A Dynamic Exit Choice Method for Real-Time Indoor Evacuation Scenarios

Oner Barut
Hacettepe University
onerbarut@cs.hacettepe.edu.tr

Murat Haciomeroglu
Gazi University
murath@gazi.edu.tr

Cumhur Y. Ozcan
Hacettepe University
cumhuryigitozcan@cs.hacettepe.edu.tr

Hayri Sever
Hacettepe University
sever@hacettepe.edu.tr

***Abstract*—Most of the evacuation simulations focus their attention to the accuracy of the simulation. Therefore, their models fail to provide real-time performance which is indispensable for almost all video games. In this paper, we propose a real-time crowd simulation model specifically designed for indoor evacuation scenarios to be used by entertainment systems. While agents in our model perform local collision avoidance using an agent-based steering technique, the optimal exit gates and the global paths to these gates are determined efficiently by the proposed system. We represent our simulation environment as a uniform grid and we perform Dijkstra path finding for all exit gates which computes shortest paths from all exit gates to all cells. Then, for each grid cell, we choose the exit gate having shortest path within all gates and assign this gate to the cell which means all agent in the grid cell will evacuate the area using the selected exit gate. Later on, we reverse the previously found paths between every grid cell and their associated exit gates in order to calculate guidance velocity of the agents in the cells. Our model takes advantage of the limited goal point availability by only performing one Dijkstra path finding per exit gate to further increase the efficiency of the simulation. Test results indicate that our method not only manages to balance the agent load on the exit gates despite non-uniform position distribution of the agents in the test scenarios, but also achieve lower average and maximum evacuation times than mostly used nearest gate selection approach.**

I. Introduction

Crowd simulation can be introduced as imitating motion and behavior of a large number of real entities such as humans or animals as close as possible to reality. This replication includes navigating vast number of virtual characters from their initial positions towards their goal positions in a virtual environment. In addition, a real-time simulation should be as efficient as possible to run on a consumer level hardware, especially in games.

Although both movie and video game industries intensively use the crowd simulation models, evacuation planning is another important application area of the crowd dynamics. Evacuation planning can be defined basically as measuring and improving the time required to move all the agents inside a building, facility, vehicle, etc. to a safe place in an emergency situation. In addition to being too expensive and time consuming, using humans for evacuation experiments may be very dangerous and be concluded with casualties. Therefore, evacuation simulations which perform these experiments on a virtual environment with virtual agents are commonly used.

Evacuation simulation of crowds has been studied over the past decades. Numerous evacuation models have been proposed to perform accurate computer simulations. Kuligowski et al. [1] published a comprehensive review of existing models. As a summary, according to the current literature, cellular automaton models [2], lattice gas models [3], social force models [4], fluid dynamic models [5] and agent-based [6] models are employed to navigate the agents in evacuation simulations. These models may be divided into two groups; microscopic and macroscopic in terms of scale. Microscopic models deal with the individual agents while macroscopic models consider the agents in the crowd as a whole. Another possible categorization of these models which labels them as discrete and continuous is based on time and/or space.

In evacuation simulations, one the most fundamental parts is determining optimal exit gates for each agent to perform the immediate evacuation of all the agents in simulation environment. Kuligowski et al. [1] performed a detailed comparison of the exit selection strategies of the existing evacuation simulation systems. According to their study, many evacuation simulations employ shortest distance metric between agent positions and exit gate positions in order to decide the minimum cost exit gate for every agent. One of the major drawbacks of this approach is that, when agents in the evacuation scenario are not distributed uniformly in the simulation environment, a large number of agents will decide to navigate towards the same gate. In that case, some of the exit gates will be congested while the others are not used intensively. As a result of the unoptimal exit gate decisions of the agents, average evacuation time and maximum evacuation time of the simulation tend to be higher. Also in a video game, agents would seem unintelligent.

On the other hand, Kuligowski et al.s [1] work also indicates that there are evacuation models that consider not only the distance but also width, congestion level, awareness, signage of the exit gates in order to plan the evacuation route of the agents. The subject evacuation models either determine the minimum cost exit gates conditionally or seek for the minimum cost path through the exit gates bearing in mind the above parameters. Recently, Ehtamo et al. [7] performed the exit route choice based on a game theoretic approach which tries to maximize the utility of all agents and Guo et al. [8] employed a logit-based method for exit choice.

Although there are successful evacuation models in the literature, these techniques are computationally too intensive for a real-time simulation. Our technique is designed for real-time performance therefore, the proposed model is not intended for accuracy but creating plausible evacuation scenarios to be used by entertainment industry.

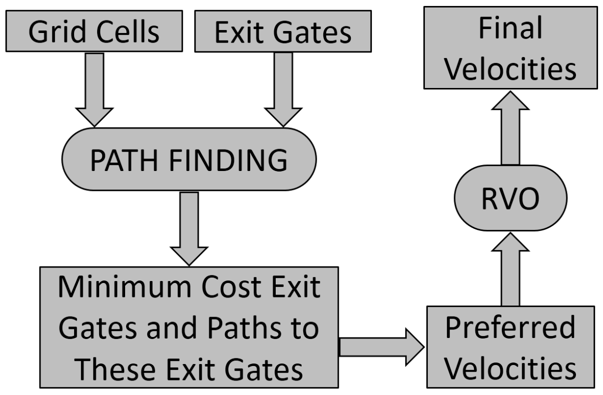

Fig. 1: System overview of the proposed technique that uses path finding for determining exit gates. A path finding process is performed per exit gate. Consequently, minimum cost exit gate of each grid cell and path to this gate from the cell are determined. Benefiting from the path information of each cell, a preferred velocity(heading towards the determined minimum cost exit gate) is calculated for all agents in the cell. RVO is used in order to calculate final velocities of all agents via utilizing their preferred velocities.

II. System Overview

In the proposed model, we developed an evacuation system for indoor simulation environments with a limited number of exit gates. We employed a uniform grid structure which consists of square cells that covers the entire simulation area. Each grid cell contains the information about an optimal predetermined exit gate that has minimum cost to evacuate all agents currently in that cell. Every cell also contains the whole path information to the determined minimum cost exit gate.

The proposed method utilizes Reciprocal Velocity Obstacles (RVO) [9] model for local navigation of virtual agents. In every simulation step, agents query their current cell and head towards the minimum cost exit gate which their current cell has already detected (see Section III). For this purpose, agents use the path information stored in cells and determine a preferred velocity vector considering first three cells along their path. They calculate average center positions of the first three cells (except from the one the agent is currently at) and construct a vector directing from the center of the current cell to the average center of the following cells. Then agents normalize their vector and scale it by their intended speed to obtain preferred velocity vector. By using the preferred velocity vectors of each agent as input, RVO calculates a final velocity vector for each agent as output. The agents use these collision-free velocity vectors while navigating towards the exit gates. A general overview of the proposed system is illustrated in Figure 1.

III. Determining Exit Gates

The proposed technique determines the minimum cost exit gate for all cells by utilizing Dijkstra path finding algorithm. Basically, instead of searching for paths from cells to exit gates, each gate determines a path from itself to every grid cell. Considering we have a very small number of exit gates relative to the number of grid cells in almost all evacuation scenarios, it is obvious that the proposed technique will require very few path finding operations independent of the simulation environment size. However, it should be also noted that the number of path finding operations is linearly proportional to the number of exit gates the evacuation environment has.

The presented evacuation system performs path finding operations for all of the exit gates once per each second. Each gate determines a cost to each grid cell based on the cost function (Equation 1). Each gate compares its cost (for each cell) with the cost of other gates. If the cost from that gate to the current cell is smaller than the other gates, the current cell updates its cost and path information. After finishing the path finding operation for all exit gates, each cell is aware of its minimum cost exit gate and the path going towards that exit gate.

$$Cost = TravelCost + WaitingCost \tag{1}$$

Benefiting the cumulative nature of the Dijkstra cost function, travel cost which corresponds the cost of passing from the current cell, that Dijkstra has arrived, to the neighboring cell of the current cell is given in Equation 2. This cost function calculates travel time in seconds which will be required while passing from the current cell to the neighboring cell. In Equation 2, $\left|\vec{P_C P_N}\right|$ indicates the distance between the current cell and one of the neighboring cells, S_A indicates the average travel speed of the agents on the simulation.

$$TravelCost = \frac{\left|\vec{P_C P_N}\right|}{S_A} \tag{2}$$

Travel cost computed using Equation 2 is used in Dijkstra path finding operations to decide minimum cost paths from exit gates to grid cells. However, to determine minimum cost exit gates for all grid cells, another cost, named waiting cost, is also included. Waiting cost takes into account the evacuation time (in seconds) of all agents who are on the grid cells that has already determined the same gate as minimum cost exit gate. For this purpose, each exit gate stores total number of agents that will be evacuated via itself and when a grid cell determines an exit gate as minimum cost exit gate, the exit gate updates its total agent number by aggregating the number of agents inside the grid cell. The cost function which computes the waiting cost is given in Equation 3. In Equation 3, N_A indicates the total number of agents that will be evacuated from the same gate, S_E indicates the evacuation speed (agents per second) of the exit gate. Instead of performing a dynamic computation of the evacuation speed variable of the exit gates, we heuristically choose a constant value (one agent per seconds) for performance considerations.

$$WaitingCost = \frac{N_A}{S_E} \tag{3}$$

One can think that grid cells might determine two different minimum cost exit gates among two consecutive path finding progresses in a one-second period and such a case might cause oscillations of agents although it is not observed in

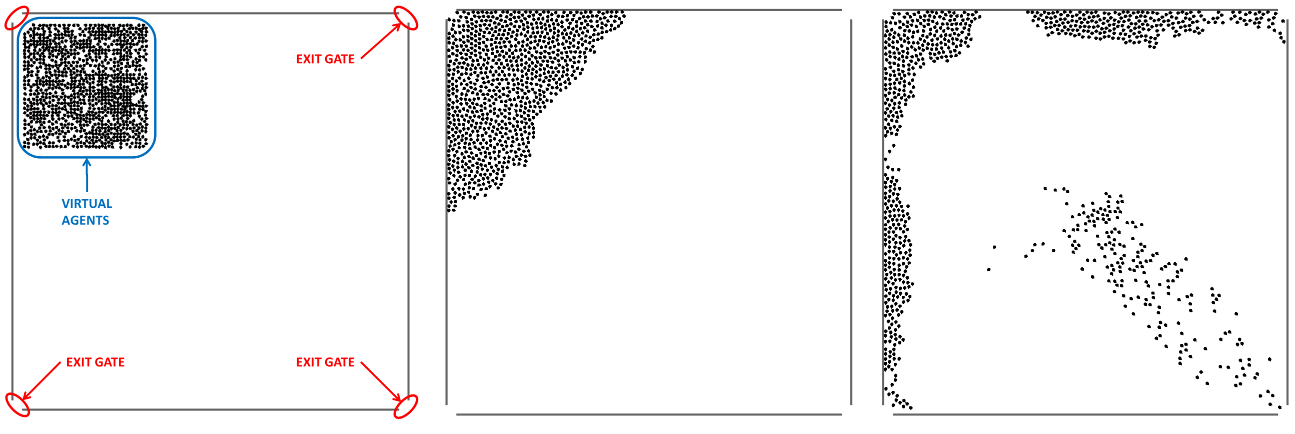

Fig. 2: (a) Top view of the initial simulation environment which has four exit gates and virtual agents standing close to the upper left exit gate. (b) Snapshot taken shortly after the beginning of the first scenario which agents choose nearest gate(the one at the upper left corner). (c) Snapshot taken shortly after the beginning of the second scenario that determines minimum cost exit gates via path finding.

our test scenarios. One possible solution to that would be introducing a penalty cost when a grid cell tries to change its previously determined minimum cost exit gate. Thus, potential oscillations of moving agents will be avoided successfully and with insignificant additional computational cost.

IV. Test Scenarios & Results

In order to evaluate the proposed evacuation system, we have generated two test environments which represents common indoor evacuation cases. Both test environments consist of a square shaped closed environment (32×32 square grid cells with 4 meters edge length). Agents' intended speeds are assigned randomly between $1 - 3m/s$.

The first test environment is a large hall that has four equally sized exit gates which are positioned at each corner of the environment. The agents are placed being close to the upper left exit gate in a way that they form a square shape as large as a quarter of the environment. An illustration of the simulation environment is given in Figure 2a.

The second test environment is a stage hall that has a rectangular stage leaning to the middle of one of the edges of the environment. The simulation environment also has three exit gates which is identical in terms of size. Two of these exit gates are positioned at the edges of the environment next to the stage. The last exit gate is located on the edge opposite of the stage. The agents are placed in front of the stage such a way that the agents are located equally distant from either of the two edges next to the stage. An illustration of the simulation environment is given in Figure 3a.

We employed two different test scenarios on the simulation environment described above. For the first scenario, we assumed that agents will choose the nearest gate which is a commonly used method in many evacuation systems. For the second one, the proposed method is used. We also employed two different metrics while evaluating the scenarios: The average evacuation time of the agents and the maximum evacuation time of the agents.

TABLE I: Average and maximum evacuation times of both scenarios (in seconds)

	Nearest Gate		Path Finding	
Agent #	Average	Maximum	Average	Maximum
500	122.75	254.57	67.40	121.92
1000	230.08	465.34	95.54	178.01

TABLE II: Average and maximum evacuation times of both scenarios (in seconds)

	Nearest Gate		Path Finding	
Agent #	Average	Maximum	Average	Maximum
500	73.23	133.74	68.40	115.68
1000	126.42	250.97	101.03	198.14

In Tables I and II, average and maximum evacuation times (in seconds) of the agents are presented for large hall and stage hall test environments respectively. Considering these results, it is obvious that the second scenario, which corresponds to the proposed evacuation system, finalizes the evacuation of virtual agents in the simulation environments quicker than the opponent scenario in terms of both the average and the maximum evacuation times.

As emphasized before, we constructed our test environments having non-uniform agent distributions among the exit gates. Because of this fact, during the simulation of the first test scenario in both test environments, the exit gates close to the agents' initial positions were very congested while there were almost no agents heading towards distant exit gates (see Figures 2b and 3b for each test environment respectively). On the other hand, in the second test scenario in both test environments (see Figures 2c and 3c respectively), the agents used the gates more intelligently and balanced the agent load among the exit gates roughly equal which leads to a decrease in

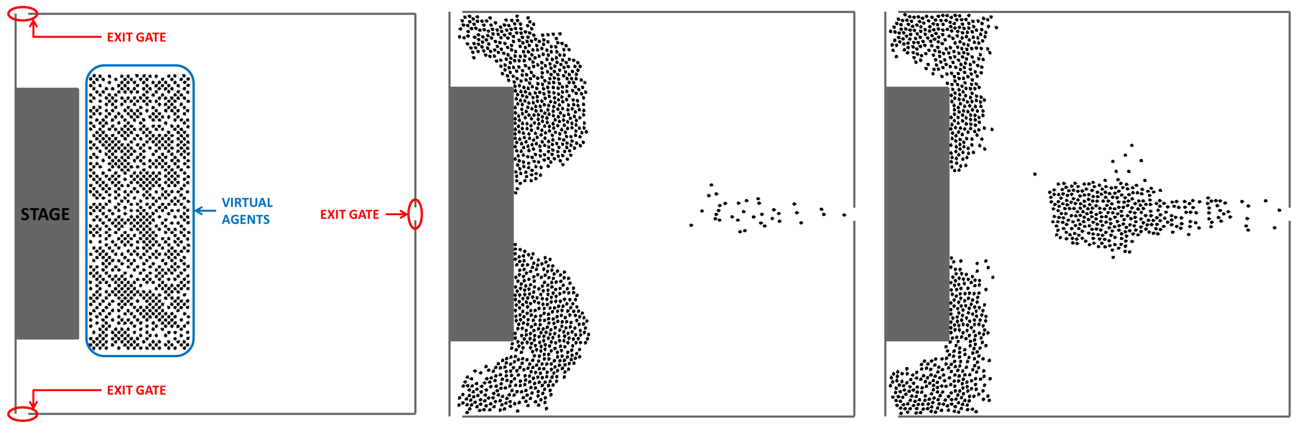

Fig. 3: (a) Top view of the initial simulation environment which has three exit gates and virtual agents standing right front of the stage. (b) Snapshot taken shortly after the beginning of the first scenario which agents choose nearest gates. (c) Snapshot taken shortly after the beginning of the second scenario that determines minimum cost exit gates via path finding.

both average evacuation time and maximum evacuation time. In short, the agents that use the proposed system look smarter (which is often desired in games) by using the exit gates more organized.

To assess the computational overhead which the path finding progress introduces, we measured average completion time of the simulation steps (excluding the rendering time)during both of the test scenarios containing 1000 virtual agents in stage hall simulation environment. According to the results obtained, the presented evacuation system requires 4.15% extra time per simulation step to perform 3 reverse path finding operations(one per exit gate) on a grid consist of 32×32 cells.

V. Conclusion

In this paper, we proposed a real-time evacuation simulation technique. The main goal of the proposed technique is intelligently planning the paths of the agents during an evacuation scenario with a minimal cost by taking advantage of the limited number of exit gates. The presented technique performs path finding from the exit gates to the grid cells and by reversing the found paths it not only obtains the paths from cells through the minimum cost exit gates but also assigns the minimum cost exit gates to these cells. While determining the minimum cost paths, we consider not only the distance but also the possible congestion of the exit gates. Agents are navigated towards the minimum cost exit gates thanks to the RVO which is a well-known and commonly used microscopic steering and collision avoidance algorithm.

The test results indicate that although we employed test environments which have non-uniform agent distributions, our model successfully optimizes the evacuation time. Using each gate effectively also increases the plausibility of the simulation (more intelligent-looking agents) as can be observed from the Figures 2 and 3 (and the accompanying video).

Another important advantage of our method is that computational overhead of minimum cost exit gate determination progress is minimized because of performing a path finding operation per limited number of exit gates once in every second. Hence our technique will achieve similar evacuation time results as nearest exit gate selection approach with a negligible computational overhead when the test environments have uniform agent distributions.

Finally, testing the recommended evacuation system with different number of exit gates and various grid sizes with the aim of estimating the scalability is not covered in this study and is left for the future work.

References

[1] E. Kuligowski and R. Peacock, "A review of building evacuation models," NIST Technical Note 1471, Tech. Rep., July 2005.

[2] Z. Daoliang, Y. Lizhong, and L. Jian, "Exit dynamics of occupant evacuation in an emergency," *Physica A: Statistical Mechanics and its Applications*, vol. 363, no. 2, pp. 501 – 511, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437105008447

[3] M. Fukamachi and T. Nagatani, "Sidle effect on pedestrian counter flow," *Physica A: Statistical Mechanics and its Applications*, vol. 377, no. 1, pp. 269 – 278, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437106012428

[4] A. Seyfried, B. Steffen, and T. Lippert, "Basics of modelling the pedestrian flow," *Physica A: Statistical Mechanics and its Applications*, vol. 368, no. 1, pp. 232 – 238, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037843710600118X

[5] R. M. Colombo and M. D. Rosini, "Pedestrian flows and non-classical shocks," *Mathematical Methods in the Applied Sciences*, vol. 28, no. 13, pp. 1553–1567, 2005. [Online]. Available: http://dx.doi.org/10.1002/mma.624

[6] X. Pan, C. Han, K. Dauber, and K. Law, "A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations," *AI & SOCIETY*, vol. 22, no. 2, pp. 113–132, 2007. [Online]. Available: http://dx.doi.org/10.1007/s00146-007-0126-1

[7] H. Ehtamo, S. Helivaara, T. Korhonen, and S. Hostikka, "Game theoretic best-response dynamics for evacuees' exit selection," *Advances in Complex Systems*, vol. 13, no. 01, pp. 113–134, 2010. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S021952591000244X

[8] R. Guo and H. Huang, "Logit-based exit choice model of evacuation in rooms with internal obstacles and multiple exits," *CHINESE PHYSICS B*, vol. 19, no. 3, MAR 2010.

[9] J. van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *14th International Symposium on Robotics Research*, Sep. 2009.