Pacific Graphics (2016)
E. Grinspun, B. Bickel, and Y. Dobashi (Editors)

Short Paper

Optimized Route for Crowd Evacuation

Sai-Keung Wong   Yu-Shuen Wang   Pao-Kun Tang   Tsung-Yu Tsai

National Chiao Tung University, Taiwan

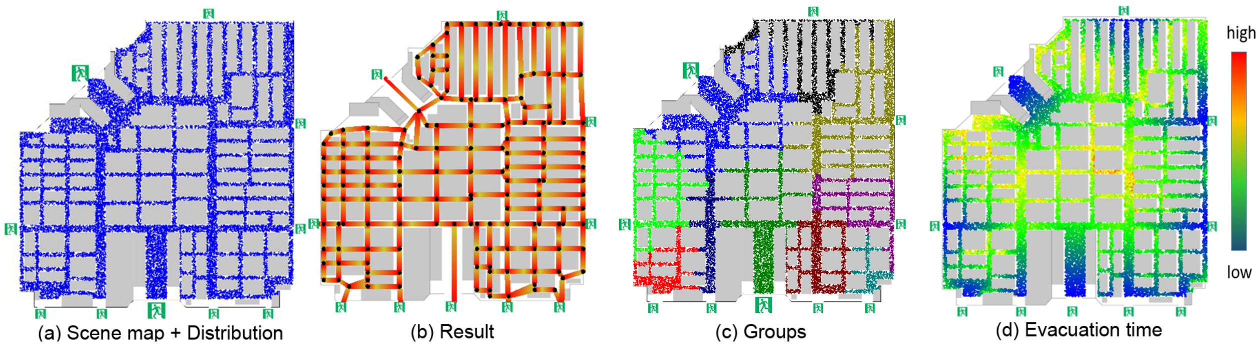

Figure 1: (a) Inputs of our system are a scene map and a distribution of evacuees (blue dots). The map is represented as a set of nodes and edges. Our system computes an optimal evacuation route of each local region such that evacuees can move away from the scene efficiently. The result is shown in (b), in which color transition from orange to red along each edge indicates the evacuation direction. (c) We also partition the evacuees into groups according to the exits they will arrive and distinguish the groups by colors. (d) The evacuation time of each evacuee is indicated by the transfer function.

Abstract

An evacuation plan helps people move away from an area or a building. To achieve a fast evacuation, we present an algorithm to compute the optimal route for each local region. The idea is to reduce congestion and to maximize the number of evacuees arriving at exits in every time span. Our system considers the crowd distribution, exit locations, and corridor widths when determining the optimal routes. It also simulates crowd movements during the route optimization. To implement this idea, we expect that neighboring crowds who take different evacuation routes should arrive at respective exits nearly at the same time. If this is not the case, our system updates the routes of the slower crowds. Given that crowd simulation is non-linear, the optimal route is computed in an iterative manner. The process repeats until an optimal state is achieved. Experiment results demonstrate the feasibility of our evacuation route optimization.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Crowd evacuation is important in building design and city planning. A well designed evacuation plan often contains an efficient evacuation guidance that can lead people to exits in a short period of time. The shortest path algorithm is the simplest way to crowd evacuation. However, the shortest distance is not equal to the shortest evacuation time because the method is oblivious to crowd distributions, corridor widths, and obstacles to exits. Evacuees often jam at particular places and wait when congestions occur. To prevent this problem, several mathematical models have been introduced to compute routes under some constraints [HT02]. These models represent an environment as a graph consisting of a set of nodes and a set of edges, which represent intersections and roads, respectively. Most of these models assume single movement direction of crowd along edges. They also require moving speed of crowd and road capacities when computing an evacuation route. However, crowd speed is time varying. It depends on the attributes of evacuees and degrees of crowd congestion. Without a physical simulation that computes the movement of each evacuee, it is difficult to understand whether an evacuation plan is effective. For example, an elderly person would move slower than a younger person; evacuees

DOI: 10.2312/pg.20161327

in a group would move together to exits; and crowd would have difficulty when moving from a branch route to a main route. These phenomena can only be observed from crowd simulation.

Our goal is to compute the optimal route for crowd evacuation. Given by the attributes and distribution of a crowd, exits, road widths, and obstacles, the presented system determines the evacuation direction of each road that can help people evacuate within the shortest time span. To achieve this aim, we represent the road network in a scene map by using a graph and initially set the evacuation direction of each edge based on the shortest paths algorithm. These directions are then iteratively updated according to the results of simulation. Note that the crowd on a road could evacuate in opposite directions. We call the place with this phenomenon a *division point*. The goal is to ensure that evacuees around a division point take similar time spans to arrive at respective exits. As the initial route is not optimal, our system updates the division points iteratively until all of them become stable.

To the best of our knowledge, the presented work is the first to combine crowd simulation and evacuation route optimization. Our system is able to handle time varying attributes of crowds and scene structures that are difficult in previous methods. In addition, our approach is adaptive to the results of crowd simulation. It can handle transportation such as buses, ferries and elevators. That is, a crowd of people would move to a specific position, wait, and only a portion of them can leave each time.

2. Related Work

Evacuation. Early evacuation methods are often based on mathematical models. They can be classified into categories such as static network, discrete/continuous time dynamic networks, and simple simulation (probabilistic and cellular automata). We refer readers to the survey [HT02, SKK*09] for more details.

Pelechano and Badler set leaders that are familiar with the scene at intersections and guide evacuees to safety regions [PB06]. Dimakis et al. employed a distributed simulator based on the multi-agent paradigm for managing emergency [DFG09]. Rodriguez and Amato [RA10] studied how the types of an interaction affect evacuation strategies. Tsai et al. [TFB*11] developed a multi-agent evacuation simulation to support several agent models such as children, parent, and authorities. Kretz et al. [KGH*11] proposed a method to compute the quickest path for each agent according to a local simulation. Given that the path is determined according to the information at the current state, congestion potentially occurs in the later stage. Agents could move back and forth during evacuation. In contrast, our system computes the evacuation route of each agent according to a full simulation so as to derive a stable result.

Crowd simulation and road map. Helbing and Molnár [HM95] introduced a social force model to simulate pedestrian dynamics. Van et al. [VdBLM08] presented a reciprocal velocity obstacle method to compute feasible movement directions of agents. Karamouzas et al. [KHvBO09] introduced the idea of predicting future collision so that the method can adjust the movement direction of each agent and achieve realistic. Guy et al. [GCC*10] computes a biomechanically energy-efficient and collision-free trajectory for crowd. Curtis et al. [CZGM13] set different priorities to agents and let agents with lower priorities to give way to those with higher ones. Johansson et al. [JHS07] applied the evolutionary optimization to compute parameters used in the social force model from real videos. Durupınar et al. [DGAB12] considered psychological parameters to form different crowd models. These kinds of models can be employed for enhancing the realism of crowd simulation. Van et al. [VTCG12] computed a density map based on the distribution of agents. Pettré [PLT05] computed the map from multi-layered and uneven terrains. Patil et al. [PVDBC*11] computed navigation fields to guide different crowds to their own destinations. Wong et al. [WTL*15] used particle swarm optimization to refine the predefined evacuation paths based on a fitness function.

3. Theoretical Background and System Overview

We adopt crowd simulation techniques to optimize evacuation routes. A crowd of people are represented by agents and their motions are simulated to obtain the time that demands for evacuation, i.e., the time taken by the agent to move from its initial position to an exit. We show an experiment in Figure 2, in which the road to the left exit is narrow, to explain why simulation is important in evacuation route optimization. The evacuation plan determined by the shortest path algorithm is oblivious to the distribution of agents. A good evacuation plan for 1000 uniformly distributed agents may not work well for 100 agents because the congestion problems are different. Accordingly, our goal is to refine the position of division point so that the evacuation time of the agents around it can be similar. To implement this idea, we initially set the evacuation route according to the shortest paths algorithm and then iteratively update the route until an optimal state is achieved. We also prevent head-on collision of agents when computing the route to prevent rapid increase of evacuation time. In other words, the following constraints should be satisfied:

1. Each edge has at most one division point to divide a crowd into two groups. In other words, agents in a group evacuate in one direction; and those in different groups separate from each other.
2. Guidance paths at an intersection cannot across. Otherwise, head-on collisions potentially occur and evacuees have to take long time to arrive at exits.

The input of our system is a scene map that indicates the positions of obstacles and roads. We transform the map into a graph $\mathbf{G} = (\mathbf{N}, \mathbf{E})$, where $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, ...\mathbf{n}_z,\}$ is the set of node positions, $\mathbf{n} \in \mathcal{R}^2$, z is the number of nodes, and $\{i, j\} \in \mathbf{E}$ indicates the node connectivity. We also denote by $\mathbf{N}_e \in \mathbf{N}$ the exits that demand agents to reach. Our goal is to compute the evacuation direction(s) of each edge and the division ratio of each node. When agents on an edge have to evacuate in opposite directions, we also determine the position of the division point to minimize evacuation time.

4. Evacuation Route Optimization

We initialize the evacuation route according to the distance to exits and then refine the route by agents' evacuation time. An optimal state is achieved when neighboring agents who take different routes arrive at respective exits simultaneously.

Route Initialization. We apply the shortest path algorithm to

Figure 2: We show an experiment to demonstrate that the optimal evacuation route depends on not only the scene structure but also the crowd distribution. The division point (the vertical red line) divides the crowd into two groups moving along the opposite directions. The division point is set at the same position in three different cases. In the experiment results, simulated agents (blue dots) evacuate to the left or right exit according to the guidance. As indicated, the division point should depend on the crowd distribution.

initialize the evacuation route. To implement this idea, we add an auxiliary node $\mathbf{m}$ that connects with all exits $\mathbf{n}_e \in \mathbf{N}_e$. The distance between $\mathbf{m}$ and $\mathbf{n}_e$ are set to zero. In addition, because the capacity of a road is inversely proportional to the road width, we set the cost of each edge $\{i,j\}$ to $\|\mathbf{n}_i - \mathbf{n}_j\|$ or $\frac{\|\mathbf{n}_i - \mathbf{n}_j\|}{w_{ij}}$, where w_{ij} is the minimal width of the road edge $\{i,j\}$. By applying the Dijkstra's algorithm to compute the shortest path tree, we obtain the initial evacuation route. We set the evacuation direction of each edge according to the shortest path tree. The directions guide the agents to evacuate in the simulation process. For edge $\{i,j\}$ that is in $\mathbf{E}$ but not on the shortest path tree, we set a division point on it because the paths from $\mathbf{n}_i$ to $\mathbf{m}$ and from $\mathbf{n}_j$ to $\mathbf{m}$ do not cross edge $\{i,j\}$. Ideally, the division point should be set based on the distances from nodes $\mathbf{n}_i$ and $\mathbf{n}_j$ to $\mathbf{m}$, respectively. But we simply set the point in the middle of $\mathbf{n}_i$ and $\mathbf{n}_j$ because the current route is only for initialization and will be updated according to the simulation result.

Route Optimization. We have assigned the initial evacuation direction based on the shortest path algorithm. A crowd can move along the evacuation direction to exits. But this approach does not consider the actual movement of agents and congestion problems. Therefore, we apply the crowd simulation method to estimate the evacuation time of each agent. The evacuation time of each local region is then determined as well. To implement this idea, we adopt an iterative method to 1) refine the division points on edges and 2) determine the division ratios at intersections to solve the mentioned problem. The refinement stops when the system converges or a prefined maximum number of generations is reached. By the preliminary tests, we set the prefined maximum number of generations to ten.

Refinement of division points. Let $\{i,j\}$ be the edge that has a division point q, our goal is to refine the position of q (denoted by $\mathbf{n}_q$) to ensure that crowds evacuate in opposite directions will arrive at respective exits simultaneously. To illustrate this idea, we further denote by T_i and T_j the evacuation time if an agent starts evacuation from node i and j, respectively, and ΔT_{qi} and ΔT_{qj} the time that agents have to take to move from $\mathbf{n}_q$ to $\mathbf{n}_i$ and $\mathbf{n}_j$, respectively (Figure 3). Considering that the two crowds should arrive at respective exits at the same time, we give the constraint

$$T_i + \Delta T_{qi} = T_j + \Delta T_{qj}. \quad (1)$$

If the constraint is not satisfied, we need to update the division point. Because crowds seldom congest when they start evacuation, ΔT_{qi} and ΔT_{qj} can be estimated for the next iteration by applying linearly proportion according to the distance from $\mathbf{n}_q$ to an end of $\{i,j\}$. Let the division ratio $\phi = \|\mathbf{n}_i - \mathbf{n}_q\| / \|\mathbf{n}_i - \mathbf{n}_j\|$. The new ϕ' should satisfy the constraint and thus we have:

$$T_i + \frac{\phi'}{\phi}\Delta T_{qi} = T_j + \frac{1-\phi'}{1-\phi}\Delta T_{qj}. \quad (2)$$

Here, $\frac{\phi'}{\phi}$ and $\frac{1-\phi'}{1-\phi}$ are the changes of the length ratios of the two edge segments of edge $\{i,j\}$, respectively. Let $\alpha = \frac{\Delta T_{qi}}{\phi}$ and $\beta = \frac{\Delta T_{qj}}{1-\phi}$, the division point q is updated by the new division ratio:

$$\phi' = \frac{T_j - T_i + \beta}{\alpha + \beta}. \quad (3)$$

Our system then refines the position of division point q and the evacuation routes. Note that ϕ' could be outside the interval [0,1]. It means that the division point should not be on edge $\{i,j\}$. If the valance of node i or j is two or higher, we propagate the division point to the adjacent edge(s) of $\{i,j\}$ if the evacuation time of the agents on the adjacent edge(s) is longer. Then we repeat the process until a stable state is achieved.

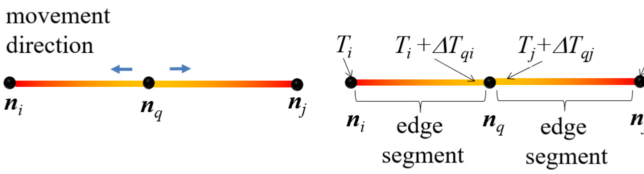

Figure 3: The travel times of agents along an edge with a division point. The division point divides the edge into two parts.

Crowd Simulation. We improve on the crowd simulation model [WTL*15] and simulate the crowd motions according to the guidance of the evacuation routes, obstacles, and crowd density of neighboring regions. Agents move along edges while avoiding collisions until they arrive at exits. As crowd simulation is not the major contribution, we ignore the implementation details in the rest of the paper. Other crowd simulation models are also possible. To obtain the evacuation direction of each agent, we find the edge $\{i,j\} \in \mathbf{G}$ that is closest to the agent. The temporal destination $\mathbf{d} = \mathbf{n}_\delta$ is then computed, where $\delta \in \{i,j\}$ is set based on the evacuation direction of edge $\{i,j\}$.

5. Experiment Results

We implemented the presented algorithm and executed the code on an Intel ®Core™ i5 CPU @ 3.20GHz with 8 GB RAM. To evaluate the performance of our method, we tested the presented system in various scenes, including indoor (Exhibition) and outdoor (Island) environments. There were 25000 and 20000 agents in Exhibition and Island, respectively. In the figures, agents and obstacles are represented as dots and gray polygons, respectively. We

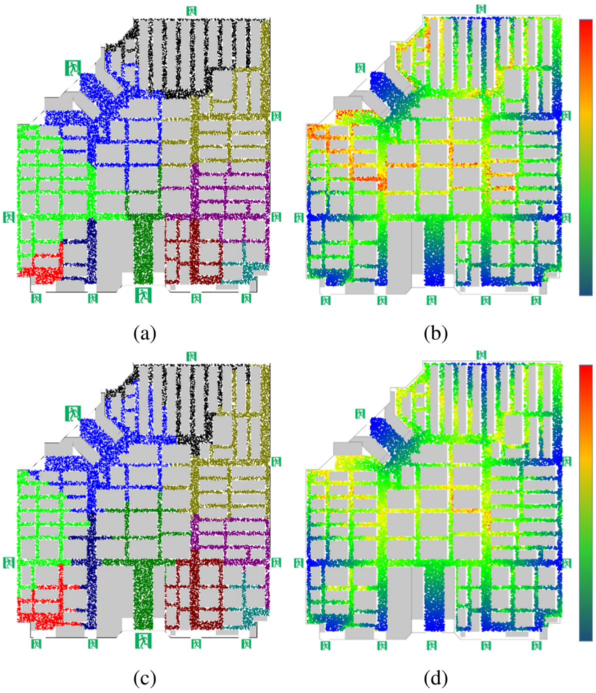

Figure 4: In this example, we compare the results determined by the shortest paths algorithm (top) and our method (bottom). (a) and (c) show the exit of each agent. (b) and (d) reveal the evacuation time of each agent. As indicated, agents who follow the guidance determined by our method can evacuate faster than those by the shortest paths method.

also applied the transfer function to colorize the dots/agents so as to represent their evacuation time. Overall, our system took eight generations on average to obtain the optimal evacuation route. The average training times (sec) per generation were 27.24 and 49.99 in Exhibition and Island, respectively.

Exhibition. Figure 4 shows the evacuation time of the agents that follow the routes determined by the shortest path algorithm and our method. The agents were evenly distributed in the scene. Given that the shortest path algorithm would guide agents to the nearest exits, the congestion problem potentially occurs. The problem often appears when the agents move from a branch to a main route because of collisions. In contrast, although our approach may ask a portion of the crowd to move to farther exits, it disperses the crowd to minimize the congestion and helps them evacuate efficiently.

Island. In this example, 20000 agents are unevenly distributed in the island due to the placement of facilities (Figure 5). Given that there are three exits at the right of the island, most agents can evacuate efficiently. We compare the evacuation time of agents that follow the guidance determined by the methods.

Island + transportation schedules. Considering that evacuation can be achieved by public transportations, we tested the presented system on an island, in which agents periodically leave the island whenever vacant ferries arrive at the pier. Specifically, we assume that the top left exit in Figure 6 to be a pier. The gate in

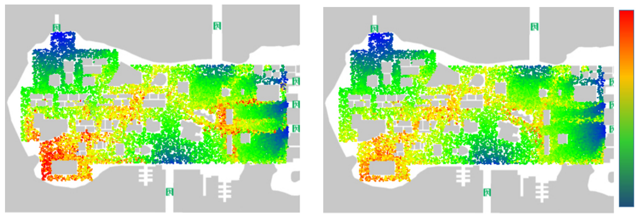

Figure 5: Island example. Evacuation time of each agent that follows the guidance determined by the shortest paths algorithm (left) and our method (right).

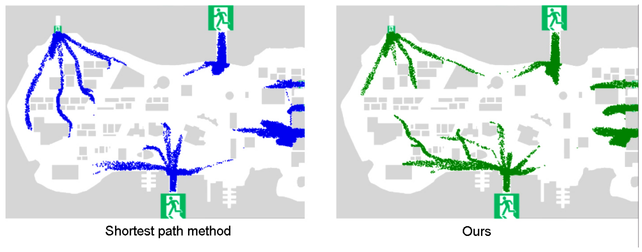

Figure 6: In this example, agents have to take ferries to evacuate from the top left corner. Since they have to wait for transportations, the evacuation from this direction could be slow. Our system thus guides more agents to another exits.

front of the pier opens every 1000 time steps and then closes after 200 agents get on the ferry. The distribution of agents are the same to that in the previous example. Since agents have to wait for ferries during evacuation, our method guides more agents to other exits and successfully minimizes the evacuation time.

Other crowd models. The presented system is flexible to crowd simulation models. To verify this issue, we integrated the social force [HM95] and reciprocal velocity obstacle (RVO) [VdBLM08] models into our framework. In general, compared with the shortest paths, the last agent simulated in these models saves 24% to 35% of time to evacuate if it is guided by our optimal route.

6. Discussions and Limitations

The inputs of our system are a scene map and the distribution of a crowd. The former is easy to obtain because the map simply indicates the positions of the obstacles. The latter could be a little bit difficult in some scenarios. Fortunately, people in urban cities usually have smart phones and their locations can be sensed by the global positioning system or wifi signals. There are also some surveillance cameras that can be used to estimate a crowd distribution [DG14]. When routes are determined, they can be conveyed by wireless devices [ISIK08] [Che12] and guide the users to evacuate. Dynamic exit signs can also be considered [DG14].

Our system relies on the results of crowd simulation. One of the strengths is its flexibility. However, as crowd behaviors are often difficult to simulate because of various psychological issues, no simulation models can be considered perfect. For example, people could become aggressive and cut into lines frequently if they do not feel good during evacuation. Our current system simply assumes that people are patient and willing to follow the guidance to evacuate. The simulation can work well in the situations such as the end of a concert or a happy new year party. But it is not able to simulate the crowd behaviors when a serious earthquake occurs. Accordingly, the evacuation route computed by our system is not guaranteed the optimum. Furthermore, as the entire simulation must be executed, the computation cost may be too high for a large scene. It is desirable to employ parallel computation techniques to reduce the computation cost. Finally, we will consider to apply data driven approaches for simulating believable crowd evacuations.

7. Conclusions and Future Work

We have presented a novel approach to combine crowd simulation and evacuation route optimization. Our system can handle crowds with various attributes and environments that are difficult to previous methods. Thanks to the integration of simulation, our approach is adaptable to elevators and public transportations that are commonly seen in evacuation. Although our system assumes that a crowd distribution in a scene is known in advance, the distribution is not difficult to estimate because of widely used smart phones and surveillance cameras. When the optimal evacuation route is computed, a simple and practical way is sending the route to each smart phone and guide the user to exits. In future, we plan to discuss this evacuation guidance with potential users, seek their feedback, and eventually develop the presented system to reach a practical level. Another issue is that the priority of evacuation in different zones are different. For example, people would prefer staying outside than inside a building because of many reasons such as fresh air and safety. Accordingly, we may compute the evacuation route in order to help people leave the first-priority zone efficiently. Finally, evacuation routes may be considered in traffic animation [LWLT16] and layout design [FYY*16].

Acknowledgement

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 102-2221-E-009-083-MY3, Grant MOST 103-2221-E-009-122-MY3, and Grant MOST 104-2221-E-009-051-MY3.

References

[Che12] CHEN C.-Y.: The design of smart building evacuation system. *International Journal of Control Theory and Applications 5*, 1 (2012), 73–80. 4

[CZGM13] CURTIS S., ZAFAR B., GUTUB A., MANOCHA D.: Right of way. *The Visual Computer 29*, 12 (2013), 1277–1292. 2

[DFG09] DIMAKIS N., FILIPPOUPOLITIS A., GELENBE E.: Distributed building evacuation simulator for smart emergency management. *The Computer Journal 53*, 9 (2009), 1384–1400. 2

[DG14] DESMET A., GELENBE E.: Capacity based evacuation with dynamic exit signs. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on* (2014), pp. 332–337. 4

[DGAB12] DURUPINAR F., GÜDÜKBAY U., AMAN A., BADLER N. I.: Psychological parameters for crowd simulation: From audiences to mobs. 1–9. 2

[FYY*16] FENG T., YU L.-F., YEUNG S.-K., YIN K., ZHOU K.: Crowd-driven mid-scale layout design. *ACM Trans. Graph. 35*, 4 (2016), 132:1–132:14. 5

[GCC*10] GUY S. J., CHHUGANI J., CURTIS S., DUBEY P., LIN M., MANOCHA D.: Pledestrians: A least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), pp. 119–128. 2

[HM95] HELBING D., MOLNÁR P.: Social force model for pedestrian dynamics. *Physical review E 51* (1995), 4282–4286. 2, 4

[HT02] HAMACHER H. W., TJANDRA S. A.: Mathematical Modelling of Evacuation Problems – A State of the Art. In *Pedestrian and Evacuation Dynamics* (2002), pp. 227–266. 1, 2

[ISIK08] INOUE Y., SASHIMA A., IKEDA T., KURUMATANI K.: Indoor emergency evacuation service on autonomous navigation system using mobile phone. In *Universal Communication, 2008. ISUC '08. Second International Symposium on* (2008), pp. 79–85. 4

[JHS07] JOHANSSON A., HELBING D., SHUKLA P. K.: Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in complex systems 10*, supp02 (2007), 271–288. 2

[KGH*11] KRETZ T., GROSSE A., HENGST S., KAUTZSCH L., POHLMANN A., VORTISCH P.: Quickest paths in simulations of pedestrians. *Advances in Complex Systems 14*, 5 (2011), 733–759. 2

[KHvBO09] KARAMOUZAS I., HEIL P., VAN BEEK P., OVERMARS M. H.: A predictive collision avoidance model for pedestrian simulation. In *Motion in Games*, Lecture Notes in Computer Science. Springer, 2009, pp. 41–52. 2

[LWLT16] LIN W. C., WONG S. K., LI C. H., TSENG R.: Generating believable mixed-traffic animation. *IEEE Transactions on Intelligent Transportation Systems PP*, 99 (2016), 1–13. 5

[PB06] PELECHANO N., BADLER N.: Modeling crowd and trained leader behavior during building evacuation. *Computer Graphics and Applications, IEEE 26*, 6 (2006), 80–86. 2

[PLT05] PETTRE J., LAUMOND J.-P., THALMANN D.: A navigation graph for real-time crowd animation on multilayered and uneven terrain. vol. 43, p. 194. 2

[PVDBC*11] PATIL S., VAN DEN BERG J., CURTIS S., LIN M. C., MANOCHA D.: Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (2011), 244–254. 2

[RA10] RODRIGUEZ S., AMATO N.: Behavior-based evacuation planning. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (2010), pp. 350–355. 2

[SKK*09] SCHADSCHNEIDER A., KLINGSCH W., KLUPFEL H., KRETZ T., ROGSCH C., SEYFRIED A.: Evacuation dynamics: Empirical results, modeling and applications. In *Encyclopedia of Complexity and Systems Science*, Meyers R. A., (Ed.). Springer New York, 2009, pp. 3142–3176. 2

[TFB*11] TSAI J., FRIDMAN N., BOWRING E., BROWN M., EPSTEIN S., KAMINKA G., MARSELLA S., OGDEN A., RIKA I., SHEEL A., ET AL.: Escapes: evacuation simulation with children, authorities, parents, emotions, and social comparison. In *Int'l Conf. on Autonomous Agents and Multiagent Systems-Volume 2* (2011), pp. 457–464. 2

[VdBLM08] VAN DEN BERG J., LIN M., MANOCHA D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE Int'l Conf. on Robotics and Automation* (2008), pp. 1928–1935. 2, 4

[VTCG12] VAN TOLL W. G., COOK A. F., GERAERTS R.: Real-time density-based crowd simulation. *Computer Animation and Virtual Worlds 23*, 1 (2012), 59–69. 2

[WTL*15] WONG S. K., TANG P. K., LI F. S., WANG Z. M., YU S. T.: Guidance path scheduling using particle swarm optimization in crowd simulation. *Computer Animation and Virtual Worlds 26*, 3-4 (2015), 387–395. 2, 3