Indoor Pedestrian Displacement Simulation

Annika Laumets
Institute of Computer Science
University of Tartu, Estonia
Supervisor: Amnir Hadachi
Email: annika.laumets@ut.ee

***Abstract*—Simulation tools are usually used to simulate traffic movement but pedestrian movement has not been studied very extensively. In this paper the model of pedestians moving inside a shopping mall in Estonia was programmed and integrated with SUMO. It was done to improve the movement of pedestrians inside the building, for example during an emergency evacuation.**

I. INTRODUCTION

There are quite many tools around to simulate the movement of vehicles and pedestrians outside to improve traffic light cycles, road networks and find junction bottlenecks. The authors of [1], for example, simulated the behaviour of pedestrians during 2008 Beijing Olympics Games using LEGION simulation software. In a result, the case study reduced the duration of the peak hour, prevented some probable safety problems and raised the level of service in the exits area. The problem is that indoor simulations have not received the attention that outdoor simulations have. One paper developed a "model for simulation of pedestrian movements in Hong Kong MTR stations" [2]. They collected data to get an origin-destination (O-D) flow matrix and used PEDROUTE, which is a pedestrian simulation model, to simulate the movements of pedestrians underground and in railway stations. With indoor simulation the movement of pedestrians could be improved, for example during an emergency evacuation. Different models can be used to simulate pedestrian evacuation and in [3] the authors "proposed an improved solution based on genetic algorithm" alongside "social force model lattice, gas model, and cellular automaton model". The idea for this project is to simulate the movement of pedestrians inside a building using an implemented model based on the work of [4] and [6].

II. BACKGROUND

The idea behind this paper was to do background research of different agent-based models for pedestrians and try to implement some of them to work with the SUMO simulation created during the last semester. The building chosen for this project is Ülemiste Keskus in Tallinn because it is a shopping center which means there might be large crowds inside and this paper might help to make the emergency evacuation plans better. For the project I used OpenStreetMap to get the map of the building and the corridors of the building which people can use to move around in, and also Simulation of Urban Mobility (SUMO) to simulate and visualize the movement of pedestrians.

III. RELATED WORK

A. Research

In [4] the authors present a new model to simulate the movement of pedestrians using both utility theory and social comparison theory. Their argument to describe a new model is that “most existing work simulates groups based on socio-psychological theories such as social comparison theory and five-factor personality theory”. However, these models only describe the dynamics of pedestrians’ socio-psychological states. The new model uses two steps – the first choosing a new group to follow or to stay with the old group and the second choosing the individual in the group to follow. Utility theory is used for the first step. It assumes that “person, even under a dangerous situation, can still make rational decisions” and it will choose the action with higher utility. For the second step, Festinger’s social comparison theory is used, which states that “when lacking objective means for evaluation of the current situations, such as emergency evacuations, people tend to follow with others that are most similar with them”. There are also multiple variables in the model that can affect the outcome – the distance to the group, sociality of the pedestrian agent, duration the agent has stayed with a group, the time the pedestrian agent can change the group (threshold), similarity of the agent which is bounded by Smin and Smax. Equations 1 - 5 show the formulas that are used to find which group an agent should follow or if the agent should stay with the old group. $DesiredDist$ and c are constants. $Duration$ is the number of time steps an agent has stayed with a group. After following a new group, $Duration$ will be set to zero. $Threshold$ is the maximum number of time steps during which an agent can join another group.

Equation 1 shows the distance between agent i and group GP_j using the closest agent j who belongs to group GP_j. U_f is the maximum utility of joining other groups and equations 2 and 3 show that the closer the group is, the more likely the agent will join this group. U_s is the utility to stay with the current group. $Sociality_i$ shows how social agent i is. The greater the value is, the more likely the agent will join another group.

In Eq. 6 - 10 a and b are constant numbers. i and j are agents. v_i and v_j are the velocitis of agents i and j. "v_l and v_m are the vectors pointing from from i’s current position to the position of the selected member and i’s destination, respectively" [4]. $Similarity$ is the agent’s similarity value,

$$Dist_{i,GP_j} = Dist(i,j) \quad (1)$$

$$t = DesiredDist/Dist_{i,GP_j} \quad (2)$$

$$U_{i,f} = max(t * e^{Sociality_i*(1-Duration/Threshold)}) \quad (3)$$

$$U_{i,s} = c * e^{Sociality_i*Duration/Threshold} \quad (4)$$

$$GroupToJoin = \begin{cases} \text{Group with } U_{i,f}, & \text{if } U_{i,f} > U_{i,s} \\ GP_s & \text{otherwise} \end{cases} \quad (5)$$

Fig. 1. Equations to find the group to follow

$$Similarity_i = Sociality_i*D_1*(a*D_2+b*e^{1-Dist(i,j)/Dist}) \quad (6)$$

$$D_1 = \begin{cases} 1 & \text{If } v_l * v_m \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

$$D_2 = \begin{cases} 1 & \text{If } v_i * v_j \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

$$MostSimilarId = max(Similarity_1, \ldots, Similarity_n) \quad (9)$$

$$AgentToFollow = \begin{cases} Agent \text{ with } MostSimilarId \text{ If} \\ MostSimilarId \text{ exists} \\ \emptyset \text{ Otherwise} \end{cases} \quad (10)$$

Fig. 2. Equations to find the individual to follow

which finds the similarity between agents i and j. Moreover, the $Similarity$ value is bounded by predefined minimum value $Smin$ and maximum value $Smax$. The formula to calculate $Similarity$ is defined in Eq. 6. It can be seen that the more social the agent is and the closer the distance between agents i and j, the more likely agent i will join the group where to agent j belongs to. The moving direction is also important, because an agent will prefer to follow an agent who moves in the same direction. The agent with the maximum $Similarity$ value will be selected to follow if it exists. Otherwise the agent will stay with the old group.

In addition, the pedestrian agent has "a set of attributes which characterize its internal states" and three behaviour models – random movement, obstacle avoidance and maintaining group. Random movement behaviour is used to generate a random destination for the pedestrian and move the agent there. Obstacle avoidance behaviour ensures that the pedestrian does not collide with obstacles, other agents and groups. The agent follows a member in a selected group with maintaining group behaviour. The experiments made show that the developed model can simulate dynamic grouping. Moreover, a second experiment was made trying different threshold values to see how it affects the group changing rate. The outcome was that the bigger the threshold, the more times agents change groups. In conclusion, a two-layer model was developed where, firstly, the pedestrian agent chooses which group to follow – follow a new group or stay with the old group – and then the agent who is most similar to himself is chosen to follow.

In [6] egress safety was analyzed by using different situations. Since building codes "only provide basic guidelines and are not exhaustive" and each building is different, simulations should be conducted to test the safety and operation of egress. A multi-agent system was used instead of system theories of organization because the latter is quite abstract. Multi-agent systems also have a large number of individuals (agents) with complex behavioural patterns. Furthermore, agent behaviour can be assessed by the user with multi-agent system simulations. Force modeling was used in the model to assess the casualties in different scenarios. The experiments included "normal and emergency situations to show that the crushing force between individuals significantly affects the results". In addition, there were three different cases of emergency evacuation – no barriers, barriers and various egress distances. In a normal crowd evacuation there were no casualties and the total evacuation time for 2m wide exit was 93.9 seconds, for 4m wide exit it was 59.2 seconds and for the 8m wide exit it was 43.3 seconds. In the emergency evacuation with no barriers the fatality probability for 2m wide exits was 75%, less than 10% for 4m wide exits and 0% for 8m wide exits. The evacuation time for 2m exits was 53.5 seconds, for 4m exits 32 seconds and 21.8 seconds for 8m exits. With barriers the evacuation times were only slightly increased, but the fatality probability was significantly increased. The last experiment tested various gaps between exits. The 2m, 4m and 8m exits were distanced 10, 30, and 50 meters apart. The test results showed that for 2m wide and 4m wide exits the exit distances did not have an obvious effect. However, for 8m wide exits the evacuation time for 30m distance between the exits was a bit smaller than for the other distances. In conclusion, it seems that the 30m exit distance is the most reasonable solution.

These models are to be implemented and integrated with the simulation created before to try to simulate the pedestrian movements during an evacuation a bit more realistically. Since the model to be implemented is quite big, it was taken into account that the implementation would not be finished during this paper.

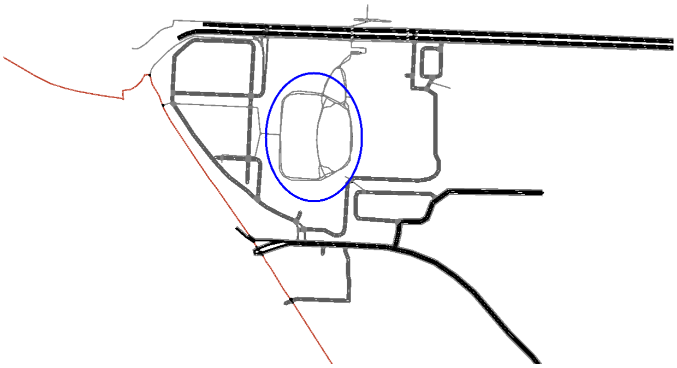

Fig. 3. Converted map from OpenStreetMap

B. *Previous work*

The author has already made a simulation using SUMO and data from OpenStreetMap. OpenStreetMap was used to get the geographic data about Ülemiste Keskus. The data in OpenStreetMap is provided by volunteers who enter "data about roads, trails, cafés, railway stations, and much more, all over the world" [7]. There are other projects related to OpenStreetMap that specifically map indoor data, e.g. OpenLevelUp and ID-indoor [8]. Unfortunately it was impossible to export the data from these sites into a .osm format so that SUMO could convert it into a suitable network file. Therefore, it was best to use the data from OpenStreetMap.

Simulation of Urban MObility (SUMO) is an open source road traffic simulation which can be used on different operating systems [9]. It can also handle large road networks. It has been available since 2001 and it is "mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center" [10]. With SUMO it is possible to visualize road vehicles, public transport (buses, trains, etc.), bicycles and pedestrians. Furthermore, SUMO is a suite of tools - it includes network generation tool (netgenerate), network convertion tool (netconvert) and tools to convert different data (O/D matrices, turn percetages, cross-section detectors) into trips. Also, it is possible to run the simulation with two options. The first option is the command line tool and the second option is the GUI application [9].

The output from OpenStreetMap is in .osm format which can be converted into .xml using SUMO's command line tool `netconvert`. The converted network map be seen in Figure 3. The blue ellipse in the map shows the location of Ülemiste Keskus. The roads inside the ellipse are corridors inside the shopping mall.

Since `netconvert` only converts nodes and edges into a network, it looses buildings, water, forests, parks, residential, commercial and industrial areas, shops and amenities. They can be added back with the `polyconvert` command which "imports geometrical shapes (polygons or points of interest) from different sources, converts them to a representation that may be visualized using SUMO-GUI" [11]. The converted map with polygons added can be seen in Figure 4.

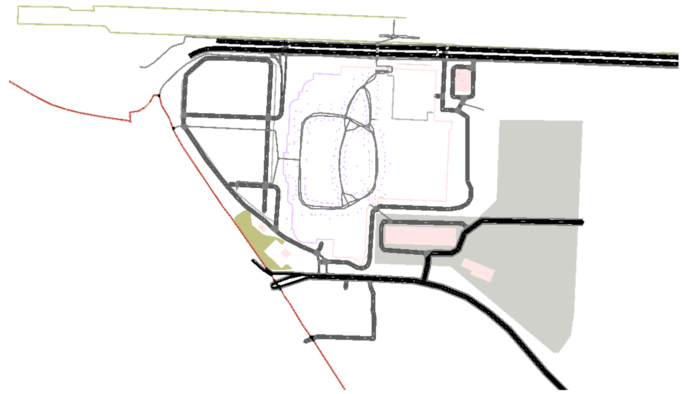

Fig. 4. Converted map with polygons

Fig. 5. Snapshot 1 of the running simulation

Fig. 6. Snapshot 2 of the running simulation

After the map is ready to be used, it is possible to move on to the simulation part of the implementation. SUMO's package includes 2 Python scripts that can be used to generate random trips. The first one is specifically for generating random pedestrian flow, but the downside is that it needs separate edge and node files. The network map that was generated from the OpenStreetMap data is one file that specifies both edges and nodes so it could not be used. The second file is for generating random trips. The options include specifying the vehicle class (bus, rail, pedestrian, bicycle, etc.), begin and end time of the simulation, weight edge probability by number of lanes or length. Moreover, there is a specific option `-pedestrians` which creates a trip file for people and not for vehicles.

In Figure 5 and Figure 6 the simulation is running. The blue numbers represent the id's given to pedestrians. When comparing Figure 5 and Figure 6, it can be seen that the pedestrians move around and some finish their route, i.e they are removed from the simulation. Pedestrians also appear on the edge of the roads because only a selected area of the map is used in the simulation.

IV. Implementation

The author is using Python to first implement the agent-based model from [4]. Python is used because SUMO uses Python generated code for the random trip generator and it is easy for the author to use this language. A class for *Agent* was partly implemented. There was a problem with the network. Firstly, the author didn't know how to read in the the network generated by SUMO. After examining the SUMO file *randomTrips.py*, the same code was used to read in the network map as in the previously mentioned file. The author also had some questions if the sidewalks should be two-way or if the pedestrians could use one-way sidewalk for moving in both directions.

V. Conclusion

This paper summarized two ideas to be implemented in the future to be used with SUMO. The implementation of the model from [4] should be finished and tested. Next, the design from [6] should be taken into account to see if the the exit design of Ülemiste keskus should be changed to make the casualty rate during evacuations smaller.

References

[1] W. Zhonghua, W. Yongyan, and L. Yang, "Application of pedestrian simulation software in beijing olympic games a case study of olympic badminton venue," in *2009 WRI World Congress on Software Engineering*, vol. 2, pp. 254–261, May 2009.

[2] J. Y. S. Lee, W. H. K. Lam, and S. C. Wong, "Pedestrian simulation model for hong kong underground stations," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pp. 554–558, Aug 2001.

[3] Z. Dandan, Y. Xiaoqing, Z. Yifan, and Y. Mingyang, "Pedestrians evacuation simulation based on genetic algorithm," in *2015 International Conference on Smart and Sustainable City and Big Data (ICSSC)*, pp. 78–82, July 2015.

[4] F. Qiu and X. Hu, "Modeling dynamic groups for agent-based pedestrian crowd simulations," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, pp. 461–464, Aug 2010.

[5] "Pedsim - a pedestrian crowd simulation." Last accessed: 28.11.2016.

[6] Q. Zhang, G. Zhao, and J. Liu, "Performance-based design for large crowd venue control using a multi-agent model," *Tsinghua Science and Technology*, vol. 14, pp. 352–359, June 2009.

[7] "Openstreetmap web page." Last accessed: 15.11.2016.

[8] "Openstreetmap wiki - indoor mapping." Last accessed: 15.11.2016.

[9] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker, "Recent development and applications of sumo - simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. (3&4), pp. 128–138, December 2012.

[10] "Simulation of urban mobility - sumo. institute of transportation systems." Last accessed: 15.11.2016.

[11] "Sumo wiki - polyconvert." Last accessed: 15.11.2016.