Dynamic Simulation of Virtual Agents and Obstacles in Virtual Cities

Roman Mankovecký*
Supervised by: Fotis Liarokapis

HCI Lab, Faculty of Informatics
Masaryk University
Brno / Czech Republic

Abstract

The aim of this paper is to investigate a simple model for simulating virtual crowds for virtual environments and computer games. This model is based on the Social Forces model and enhanced using Monte Carlo simulation. The focus is given on the behavior of the actual simulation. In this model we can see interactions between virtual agents (virtual pedestrians) in two scenarios, walking towards a path and crossroads. In both scenarios, these agents are avoiding each other, avoiding obstacles and walls in different scenarios like crossroad or narrowed street. Moreover, users can move, scale or rotate these obstacles and place them interactively into the scene.

Keywords: crowd modeling, social forces model, Monte Carlo simulation, virtual agents

1 Introduction

Modeling human behavior of crowds is extremely interesting for a wide range of applications, ranging from games, film effects, simulators, evacuation simulations to urban planning. Understanding the movement of the crowd can help us to improve public places, we can expedite and facilitate the movement of citizens. However, pedestrians in real-life are reacting to a diverse and complicated stimuli that cannot be easily reproduced to computer simulations. As a result, most of the studies that have been up to now done focus on specific scenarios, such as cross-road crossings or evacuations. On the other hand, simple models can be computationally efficient. They are really good solution for simulations of the crowd [1].

Human behavior and interrelationships of humans can be divided into microscopic, mesoscopic and macroscopic [2]. In microscopic model, every single pedestrian is a simple entity, simple agent who is situated in the space in some specific time [3]. Mesoscopic models is accurately observing the behavior of individuals while relatively large number of individuals in the crowd is simulated [3]. On the other hand, in macroscopic models we can observe flows similar to gas or water flows [3], [18].

This paper aims in investigating the behavior of a mesoscopic model for agents walking towards a path and for crossroads. This is done by implementing the social forces model and agents are represented as cubes. To make their movement more realistic, the approach is enhanced by Monte Carlo simulation. Each agent has specific goal, destination, that has to reach. During this journey, the agent can collide

with different agents (cubes), obstacles (specific cubes agents cannot walk through) but also walls. The rest of the paper is structured as follow. Section 2 describes simple models used for modeling crowds in general. In chapter 3 we discuss implementation of this model in detail. Section 4 presents results and experiments done with this model and finally chapter 5 illustrates conclusions and future work.

2 Background

One of the first attempts to model crowd behavior was based on simulation of the motion of a generic population in a specific environment [4]. The individual parameters were created by a distributed random behavioral model which is determined by few parameters.

Nowadays, simple but effective crowd modeling systems include a decision-making component [5], pathfinding navigation [6], [17], [19] and local steering mechanics [17]. Another popular approach is the social forces model [18] or 'agent-based' model [19], [10], [11] which can be used to describe the forces of an virtual avatar (agent) to perform movement.

Monte Carlo method for simulating the dynamics of crowds has been used previously in different occasions. In one approach, Monte Carlo dynamics were used for the rearrangement of the group of agents [12]. The rules for the combined steps are determined by the specific setting of the granular flow from stampedes in panic scenarios to organized flow around obstacles or through bottlenecks.

Numerical simulations based on a Monte Carlo particle method demonstrated that when applied to crowds it

*mankoveckyroman@gmail.com

has the capability to qualitatively depict emerging behaviors and to provide a realistic description of the crowd dynamics in complex evacuation scenarios [13]. In another work, computation of escape probabilities using Monte Carlo method was presented for evacuation simulations in the context of the fire safety engineering [13].

An evacuation model using game theory combining the greatest entropy optimization criterion with stochastic Monte Carlo methods to optimize the congestion problem and other features of emergency evacuation planning was also proposed [15].

Moreover, macroscopic models simulate a particular pattern of moving crowd (Figure 1) [3]. Since the model is only interested in the overall look of the crowd, not interested in individual attitudes of individuals in the crowd and their behavior in the crowd, it is possible to simulate the abstract behavior of the large crowd.

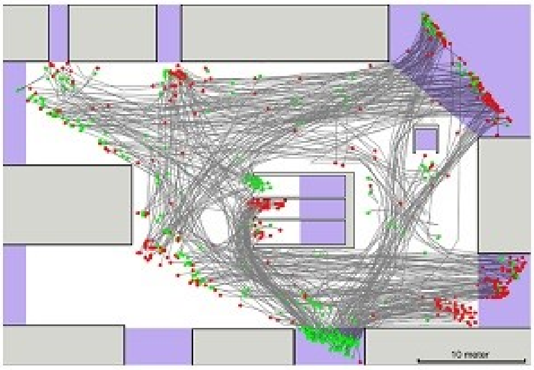

Figure 1: Flow (pattern) of agents [16]

The model uses basic concepts of physics, like the flow or movement of the particles. It does not include the individual behavior of the individual, does not deal with its characteristics and does not address the interaction or collisions with other individuals in the crowd. Thus it is engaged by simulating a large number of individuals in the crowd, respectively density of the crowd. These models have the advantage in terms of computational burden because there is no need to address the logic and behavior of individuals in the crowd. In particular, mesoscopic models are accurately observing the behavior of individuals while relatively large number of individuals in the crowd is simulated.

Mesoscopic models are based on cellular automata [3]. The area of movement has a regular square grid, where each cell may contain one or zero individual or an obstacle for one simulation step (Figure 2 on the right). Every individual has the opportunity to avoid with eight different directions, which are predefined and fixed (Figure 2 on the left). This behavior is not so suitable for a realistic simulation, but on the contrary, in this model, the individuals can move easily everywhere in the area because of pre-defined directions [16].

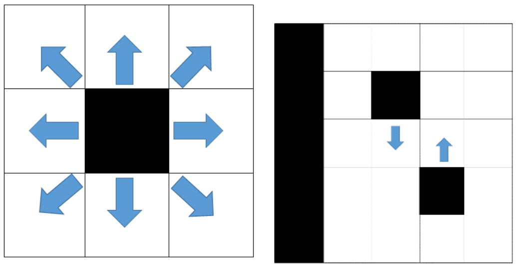

Figure 2: On left: Possible movement of one individual, black cube in the middle is an individual. On right: square grid, where we can see two individuals and a wall.

3 Implementation

3.1 Social force model

As mentioned earlier, to examine the behavior of the crowd simulation, the Social forces model was implemented allowing dynamic interactions between obstacles and agents using other stochastic techniques like Monte Carlo.

Cubes are representing the agents used in the project (Figure 3). Each agent has starting point and a point where it 'dies', which is the destination. Speed of agent is randomly chosen at start (0.5/ 1/ 1.5).

3.1.1 Speed

Speed is changing in some scenarios when agent has to slow down, because he has other agent with slower speed (v_p) ahead and he cannot overtake him in that specific time.

$$v(t) = \begin{cases} v^p(t), if\ v(t) - v^p(t) \leq 0.5 \\ v(t), if\ otherwise \end{cases} \tag{1}$$

Agents are moving forward with speed v in time t. We can define the force that is applied to agents $F^F_{(X,0)}$.

3.1.2 Changing directions

The next step required to add two rays (Figure 4a, red arrows) which are pointing ahead (leftRay, rightRay), so they can predict if something is ahead (i.e. other agents or obstacles). Next, we can define Y from $F^{CH}_{(0,Y)}$, which represents force applied while changing direction in specific time.

$$F^{CH}_{(0,Y)}(t) \rightarrow Y \tag{2}$$

$$Y = \begin{cases} 1, if\ leftRay \neq null \\ -1, if\ rightRay \neq null \end{cases} \tag{3}$$

Because of these rays, agents can change direction: if the left ray is hitting agent or obstacle and right one is hitting nothing, that means, agent will change direction to the right (Y = 1) and vice versa (T = -1).

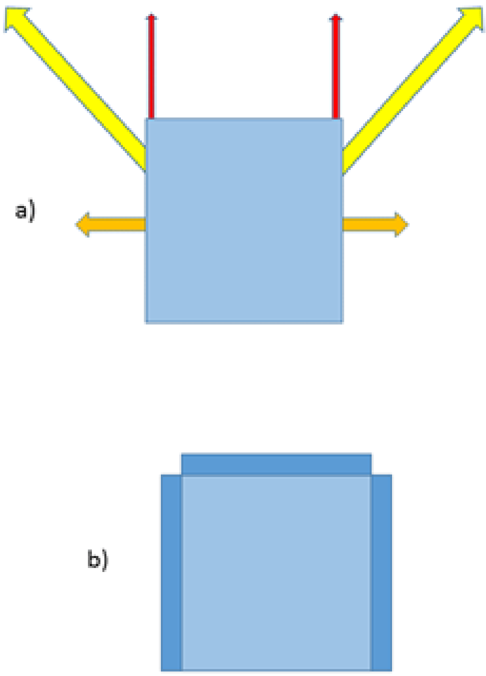

Figure 3: a) Blue cube is representing agent in real life, arrows in this picture are representing rays: red arrows are checking if something like obstacle or other agent is ahead, yellow arrows, “obstacle rays”, are checking obstacles and walls for better orientation and steering, orange arrows are mainly for checking, if something is around agent, so he can overtake. b) blue cubes around agent (cube in the middle) are colliders, they are checking if somebody or something is really close to the agent, they are creating comfortable zone.

3.1.3 Colliders

Agents have colliders as it can be illustrated from Figure 4b. These colliders are situated on the left, right and front side of each agent. They are checking, if other agents or obstacles or walls are nearby. Basically, these colliders are creating area around agents, where they are ’feeling comfortable’. It is repulsive force $F^R_{(X,Y)}$ that is applied on other agents.

3.1.4 Obstacle rays

Secondly, “obstacle rays” were added, with different angle $(1/4\pi, 3/4\pi)$, see Figure 7a yellow arrows. These rays are checking if on the right or left side is obstacle or wall. If for example an obstacle is on the right side, and agent will hit the obstacle, the agent will try to avoid it and he will steer left because right ray was hitting the obstacle. In cases where both these rays are hitting obstacle, only the first one is the key one, so the other ray has no function in these cases.

3.1.5 Overtaking

Each agent has also an overtaking parameter, so faster agents are overtaking slower ones. For this overtaking method two more rays were needed (Figure 4a, orange arrows). These rays (R^{left}, R^{right}) are checking, if the agent, obstacle or wall is on the left or right side. Agent will overtake only in cases he has the speed and space, in cases agent has the speed but not the space around, he will wait till there will be some free space. In waiting part he is checking with these rays, if there is or there is not the free space around. In cases we do not have speed, they will slow down only.

$$F^O_{(0,Y)}(t) \to Y \tag{4}$$

$$Y = \begin{cases} 1, if\ leftRay \neq null, \\ \qquad R^{right} = null \\ -1, if\ rightRay \neq null, \\ \qquad R^{left} = null \end{cases} \tag{5}$$

3.1.6 Total equation

All these forces influence a agent’s decision at the same moment, it can be assumed that their total effect is given by the sum of all forces:

$$F_{(X,Y)}(t) = F^F_{(X,0)} + \sum_P F^{CH}_{(0,Y)} + \\ + \sum_O F^O_{(0,Y)} + \sum_P F^R_{(X,Y)} \tag{6}$$

3.2 Monte Carlo

Monte Carlo is a stochastic method based on the use of random numbers and probability statistics to simulate problems [21]. First, we need to determine the probability density function, then perform random sampling from this function. Monte Carlo method allows us to examine complex system. Solutions are imprecise and it can be very slow if higher precision is desired.

In this work, Monte Carlo method was used to simulate random motion of agents. Total equation of the movement in this model is:

$$F_{(X,Y)}(t) = F_{(X,Y)} + F^{MM}_{(0,Y)} \tag{7}$$

Here we can see $F_{(X,Y)}$ which represent movement of the agent during the time t, $F^{MM}_{(0,Y)}$ is Monte Carlo force.

Monte Carlo simulation, forces agent to move with some probability to the left or right or nowhere. This means that during the time our agent has random movement.

$$F^{MM}_{(0,Y)} \to Y \tag{8}$$

$$array[10] \in (0,1) \tag{9}$$

$$Y = \begin{cases} random(-1,1), \ if array[\\ random(0, Length(array))] = 0 \\ 0, \quad otherwise \end{cases} \tag{10}$$

Before the start of simulation, it is possible to set the probability required, in the array. Every single agent is choosing movement each second, if he moves right (Y = -1) or left (Y = 1) or if he stays without moving

(Y = 0). So, in the end every frame agent will check and compare number 0 with a number in the array, if they are the same, will move to the right or left, if not will stay at the same position.

3.2.1 Monte Carlo Simulation

In this simulation, a scene with 100 agents with random speed (0.5 - 1.5) is shown. In first simulation, agents do not use the Monte Carlo method (Figure 4). The movement is occurring in a straight line, changing direction only in cases when agent wants to overtake slower ones. These results are not really realistic. However, in the second simulation (see Figure 5) we can see agents walking with Monte Carlo method. More randomness is in this case better and more realistic and it is better for overtaking as well.

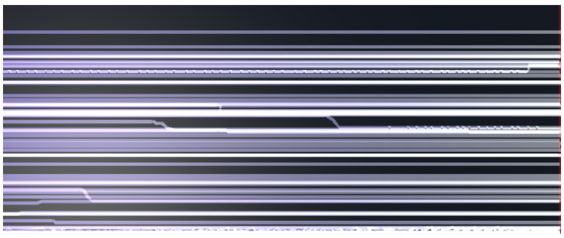

Figure 4: A scenario without Monte Carlo method. We can see lines, these lines are paths of the agents.

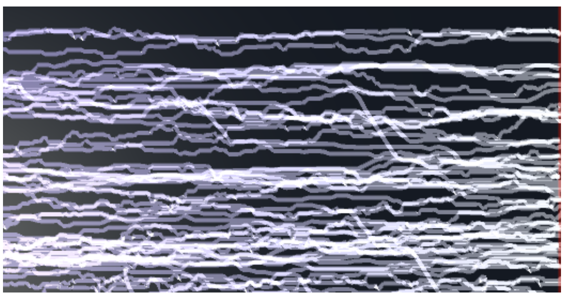

Figure 5: A scenario with Monte Carlo method. We can see lines, these lines are paths of the agents. We can also see the better randomness.

4 Results

In this paper, all these simulations were applied in two specific scenarios: path and crossroad as shown in Figure 6. In the path scenario, we can see some specific interactions (overtaking, grouping) but we can also see interactions between agents and obstacles. On the other hand, in crossroad scenario, we can observe interactions between agents walking from all sides, as well as path choosing.

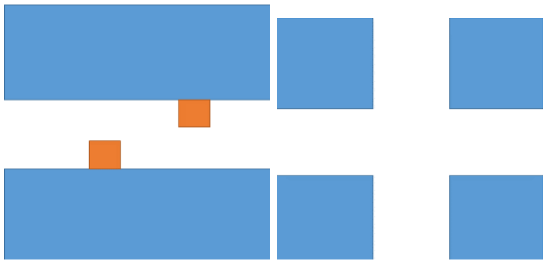

Figure 6: On left: path scene (with orange obstacles), on right: crossroad scene.

4.1 Grouping and tracking trajectories

There are various self-organization phenomena that lead to fascinating collective patterns of motion. For example, when agents are entering a corridor on two sides, we observe the formation of lanes of uniform walking direction [20]. Agents are grouping together, they are following each other, because of better movement, faster movement like we can see in Figure 7. They are avoiding other agents faster because of repulsive forces. This is classic scenario we can see during a normal day, walking the crowded street.

Figure 7: Grouping effect (blue circles), each agent is following the leader. Red agents are going from left to right, green ones are going from right to left.

With more agents, it is possible to track the trajectories of the agents. It is necessary to have more agents, so they can follow each other. As a proof of concept, 100 simulated agents are illustrated in Figure 8. The trajectories of these agents are shown, as well as the fact that they are following exactly the same way as the 'first' agent was moving. With even more agents, it is possible to observe bigger groups (bigger flocks). In conclusion, this model is showing that flocking is part of our life, we are doing it in normal crowded situations without notice.

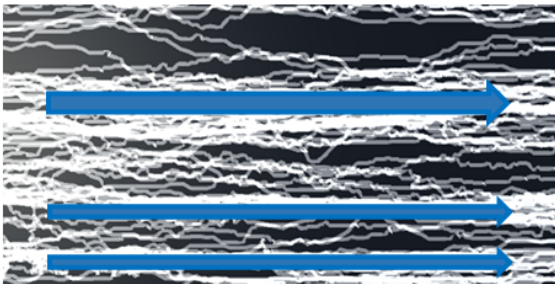

Figure 8: Movement tracking, we can see motion of agents, red arrows are indicating the motion of agents.

4.2 Frame rate measurements

In this section, the effectiveness of the simulation was measured. In particular, the aim was to quantify how many agents this model can simulate. In Figure 9, both tables are showing us frames per second (FPS) in different cases (with 50, 100, 500 and 700 agents in scene). Case 1 is without obstacles, without Monte Carlo method, case 2 is with obstacles but without Monte Carlo method, case 3 is with no obstacles but with Monte Carlo method and in the end we have case 4 with obstacles and using Monte Carlo method. In these measurements I used 2 obstacles. All these measurements have offset error $\pm$ 1 - 10 FPS, depending how many agents we are simulating.

In conclusion we can see that this model can simulate big number of agents in different scenarios. It is possible to observe that interactions with obstacles or using / not using Monte Carlo method are causing frame drops, but not so significant. All these measurements were computed on a laptop computer with specifications: Intel Core i7, RAM 8GB DDR3L, NVIDIA GeForce GTX 950M 2GB DDR3.

5 Conclusions

This paper has examined mesoscopic behavior of crowds for virtual environments and computer games based on the social forces model. In this model we can see interactions between virtual agents (virtual agents) in virtual city. These agents are avoiding each other, avoiding obstacles and walls in different scenarios like crossroad or narrowed street. In this model, user can move, scale or rotate these obstacles and place them into the scene.

In the future, it will be certainly useful to implement better steering. Obstacles can have different shapes, not only cubes and of course interactions with agents are sometimes not so realistic. Big bonus can be exchanging cubes with three-dimensional representations of humans. Finally, it will be of great importance to perform a user testing to assess how 'realistic' the behavior of the simulation is.

Acknowledgements

Authors would like to thank Human-Computer Interaction (HCI) Lab members for their support and inspiration. A video that demonstrates the functionality of the system can be found at: https://www.youtube.com/watch?v=4QaOdOVeYlk%20

Framerate measurement				
path	no MC, no obs.	no MC, with obs.	no obs., with MC	with obs., with MC
50	~160	~130	~150	~125
100	~80	~80	~80	~80
500	~20	~15	~18	~15
700	~5	~3	~5	~3

Framerate measurement				
cross	no MC, no obs.	no MC, with obs.	no obs., with MC	with obs., with MC
50	~100	~80	~80	~80
100	~45	~35	~40	~40
500	~7	~6	~8	~7
700	~4	~3	~4	~3

Figure 9: Framerates in different scenarios (path scenario on top, crossroad scenario on bottom.

References

[1] Stuart O'Connor, Fotis Liarokapis, Jayne Chrisina, *Perceived Realism of Crowd Behaviour with Social Forces*, Proc. of the 19th International Conference on Information Visualisation (IV 2015), IEEE Computer Society, Barcelona, Spain, 494-499, 2015.

[2] Dirk Helbing, *Models for pedestrian behavior*. arXiv preprint cond-mat/9805089, 1998.

[3] Kiran Ijaz, Shaleeza Sohail, Sonia Hashish, "*A Survey of Latest Approaches for Crowd Simulation and Modeling using Hybrid Techniques.*"

[4] Soraia R. Musse, Daniel Thalmann, *A Model of Human Crowd Behavior : Group Inter-Relationship and Collision Detection Analysis*. Computer Animation and Simulation 97, Part of the series Eurographics pp 39-51, 1997.

[5] Rachel McDonnell, Fiona Newell, Carol O'Sullivan, "*Smooth movers: perceptually guided human motion simulation*.". In Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 259-269. Eurographics Association, 2007

[6] Xiao Cui, Hao Shi, "*A*-based pathfinding in modern computer games.*". International Journal of Computer Science and Network Security 11, no. 1, pp. 125-130. 2011.

[7] Craig W. Reynolds, "*Steering behaviors for autonomous characters.*". In Game developers conference, vol. 1999, pp. 763-782. 1999

[8] Dirk Helbing, Peter Molnar. "*Social force model for pedestrian dynamics.*". Physical review E 51, no. 5, 4282, 1995.

[9] Xiaoshan Pan, Charles S Han, Kincho H. Law. *A multi-agent based simulation framework for the study of human and social behavior in egress analysis.* In: Proceedings of the ASCE International Conference on Computing in Civil Engineering. 2005.

[10] John Funge, Xiaoyuan Tu, Demetri Terzopoulos, *Cognitive modeling: knowledge, reasoning and planning for intelligent character*s. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1999. p. 29-38.

[11] Joo E. Almeida, Rosaldo Rosseti, Antnio Lea Coelho, "*Crowd simulation modeling applied to emergency and evacuation simulations using multi-agent systems.*". arXiv preprint arXiv:1303.4692 (2013).

[12] Francesco Piazza, *A simple Monte Carlo model for crowd dynamics.* Physical Review E - 82, 02611, 2010.

[13] Nicola Bellomo, Livio Gibelli, *Behavioral Crowds: Modeling and Monte Carlo Simulations toward Validation*. Computers and Fluids, 2015, In Press.

[14] Timo Korhonen, Simo Hostikka, Olavi Keski-Rahkonen, *A Proposal For The Goals And New Techniques Of Modelling Pedestrian Evacuation In Fires.* Fire Safety Science 8: 557-567, 2005.

[15] Wenhui Li, Jinlong Zhu, Huiying Li, Qiong Wu, Liang Zhang, *A Game Theory Based on Monte Carlo Analysis for Optimizing Evacuation Routing in Complex Scenes.* Mathematical Problems in Engineering, Volume 2015 (2015), Article ID 292093, 11 pages.

[16] Victor Blue, Jeffrey Adler, *Cellular automata model of emergent collective bi-directional pedestrian dynamics.* In: Proc. Artificial Life VII. 2000. p. 437-445.

[17] Cherif Foudil, et al, "*Path finding and collision avoidance in crowd simulation.*". CIT. Journal of Computing and Information Technology 17.3 (2009): 217-228.

[18] Penny Marno, "*Crowded-macroscopic and microscopic models for pedestrian dynamics.*". (2002).

[19] Miho Asano, Takamasa Iryo, Masao Kuwahara, "*Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour.*". Transportation Research Part C: Emerging Technologies 18.6 (2010): 842-855.

[20] Dirk Helbing, *SOCIAL FORCES: Revealing the causes of success or disaster.* Available from: http://futurict.blogspot.sk/2014/12/social-forces-revealing-causes-of.html/, Accessed at: 30/01/2016.

[21] Investopedia.com, *Monte Carlo Simulation*, [online]. Available: http://www.investopedia.com/terms/m/montecarlosimulation.asp, [Accessed: 3-January-2016].