Robot Assisted Evacuation Simulation

Ibraheem Sakour and Huosheng Hu
School of Computer Science & Electronic Engineering, University of Essex
Email: isakou@essex.ac.uk, hhu@essex.ac.uk

***Abstract*—It is extremely challenging to safely evacuate a crowd from massive events and buildings such as concerts and football fields. Since the panic caused by the emergency evacuation has a higher risk to human lives, careful planning and suitable measures have to take place to ensure the safety of the citizens. This paper addresses the evacuation problem by suggesting a system utilizing robotics technologies to help the evacuation process. The proposed evacuation system provides a crowd behavior simulation using an agent-based model. The simulated evacuation will be observed by an unmanned aerial vehicle which communicates with the individuals and redirects them towards less crowded exits and away from the the danger source. Finally, the proposed system performance is evaluated and the benefits of using robot aid in evacuations are presented.**

I. Introduction

The problem of safe crowd evacuation at mass gathering events and crowded office buildings has attracted the attention of researchers. In crowded spaces, panic can be a main reason of the loss in lives [1], bad designs in buildings and staged areas can also be a reason of making the situation even more dangerous. Over the years many related accidents have been reported. One of the recent incidents was the Phnom Penh stampede in Cambodia in 2010 which was caused by the lack of crowd control, where people got stuck on a bridge and people on both sides were pushing in opposite direction. This incident resulted in hundreds of casualties and injuries [2]. Mina stampede in 2015 was another disaster where the lack of control resulted in hundreds of casualties [3].

Emergency evacuations usually have similar problems such as the physical interactions between individuals and the bottleneck problem at common exits. Bottleneck effect can happen when people run towards an exit at the same time, in high density crowd situations human arch forms and causes the evacuation process to slow down and injuries can occur [4]. To simulate the human behavior in evacuations, several methods can be used, including Cellular Automata (CA), Force-Based model, Social-Based model and Agent-Based model.

Cellular Automata (CA) model was first introduced by Wolfram for crowd simulation in 1983 [5]. This model has a simple representation, and can be inaccurate in some environments. The Force-Based model describes the environment as a set of forces that will affect the individuals in their movement, and mathematical methods govern these types of simulations [6]. Force Based models can give precise results of general movements of individuals and provide a continuous time description of a moving crowd. However, decision making and communications between individuals are very hard to be modelled using mathematical equations. Based on that, Helbing et al. suggested the Social-Based model in 1995 which is a form of the force based model [6].

The Agent-Based models are used to add more human like details to crowd simulations, in which each agent in a crowd can be fully autonomous [7]. This model also made it easier for individuals to communicate with others during the simulation. Furthermore, some studies have suggested combining different models together, e.g. combining the Agent-Based model with the Force-Based model [8]. In particular, the Agent-Based model was used for the top level of the design and responsible for the exchanges between individuals and general movements, and the Force-Based model can be used for local movements at the low level of the model.

To save the lives of people during emergency situations, sensors could be installed in crowded environments to detect potential accidents and indicate safe passageways for evacuation. However, the sensors in these structured environments may be damaged during the accidents. This paper aims to use robots to guide and redirect evacuees from overcrowded and dangerous exits, to avoid the dangerous areas and take the safest path possible. The goal is to achieve less bottleneck situations and faster evacuation times. Robotic help can also provide dynamic instructions to the crowd during the evacuation process.

The proposed system consists of an agent-based evacuation model for the crowd and the robot model for an unmanned aerial vehicle (UAV). The evacuation simulation environment is created using the Unity3D game engine, where it combines the NVidia PhysX engine and a custom made rendering engine with Mono [9]. The PhysX engine is able to simulate complex 3D movements and particle systems to simulate fire, water and gas. It is very reliable, efficient and suitable for large scale simulations. The custom rendering engine is able to simulate complex environment provided with the user friendly GUI. Mono is an open source development platform based on the .NET framework in order to use C# to write scripts, it is used like a programming platform to control the different part of simulation system.

In the crowd model that combines the Force-Based model and the Agent-Based model, the movement of each individual is determined by forces generated from the corresponding behaviours. The Force-Based model is at the lower level, which is responsible of translating the behaviours to movements through set of rules and equations. On the other hand, an agent-based model is at the higher level to represent the decision making process, and responsible for behavior set selection. The combined effects will be calculated at the lower level. The functionality of the simulated UAV depends on the physics engine. Different sensors can be simulated using the abilities of the game engine [10] (Ray casting, Apply forces), where the ray casting is used to simulate the different sensors for the altitude, surroundings detection and other sensors. Moreover, the ability to apply forces on the simulated UAV body makes it possible to simulate the flight process.

The rest of the paper is organized as follows. The second section, related literature and previous work is reviewed. In section III, the crowd model design is explained providing details for the system responsible for the evacuation process. After that, the environmental design will be discussed in section IV. Different scenarios for the simulation are explained in the next section V, while in section VI the experimental results for the proposed system are described. Finally, in section VII, conclusions about the presented work are drawn.

II. Related Work

A. Crowd behaviour modelling

To date, different models have been proposed to describe a crowd behaviour, which can be divided into two categories: macroscopic models and microscopic models. Macroscopic models focus on the general movement of the crowd, most recognized model for this category is the Navier-Stokes fluid dynamics model [11]. However, macroscopic models do not provide details of the individuals. On the other hand, microscopic models focus on the details of individuals. Many studies on crowd simulation are carried out using the microscopic models to understand the crowd behaviour in emergency events so that safer environments and better techniques can be deployed to control crowds in such situations. Typical microscopic models include cellular automata models, force-based models and agent-based models.

The Agent-based models or (ABMs), treat individuals as distinct entities, each one has attributes differ from others. Agent-based models formed the main basis for the crowd simulation during the last decade. Regardless of the ABMs computational costs, they offer the ability to describe each individual with multiple unique attributes, what make them suitable to model distinct human behaviours [12]. In [13], Crociani et al. introduced a new system to simulate individuals and crowds, this system considers individuals as utility-based agents. In [14] Yeh et al. proposed a system that takes the social and psychological factors into consideration by associating the basic agent with one or more proxy agents. Furthermore, Agent-based models can be used for more realistic evacuation simulations where special attributes are required such as panic, disorientation and even injuries [15].

B. Robots In evacuations and public safety

In previous evacuation robot research, robots with directional audio beacons [16] were deployed in optimal positions to reach as many people as possible. These robots were shown to decrease the total amount of time to evacuate in a simulation of an emergency. Physical robots were also deployed in a building to show that the system can automatically redeploy due to the loss of a robot. Others have developed simulators to test different algorithms for robots in emergency scenarios [17]. Zhang and Guo used potential fields to model the physical environment during a fire emergency and simulated evacuees in a manner [18]. Boukas et al. developed a new crowd modelling technique using cellular automata and tested it by having a robot provide guidance to human participants in a simulated evacuation [17].

Participant trust in the robot was not evaluated and the authors did not report motivating participants to find an exit quickly. Instead, the authors used this experiment to validate their prior simulations on crowd movement. Additionally, other researchers have created a 3D simulator for evaluating human-robot trust in emergency scenarios. Researchers in [19] performed work in a social interface for human-machine trust, trust in human-robot teamwork, and using simulated emotions to describe trust. In [20], the authors presented robots equipped with sensors that can detect some chemical presence, and have the ability to assist in crisis management. Also the impact of the trust relationship between the robots and human agents is simulated in [21].

III. Crowd Model design

The crowd will be represented as a group of agents, different attributes will be assigned to each single agent which will make them act like different individuals. In other words these attributes will define the personality of each agent. Having different reactions to the surrounding environment gives the crowd the microscopic aspect we discussed in the literature review. The behaviours available are shared between all individuals, but the data received from the surrounding environment through their preceptors, and the different attributes they have will determine the behaviour they are going to perform.

The crowd model works on two levels, the first level is the simple control level which determine the general direction of the agent and general movements towards the first logical goal. The second layer is the main controller of the agent which is responsible for all the actions the agent takes in order to move around and avoid obstacles and danger zones.

In this chapter we will discuss the agent design and the action plan that every agent follows during its life cycle. The general structure of the crowd model can be shown in Figure 1 where the behaviour library consists of a collection of general purpose behaviours. Each one of these behaviours explains the effects on the agent through unified formula using the unique attributes, status and perceptions of the related agent. The main behaviours included in the behaviour library are: 1) move to target: static target and here it means to move to specific exit, 2) Avoid target, 3) Follow: follow another agent or robot.

Next, agent action plan, or the brain of the agent, where it is responsible for every action and every single step of the simulation time. Action plan will retrieve information from the updated map, surrounding environment and the available behaviours, all that in order to update the agent status. Then the simulated world will be updated according to the latest activity of the agent. Objects in the simulation world may be affected too. This process will be repeated during the life of

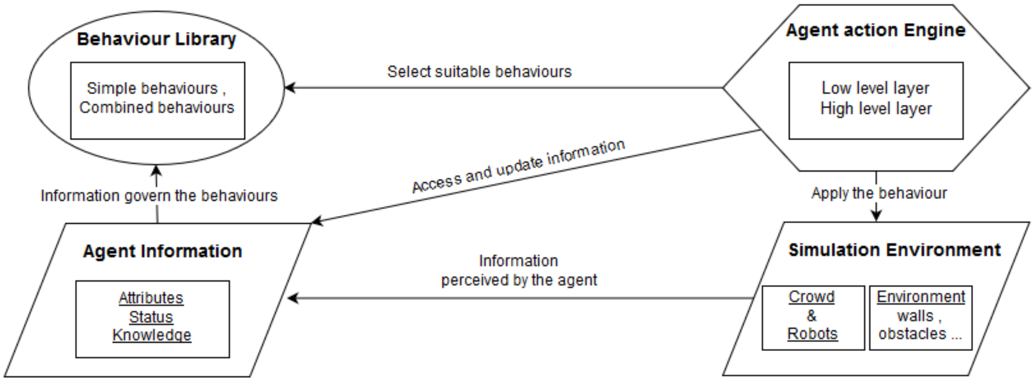


Fig. 1: Crowd behaviour analysis

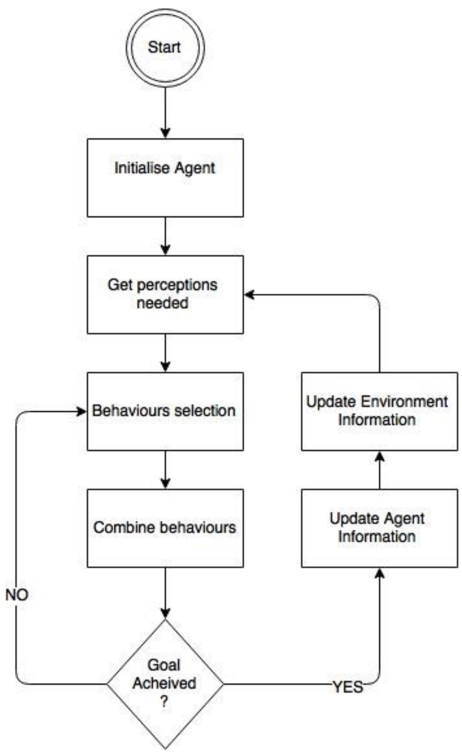


Fig. 2: Agent action process

the agent every single step in order to decide the next action. Action engine can be described in Figure 2.

Agent information consists of attributes, status and knowledge. Attributes determine the character and personality of an agent. The attributes of the agent contains information necessary to make the next decision. Attributes include the physical specifications that describe the size, walking and running speed and location. Other attributes are the range and personality attributes. Which is responsible for defining the sight range and preferred distance from others and obstacles. Personality attributes give the agent the level of experience that can influence others during an evacuation and the preference level for joining a group during the panic. Agent status describes the set of attributes through the simulation life cycle, these attributes can be static or dynamic changing with every simulation step Δt . The knowledge possessed by the agent means the map awareness and the data the agent gets from the surrounding environment, in this system the map can be fully known or partially, depending on the level of experience. The pseudo code for the agent process is presented as follows:

```
Initialise behaviour variables for calculation;
Retrieve the agent status in the previous loop;
Retrieve the environmental information;
Decide the active behaviour;
Pick the possible side behaviours;
Calculate the next step from current behaviours;
if next step can cause a collision then
    Cancel the calculated step;
    Pick the collision avoidance behaviour;
    Calculate the next step;
end
Update the agent status and relative data;
End;
```

Algorithm 1: Agent process algorithm

IV. Environment Design

In order to model the environment, Blender has been used to build the 3D walls and it was imported to the Unity editor as a mesh, a mesh collider was applied to the walls in order to represent it as an obstacle in the environment. In figure 3 we show the 3d representation of the environment and the objects. Colliders are used to detect the collisions between objects every collision can be used as a trigger to invoke a specific method. For the hazard we used the particle system available by the editor. The fire particle system was given a collider to make it detectable by the agents if they came at a close range of the fire, and for the UAVs when they are used to investigate the scene looking for hazards.

Quad-rotor model was created using the physical engine provided by Unity, the mathematical model was used as a reference to create the engines. In figure 3 we can see the sensor module activated to measure the height. In fact the sensor module is using RayCast() [10] command which launches a ray beam in a specific direction and returns different

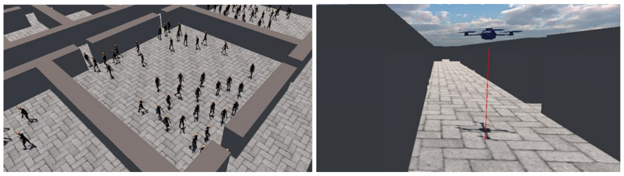


Fig. 3: Simulation Environment

information about the object the ray collided with such as the distance between the ray caster and the object that ray reflected off. Cameras can also be implemented with ray casts. Inertial measurement unit (IMU) can be implemented using the translation and rotation for the object simulating the IMU. Next we mention the main sensors implemented in the unity environment:

- LRF sensor: Ray-cast function can be used in its original intended use in Unity, where Ray-cast function takes different parameters, the maximum distance for the laser, scan angle, and the frequency.
- Compass: the compass implementation is very simple using the references.
- IMU: the inertial measurement unit calculates inertial changes for every frame by comparing the current position and orientation with the previous state.
- Camera: Like the laser scanner the simulated range camera uses of Unitys Ray cast function. By default, this produces a 64x64 array of range data.

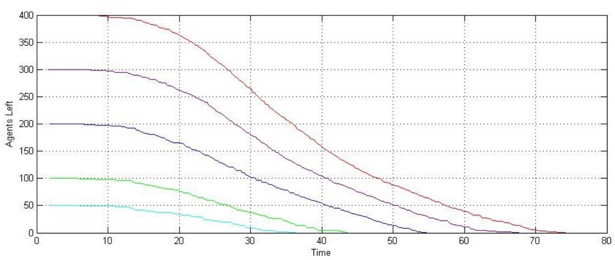


Fig. 4: Evacuation in normal situation

V. Evacuation Scenarios

First of all the evacuation will take place without the help of the robots. We will record the performance in an unassisted evacuation in order to compare the performance with an assisted evacuation. Different specifications will be applied on the crowd where, the crowd size and level of experience for agents will vary for number of experiments. the main purpose of these experiments is to show the relation between evacuation times and affecting factors. Also panic will be applied to see the change in evacuation times. The panic model is simple where the walking and running speed will be increased and the inexperienced individuals will tend to follow others and form larger groups.

In the second scenario the UAV will try to locate the source of danger and then move around the area giving warnings to people about the danger nearby and try to guide them to the safe exit. The guidance signals in our system is simply a message transmitted from the robot to the detected agent, which is responsible for updating the knowledge base for the agent. Pseudo code for the UAV process is shown as follows:

Initialize the all the movement and behaviour variables;
Get UAV status in previous simulation step;
Get environmental information;
if *Hazard Not Detected* **then**
  Get a new area location to set goal;
  Decide the Suitable behaviour;
  Calculate the next step;
  if *Next step will cause collision* **then**
    Change Behaviour to collision avoidance;
  end
  Update the UAV status and relative data;
end
Set behaviour to hover near;
if *Agent Detected* **then**
  Send warning signal to detected agent;
end
End;

Algorithm 2: UAV Search process

VI. Experimental results

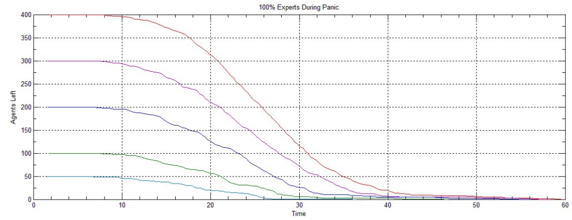
In this section we will present the results we achieved from running the simulation in different situations. First of all will show the normal evacuation times, with no panic or any danger source, agents will be evacuating in normal speed without any alerts, then the agents will be given a panic factor to see how panic can affect the evacuation scenario. Then the UAV will be used in the experiment to assist the evacuation, where the UAV will be roaming the environment looking for the hazard source. After identifying the source of danger the UAV will give warnings to the close agents in order for them to choose an alternative exit.

A. *Evacuation in normal situation*

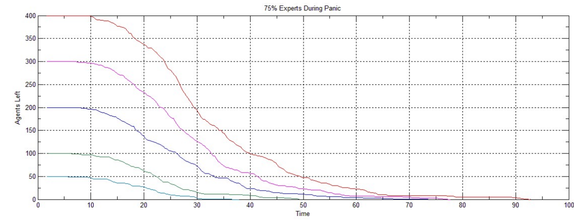
In this experiment the agents will start evacuation at the start of the simulation time, there is no panic model applied, agents are moving in normal speed. Also the agents are all experts which means they are familiar with the environment and they know all the routes that lead to emergency exits, the reason of this experiment is to set a normal case, which we can relate to when we analyse the other situation. In figure 4, all the agents are considered to be familiar with the environment. We notice that the evacuation time is related to the number of agents being evacuated, the individuals move during this scenario in a normal walking speed. The evacuation time consists of the time needed to walk each agent from the initial position to the closest exit. in addition to that we have some delays where large number of agents will gather around one of the exits to evacuate.

B. *Evacuation in panic mode*

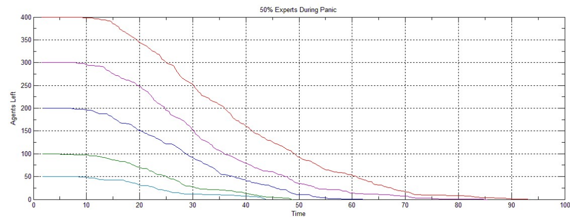
In this experiment we implemented a panic model where it depends on the level of experience the crowd hold, experienced



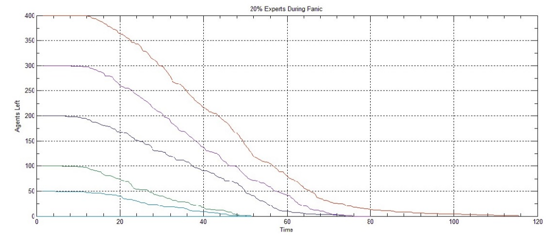
(a) All agents are experts



(b) 75% of agents count are experts

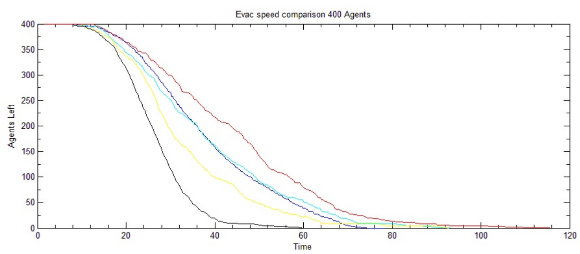


(c) 50% of agents count are experts

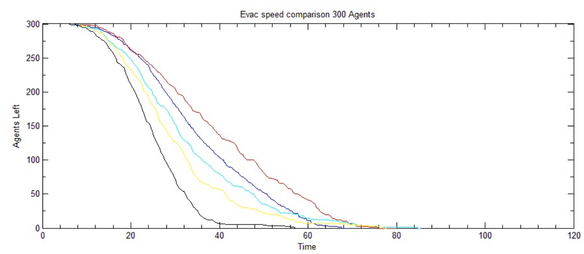


(d) 20% of agents count are experts

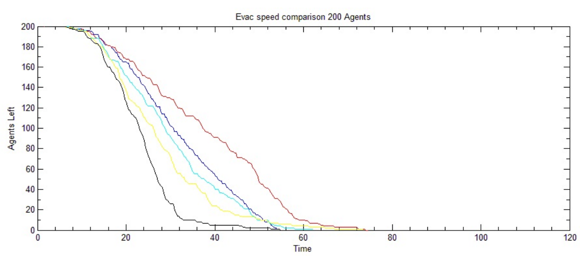
Fig. 5: Evacuation times for different types of crowd



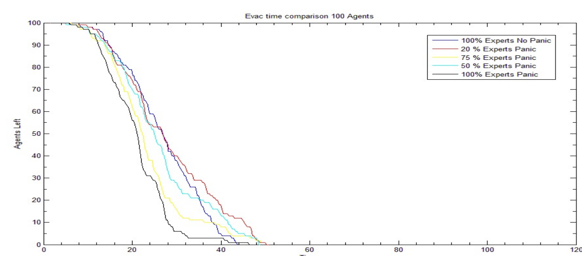
(a) 400 Agents



(b) 300 Agents



(c) 200 Agents



(d) 100 Agents

Fig. 6: Comparison between evacuation times for different types of crowd

agents will know which exit is the closest to their position, and inexperienced agents will randomly choose an exit to represent visitors to the environment. These agents with little knowledge about the environment will pick a random exit, or simply follow the crowd in order to reach an exit. after applying the simulation multiple times we draw the results in fig.5. In figure 5a we show the average evacuation times for different sized crowds in panic situation, but all the agents are experienced. In comparison between normal mode and panic mode we notice the time has increased from the normal mode to the panic mode situation. This delay is happening due to the panic factor that is affecting the movement speed where the agents started to affect each other while running to the exits. At the intersections multiple agents will start blocking one another, in panic situation the agents range attributes are affected. The preferred distance from other agents and obstacles is decreased where they are allowed to get closer and as a result they start pushing each other affecting the evacuation time, causing it to increase.

We repeated the process for different settings. In figures 5b, 5c and 5d the number of expert agents was lowered to be at 75%, 50% and 20% of the overall agent count, where we notice the time increased whenever we lower the percentage of overall expert agents. That represents agents with limited knowledge about the environment where most of them will take random directions and join other groups.

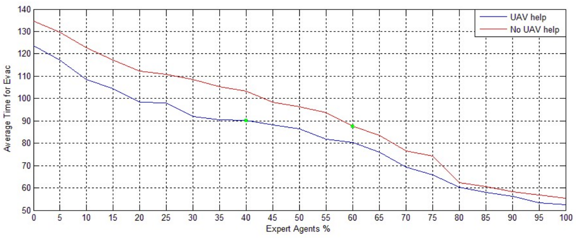


Fig. 7: Time comparison between the different scenarios in relation to the number of experienced agents

As a result, bottlenecks will occur and more delays will take place. In figures 6a, 6b, 6c and 6d we see comparisons between evacuation times in different scenarios where the number of agents and panic mode are taken into consideration each graph is a comparison between evacuation times for the same number of agents. Each graph is showing the different evacuation times for a crowd of the same size. All four graphs have similar characteristics, for 100% Expert crowd with panic evacuation, the time is noticeably lower than the normal situation time. That can be related to the increase in speed in panic state. The second characteristic is that the evacuation time is Inversely proportional to the quality of the crowd in terms of the map awareness and panic state.

Following the steps in the UAV flight scenario the search will start from the spawn point to locate the hazard source. After that it will fly in the area close to the source and work on detecting individuals in order to send warnings to change their routes and lead them to a safer exit. In figure 7 we have a comparison between evacuation times where the first case without the UAV help. Where multiple simulation runs were performed for each crowd percentage. Then, the second case performed with the help of the UAV. We notice that the average time to complete the evacuation has decreased in average value of 10 seconds. If we take the time of 90 seconds as an acceptable evacuation time we notice that the average time for a safe evacuation will take place for a crowd with 60% experienced individuals without the help of a robot. On the other hand, when we use the robot to help with the evacuation we notice that we can achieve similar results for 40% experienced agents of the crowd.

VII. Conclusion

This work is motivated by the necessity to introduce a reliable robot assisted crowd evacuation system. The proposed system relies on the high level agent-based models to describe crowd interactions and movements, and is used to simulate a crowd evacuation in normal and panic situations. An evacuation method using UAV was proposed to assist the crowd and finish the evacuation task. The experimental results show the difference between different scenarios, and the benefits of using the robot assisted evacuation. The time needed to evacuate the building was enhanced, where the inexperienced crowds can perform a fast evacuation. This can be beneficial in general facilities where most of the visitors are new to the facility.

Our future research will be focused on the investigation of how the proposed robot aid system could be effectively deployed in different evacuation scenarios, and how the crowd model could be used to emulate a real life evacuation with a high degree of reliability.

References

[1] M. Liu and S. M. Lo, "The quantitative investigation on people's pre-evacuation behavior under fire," *Automation in construction*, vol. 20, no. 5, pp. 620–628, 2011.

[2] X. Wei, W. Lv, W. Song, and D. Wang, "Evacuation analysis of a hospital based on fds+evac software," in *Remote Sensing, Environment and Transportation Engineering (RSETE), 2011 International Conference on*, June 2011, pp. 3147–3150.

[3] Mecca: hajj crush kills hundreds near holy city as it happened. [Online]. Available: https://www.theguardian.com/world/live/2015/sep/24/hajj-crush-kills-scores-near-mecca-live-coverage

[4] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, p. 2000.

[5] O. Martin, A. Odlyzko, and S. Wolfram, "Algebraic properties of cellular automata," *Mathematical Physics*, 1984.

[6] W. Daamen, D. Duives, S. P. Hoogendoorn, S. Seer, C. Rudloff, T. Matyus, and N. Brndle, "Pedestrian and evacuation dynamics 2014 (ped 2014)," *Transportation Research*, vol. 2, pp. 724 – 732, 2014.

[7] X. Pan, C. S. Han, K. Dauber, and K. H. 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.

[8] F. Qiu and X. Hu, "Modeling group structures in pedestrian crowd simulation," *Simulation Modelling Practice and Theory*, vol. 18, no. 2, pp. 190 – 205, 2010.

[9] R. TETI, G. Michalos, P. Karagiannis, S. Makris, and G. Chryssolouris, "Research and innovation in manufacturing: Key enabling technologies for the factories of the future," *Procedia CIRP*, vol. 41, pp. 370 – 375, 2016.

[10] (2016, 10, June) Unity3d documentation: Unity scripting reference. [Online]. Available: https://docs.unity3d.com/ScriptReference/

[11] A. Palmer and R. Bailey, "Sex differences and the statistics of crowd fluids," *Behavioral Science*, vol. 20, no. 4, pp. 223–227, 1975.

[12] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[13] L. Crociani, L. Manenti, and G. Vizzari, *MAKKSim: MAS-Based Crowd Simulations for Designer's Decision Support*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 25–36.

[14] H. Yeh, S. Curtis, S. Patil, J. van den Berg, D. Manocha, and M. Lin, "Composite agents," in *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '08. Switzerland: Eurographics Association, 2008, pp. 39–47.

[15] X. Zheng, T. Zhong, and M. Liu, "Modeling crowd evacuation of a building based on seven methodological approaches," *Building and Environment*, vol. 44, no. 3, pp. 437 – 445, 2009.

[16] D. A. Shell and M. J. Mataric, "Directional audio beacon deployment: an assistive multi-robot application," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 3, April 2004, pp. 2588–2594 Vol.3.

[17] E. Boukas, I. Kostavelis, A. Gasteratos, and G. C. Sirakoulis, "Robot guided crowd evacuation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 739–751, April 2015.

[18] S. Zhang and Y. Guo, "Distributed multi-robot evacuation incorporating human behavior," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, June 2013, pp. 864–869.

[19] D. Atkinson and M. Clark, "Autonomous agents and human interpersonal trust: Can we engineer a human-machine social interface for trust?" 2013.

[20] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*, 2nd ed. Hoboken, N.J. ; [Great Britain] : Wiley, 2003.

[21] P. Robinette, P. A. Vela, and A. M. Howard, "Information propagation applied to robot-assisted evacuation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 856–861.