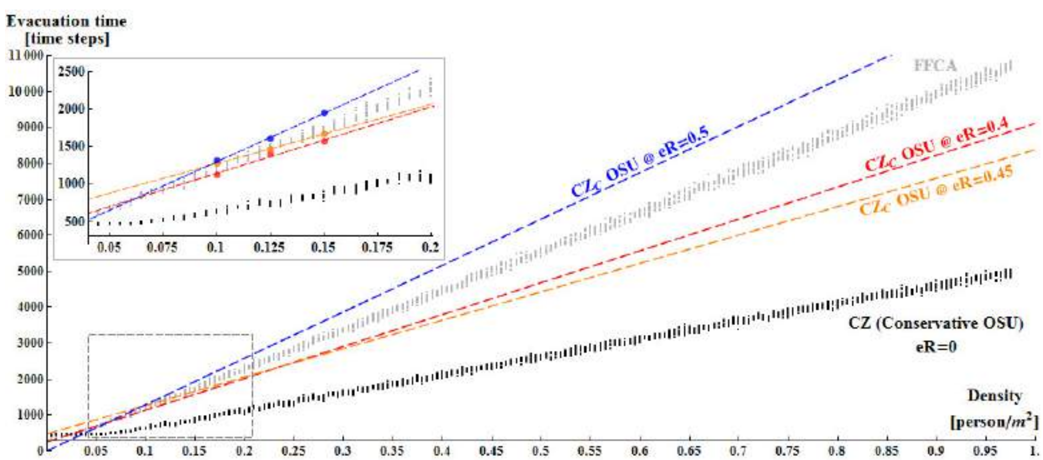

Figure 9: The evacuation time steps as a function of crowd density for CZ Conservative agents compared to FFCA. Additionally coarse extrapolations from three experiments for $eR: 0.4, 0.45, 0.5$ are shown in red, orange and blue, respectively.

As figure 9 indicates the CZ Conservative agents evacuate from the space over two times faster than the pedestrians simulated by FFCA. This is because in CDC basic model, the local interactions among agents are limited to a simple avoidance of collisions. In FFCA these interactions are more subtle, which makes the model more sophisticated but also - slower. Thus in CZ simulations the agents move with rather unnatural efficiency towards the exits, so the simulations run much faster than FFCA. However, the evacuation time grows approximately linearly with the density of the crowd, exactly like in the FFCA model. Another qualitative conclusion from this experiment can be drawn that the exit no. 1 (see figure 2) is the most congested and shows the bottleneck effect. It is also straightforward in CZ to extract the data from the heat map to analyze the exits usage by the agents, as shown in figure 2d. As that figure indicates, over half of the agents proceed to the exit no.1 (out of 15). This means that it is greatly overloaded, while exits 3, 4, 5, 6, 7, 8, 14 & 15 are hardly used at all. Such an information is crucial for designing safe layouts in respect to evacuation and CZ allows performing this kind of simulations on practically any digitally acquired image.

4.4. Evacuation from a bottleneck with empirical validation

In this subsection CDC is validated quantitatively against empirical data. For this purpose several simulations in a bottleneck-scenario as described in [28] have been performed. The experiment and simulation setups are shown in figures 10a and 10b–d, respectively.

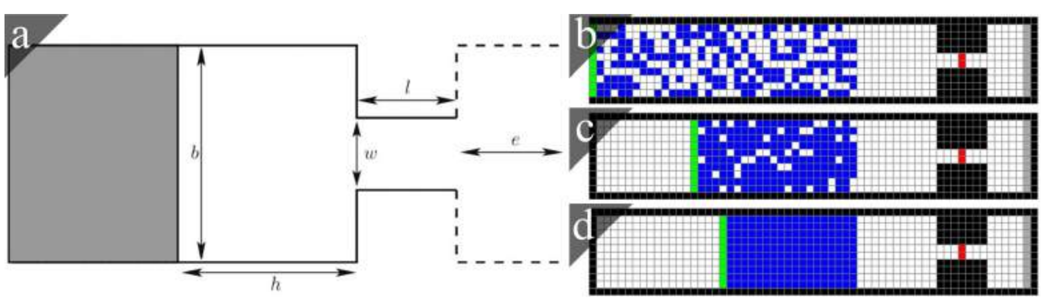

Figure 10: a). The bottleneck setup: the pedestrians move from the gray area through the bottleneck ($l = 2.8$ m, $h = 4.5$ m, $b = 4$ m and w variable). An adjacent area of length $e = 2$ m is added to consider the backward effect of leaving pedestrians on those still in the bottleneck. The right column b–d). The CZ simulation setups: $\rho_0 = 0.5\,\mathrm{m}^{-1}$, $\rho_0 = 0.8\,\mathrm{m}^{-1}$ and $\rho_0 = 1.0\,\mathrm{m}^{-1}$. The exits, initial positions of the agents and the measurement line are indicated in gray, blue and red, respectively.

The size of the holding area is changed systematically to allow different initial densities ρ. The flow through the bottleneck is measured as follows:

$$J = \frac{N}{\Delta t}, \tag{2}$$

with N the number of pedestrians and $\Delta t = t_{\text{last}} - t_{\text{first}}$ the time gap between the first and the last pedestrian passing the bottleneck at the measurement line.

In the CZ simulations the pedestrians have been initially randomly distributed in the holding area 4.4 m (11 cells) away from the opening of the bottleneck.

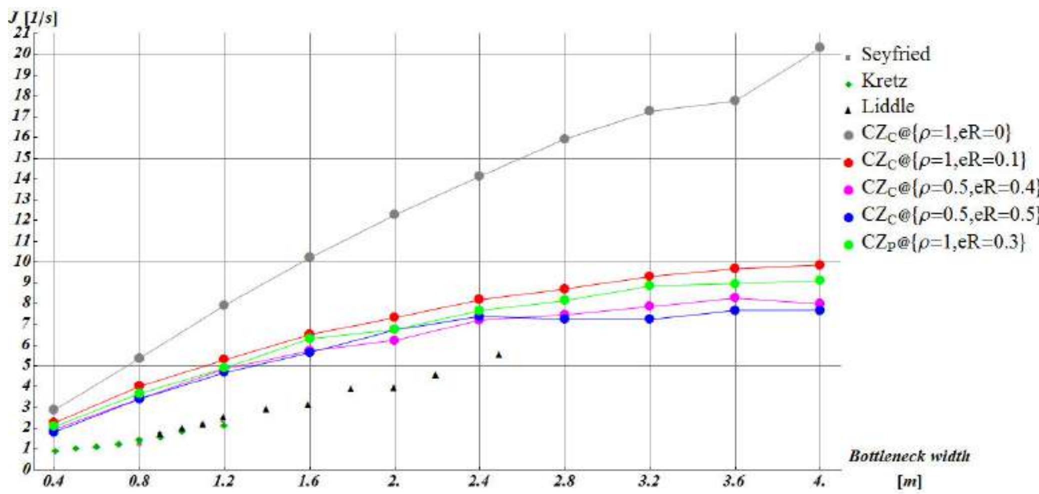

Figure 11: Simulated flow through the bottleneck with respect to its width w compared to the empirical data from Liddle et al. [28], Seyfried [30] and Kretz [29]. The results are averaged from 10 trials of $N = 180$ agents at different perkiness, eR-values and initial densities.

As figure 11 indicates, the flow rate of the Conservative crowd at $eR = 0$ grows rather continuously with the width w of the bottleneck (indicated in gray), which does not coincide with the empirical observations. However, the flow rate of the same agents with $eR = 0.1$, already shows a stagnation in the flow, which is more realistic since at the certain w the flow is practically unaffected by the bottleneck. The same figure indicates that, in general, the simulated flow is higher than the empirical data. However, it is possible to reduce the flow rate by increasing the value of eR and hence reducing the discrepancy to the empirical values. Introducing other types of neighborhoods such as Moore's and diversification of the walking speeds is currently under consideration for further validation.

5. Conclusions

The main contribution of Crowd-Z (CZ) described here is a practical, highly compatible and user-friendly platform for crowd simulations on any floor plan. Since working with CZ is very intuitive, it could be used not only by designers and specialists, but also other kind of users such as security staff e.g. at a football stadium, etc. For individual validation, free interactive demonstrations illustrating almost every CZ simulation described above are available at [24]. The dynamic animations are smoothly displayed on the CZ interface and users can interactively control a number of parameters of the running simulations. The model presently used in this framework is capable of reproducing quite realistic results, most importantly, however, due to the procedural architecture other CS models can be easily implemented in CZ. The information on each individual agent, such as position, local speed at each position, average speed etc., can be recorded for further analyses. Since each agent can be also attributed with other characteristics such as hierarchy, age, tiredness - it is also possible to simulate crowds with agents of various priorities, performances, grouping etc. Nevertheless, further empirical validation of CZ is also necessary and is planned for the upcoming works.

Acknowledgments

This is part of a postdoctoral project grant-funded by the Japanese Society for the Promotion of Science. The research is titled "Improvements of the Seniors' Quality of Life through Application of Innovative Computational Systems to the problems of Accessibility, Ergonomics and Housing & Living Environment". The authors thank Christian Eilhardt for his collaboration on the FFCA simulations and Hubert Klüpfel for granting the license to use PedGo.

References

References

[1] K. Hirai, K. Tirui, A simulation of the behavior of a crowd in panic, Systems and Control 21 (6) (1977) 409–411.

[2] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, Phys. Rev. E 51 (1995) 4282–4286.

[3] W.J. Yu, L. Chen, R. Dong, S. Dai, Centrifugal force model for pedestrian dynamics, Phys. Rev. E 72 (2) (2005) 026112.

[4] M. Chraibi, S.A. Kemloh, A. Schadschneider, Force-based models of pedestrian dynamics, Networks and Heterogeneous Media 6 (3) (2011) 425–442.

[5] P.G. Gipps, Simulation of Pedestrian Traffic in Buildings, Tech. Rep. 35, Institut für Verkehrswesen Universität (TH) Karlsruhe (1986).

[6] V.J. Blue, J.L. Adler, Cellular Automata Microsimulation of Bidirectional Pedestrian Flows, Transportation Research Record 1678 (1999) 135–141.

[7] N. Rajewsky, L. Santen, A. Schadschneider, M. Schreckenberg, The asymmetric exclusion process: Comparison of update procedures, J. Stat.Phys. 92 (1998) 151–194.

[8] T. Kretz, M. Schreckenberg, The F.A.S.T.-Model, in: S. El Yacoubi, B. Chopard, S. Bandini (Eds.), Cellular Automata – 7^{th} International Conference on Cellular Automata for Research and Industry, ACRI 2006, Proceedings, Vol. 4173/2006 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2006, pp. 712–715.

[9] J. Kerridge, J. Hine, M. Wigan, Agent-based modelling of pedestrian movements: the questions that need to be asked and answered, Environment and Planning B: Planning and Design 28 (2001) 327–341.

[10] S. Bandini, G. Vizzari, Regulation Function of the Environment in Agent-Base Simulation, Vol. 4 of Lecture Notes in Computer Science, Springer, 2007, pp. 157–169.

[11] J. Shi, A. Ren, C. Chen, Agent-based evacuation model of large public buildings under fire conditions, Automation in Construction 18 (3) (2009) 338–347.

[12] T. Kretz, S. Hengst, P. Vortisch, Pedestrian flow at bottlenecks - validation and calibration of vissim's social force model of pedestrian traffic and its empirical foundations, in: M. Sarvi (Ed.), International Symposium of Transport Simulation 2008, Monash University, Gold Coast, Australia, 2008, p. electronic publication.

[13] T. Kretz, Pedestrian traffic: on the quickest path, J. Stat. Mech.–Theory E., P03012 (2009).

[14] T. Kretz, A. and Große, S. Hengst, L. Kautzsch, A. Pohlmann, P. Vortisch, Quickest paths in simulations of pedestrians, Advances in Complex Systems, 14(5) (2011) 733–759.

[15] V. Schneider, R. Könnecke, Simulating evacuation processes with ASERI, in: M. Schreckenberg, S. D. Sharma (Eds.), Pedestrian and Evacuation Dynamics 2002, Springer, 2002, pp. 303–314.

[16] http://traffgo-ht.com/en/pedestrians/products/pedgo/

[17] H. Klüpfel, T. Meyer-König, M. Schreckenberg, Comparison of an Evacuation Exercise in a Primary School to Simulation Results, in: S. F. et. al. (Ed.), Traffic and Granular Flow '01, Springer, Berlin, 2002.

[18] R. Löhner, On the modelling of pedestrian motion, Appl. Math. Model. 34 (2) (2009)366–382

[19] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, A. Seyfried, Evacuation Dynamics: Empirical Results, Modeling and Applications, Encyclopedia of and System Science (5), ed. R. A. Meyers, Springer (2009).

[20] V.J. Blue, J.L. Adler, Flow capacities from cellular automata modeling of proportional spilts of pedestrians by direction. In: M. Schreckenberg, S.D. Sharma (eds) Pedestrian and Evacuation Dynamics. Springer, Berlin, pp. 115–121 (2002).

[21] S. Willard, General Topology, Addison-Wesley, Reading, MA, 1970.

[22] U. Weidmann, Transporttechnik der Fur Ausganger Transporttechnische Eigenschaften des Fur Fugngerverkehrs (Literaturauswertung). Schriftenreihe des IVT 90, ETH Zurich, 3 1993. Zweite, erganzte Auflage (in German) (1993).

[23] T. Kretz, C. Boenisch, P. Vortisch, Comparison of Various Methods for the Calculation of the Distance Potential Field, Pedestrian and Evacuation Dynamics 2008, 2010, pp 335–346.

[24] M. Zawidzki, Interactive demonstrations of Crowd-Z, http://zawidzki.com/Crowd-Z/

[25] C. Rogsch, A. Schadschneider, A. Seyfried, Simulation of Human Movement by Cellular Automata Models Using Different Update Schemes, Human Behaviour in Fire 2009 - Conference Proceedings, pp. 543–548, 2009.

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

[27] A. Kirchner, A. Schadschneider, Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics, Physica A, 312, pp. 260–276, 2002.

[28] J. Liddle, A. Seyfried, W. Klingsch, T. Rupprecht, A. Schadschneider, A. Winkens, An experimental study of pedestrian congestions: Influence of bottleneck width and length, Conference proceedings for Traffic and Granular Flow 2009.

[29] T.Kretz, A.Grňebohm, M. Schreckenberg, Experimental study of pedestrian flow through a bottleneck, J. Stat. Mech.–Theory E., 2006(10), pp. 10–14, 2006.

[30] A. Seyfried, O. Passon, B. Steffen, M. Boltes, T. Rupprecht, W. Klingsch, New insights into pedestrian flow through bottlenecks, Transport. Sci., 43(3), pp. 395–406, 2009.