

Article

# Robot-Assisted Crowd Evacuation under Emergency Situations: A Survey

Ibraheem Sakour \* and Huosheng Hu

School of Computer and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK; hhu@essex.ac.uk

\* Correspondence: isakour@essex.ac.uk; Tel.: +44-1206-872-477

Academic Editor: Genci Capi

Received: 27 December 2016; Accepted: 31 March 2017; Published: 7 April 2017

**Abstract:** In the case of emergency situations, robotic systems can play a key role and save human lives in recovery and evacuation operations. To realize such a potential, we have to address many scientific and technical challenges encountered during robotic search and rescue missions. This paper reviews current state-of-the-art robotic technologies that have been deployed in the simulation of crowd evacuation, including both macroscopic and microscopic models used in simulating a crowd. Existing work on crowd simulation is analyzed and the robots used in crowd evacuation are introduced. Finally, the paper demonstrates how autonomous robots could be effectively deployed in disaster evacuation, as well as search and rescue missions.

**Keywords:** crowd evacuation; emergent events; crowd simulation; robot; search and rescue missions

## 1. Introduction

Since the panic in crowded environments can cause loss of life [1], the study of effectively dealing with emergency events becomes necessary and a large number of experiments are required. As real-life experiments can be very costly and dangerous, simulated experiments become necessary. Crowd simulation was firstly introduced in [2] to investigate the behaviors and movements of a crowd in different situations. After that, a number of tools were created to deploy robots to help the crowd in a panic situation or to prevent a panic from happening.

In general, the models describing a crowd behavior can be divided into two categories, namely macroscopic models and microscopic models. Macroscopic models describe general crowd movement, and do not provide details of the individuals. Fluid dynamics is the most recognized macroscopic model, namely Navier-stokes [3]. In contrast, microscopic models are focused on the details of individuals. They have been used for many studies on crowd simulation in order to understand crowd behavior in emergency events. Cellular automata models, force-based models and agent-based models are typical microscopic models. The cellular automata (CA) model was first introduced by Wolfram for crowd simulation in 1983 [4]. 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 movements, and mathematical methods govern these types of simulations. Force-based models can give precise results of general movements of individuals and provide a continuous time description of a crowd. However, decision making and communications between individuals are very hard to model using mathematical equations.

As we know, the social-based model suggested by Helbing et al. in 1995 is a force-based model [5]. After that, agent-based models were used to add more human-like details to crowd simulations, in which each agent in a crowd can be fully autonomous [6]. This model also made it easier for individuals to communicate with others during the simulation. Furthermore, some studies have suggested combining different models together like the agent-based model with the force-based

model [7]. More specifically, the agent-based model was used for the top level of the model and was 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.

When an evacuation is required in a crowded place, robotic systems can help people to reach safety in an well-organized way. For instance, robots were used as beacons to help them to reach a safe exit by sending audio messages to the individuals [8]. In this implementation, a multi-robot system and a flow-based approach were used. Autonomous robots have been deployed into a number of evacuation missions, such as the hazard environment mapping and human detection [9], and some similar approaches are presented by authors of [10–12].

The rest of the paper is organized as follows. In Section 2, we discuss how human behaviors in crowd situations could be modeled in terms of the individual, interaction and the group. Then, a few crowd simulation methods are reviewed in Section 3, including the macroscopic approach, microscopic approach and their combination. Section 4 outlines some existing work on crowd simulations proposed by many researchers. Section 5 describes the simulated robotic platforms that have been used in crowded evacuation experiments. Finally, a brief conclusion is given in Section 6.

## 2. Crowd Behavior Analysis

It is necessary to understand the concepts that can be taken into consideration for modeling a crowd. In general, the description of crowd behavior can be divided into three types [13], namely (1) the individual; (2) the interactions between the individuals and (3) the group, as shown in Figure 1.

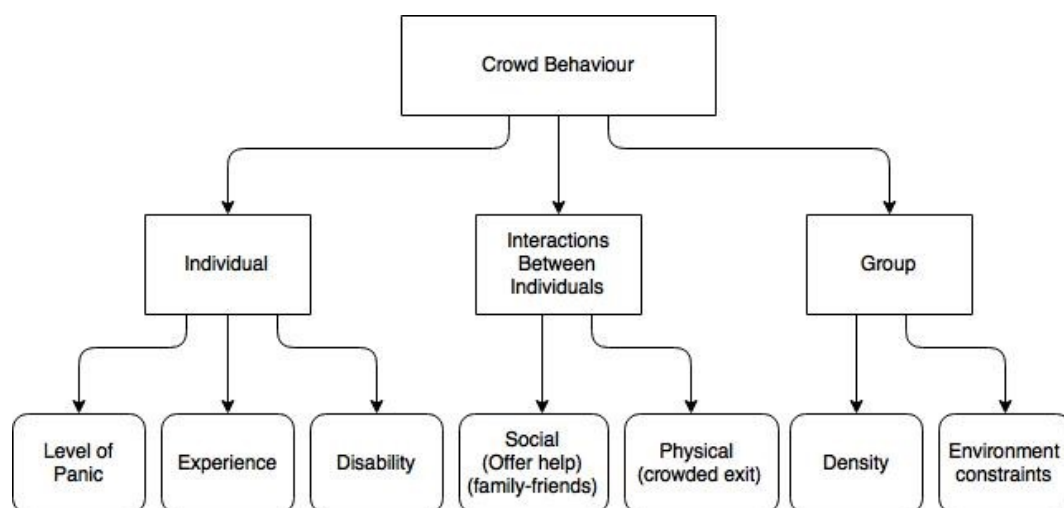


Figure 1. Crowd behavior.

### 2.1. The Individual

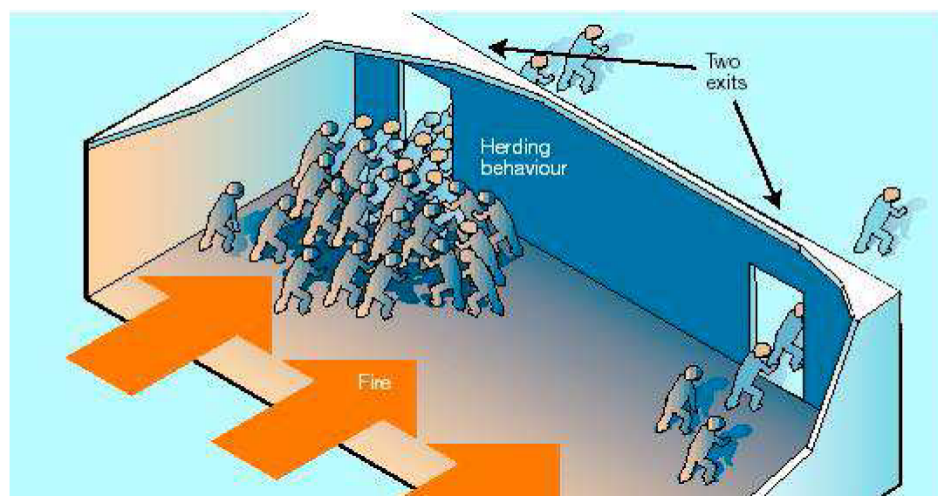
As we know, an individual can make a rational decision when the time is adequate, but an unwise decision if he or she is in hurry. For instance, in case of a building on fire, the individual is likely to follow instinct, attempting to get out of danger as quickly as possible without thinking. Therefore, many people may choose a main entrance or exit to escape from the building in fire. This is a quick reaction and no high level of rational decision making process is involved.

The evacuation process depends on the experience of individuals. Individual instincts may result in uncoordinated movements that slow down the whole crowd. In other words, experienced individuals can keep others calm and affect the crowd behavior in a good way. The panicked or inexperienced individuals can make the panic level of a crowd worse or chaotic [13].

## 2.2. Interactions between Individuals

In many evacuation scenarios, individuals may change their routes randomly and interact each another unintentionally. Some panicked people may be driven by their instincts and ignore some of the social norms, and as a result people can start pushing one another. Individuals pushed to the ground can be trodden on by other panicked evacuees and in turn obstruct the exit route. Moreover, the high level of encountered hazards can increase the possibility of negative interactions.

There are different types of interactions between individuals. In emergency situations, inexperienced individuals are more likely to follow other individuals. If an individual runs towards an exit, others may follow them without thinking, resulting in heavy congestion. Figure 2 shows a group of individuals who are trying to exit the room using one of the exits, which is called the herding or flocking, and the other possible escape route is less used [14].



**Figure 2.** Flocking problem in emergency evacuation [15].

## 2.3. The Crowd

In general, a crowd is formed by a collection of individuals, and its behaviour is inducted by individual behavior. The density of a moving crowd can change dynamically and influence the behavior of a crowd. For instance, the high density of a crowd can degrade the comfort level for its individuals, and cause a higher risk level. Also, the flow of a high density crowd forms a moving mess that can sweep individuals along with them. As a result, accidents and injuries are more likely to happen.

## 3. Crowd Simulation Methods

The methods for crowd simulation can be categorized according to the way they perform the simulation on the group as whole or counting each agent as an entity. Different scenarios in this section describe the idea behind crowd simulations, and some of the main simulation methods are explained. In the end, both the advantages and disadvantages of each model are discussed.

### 3.1. Two Types of Models

Crowd simulation usually refers to the representation of the movements and behaviors of a group of people, which is one of the main challenges in the modeling and simulation domain. Using computer simulation for studying crowd behavior aims to save time and resources since it does not require large numbers of participants to perform experiments. Moreover, using simulation to carry out an experiment is much faster than real life experiments in terms of data collection. The experiment

configuration change in simulation, such as the number of individuals, the movement speed and the experiment scenario, is much easier than the real experiments.

Working with a crowd means that we need to work with multiple entities and form a group of pedestrians, each of whom has the ability to interact with the environment according to the information they absorb. In particular, the motion planning problem of multiple agents moving simultaneously is that the number of possibilities or the degree of freedom could become large. The complexity of the situation depends on the number of the entities and the size of the environment, which is a major problem in this field of study [16]. Both centralized and decoupled planning approaches are key to addressing this problem.

The centralized approach treats all the robots as a single entity, and Cartesian product is used to form a single form of configuration in order to provide a solution for the next step [16]. Each additional robot will increase this single form of configuration and add more degrees of freedom. On the other hand, the decoupled planning approach proceeds in a distributed way and the path planning usually depends on exploring the environment space. In general, the decoupled methods [16] are faster than the centralized methods. They are less computationally demanding, but however do not guarantee optimal path, or sometimes it does not reach the final goal.

In general, various simulation models used in the crowd evacuation simulation could be divided into two types.

- The first type is the macroscopic-simulation model, which focuses on the building structure and its performance, and the group of people being evacuated will be considered as a moving unit. In this model, there are no individual characteristics for the pedestrians, and processing time is relatively small.
- The second one is focused on the individual as a singular unit, namely the microscopic-simulation model. By considering different factors that can affect the individuals of the moving crowd, this method is more realistic and accurate. However, it involves numerous objects and their interactions, which has several interaction factors that may lead to very complicated algorithms.

Figure 3 shows the two major simulation types and their methods.

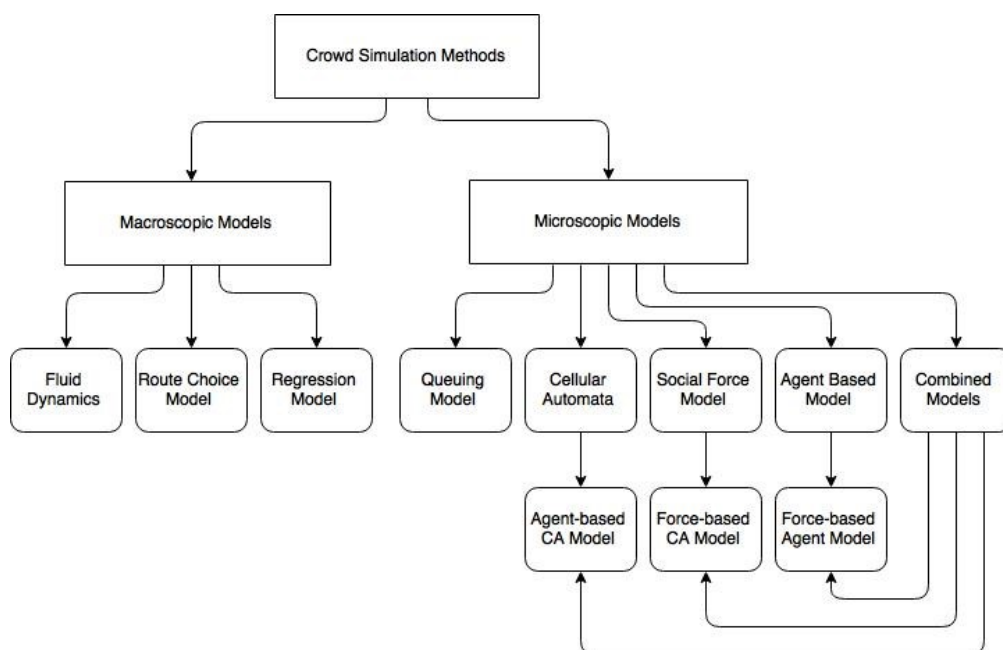


Figure 3. Crowd simulation methods. CA: cellular automata.

### 3.2. Macroscopic Models

This section will present different macroscopic models that consider a crowd to be a whole entity. A macroscopic model describes the movement as a flow at a macro level. This flow moves throughout the environment and could be applied on the group of individuals as a whole.

#### 3.2.1. Fluid Dynamics Model

Early simulation studies of crowd movement behaviors were based on fluid dynamics and the full Navier–Stokes equations are used to simulate large density crowd flow [17]. Later, the full Navier–Stokes equations were combined with additional properties derived from crowd behavior studies [15] to achieve more realistic modeling. Therefore, in an evacuation situation, a crowd of pedestrians can be treated as a flow of liquid trying to escape from a container. Individuals of a crowd could have different characteristics of walking abilities and different goals to achieve.

In [18], continuum behavior was derived by combining the Navier–Stokes equations with the results of the following three basic hypothesis:

- Hypothesis 1: Only three factors can affect the speed of a single pedestrian:
  - The density of the crowd that surrounds this individual.
  - The behavioral properties of the individual.
  - The terrestrial type where the individuals walk.
- Hypothesis 2: Individuals have a common sense of a task, and the crowd will try to execute the evacuation through the available exits.
- Hypothesis 3: Individuals will try to find the least time-consuming path in order to minimize the task time. However, using this hypothesis, a problem arises with respect to high crowd densities. Therefore, this principle applies as a function of density.

Both Hypothesis 1 and 2 were found to be related to crowd flow. Hypothesis 1 has many forms that describe the speed function, depending on the three factors mentioned above. However, this hypothesis is not efficient in case of extreme density crowds because of the huge contact forces between individuals. Early studies show that Hypothesis 2 has gained the trust of many researchers. This hypothesis affects the flow function by its focus on how pedestrians collect their surrounding information; some pedestrians acquire information from neighboring individuals. Hypothesis 3 depends on the end goal of the pedestrians or the reason of the motivation.

According to three hypotheses described above, the following equations describes the flow of an individual of a known type, which can be found in [15]:

$$-\frac{\partial p}{\partial t} + \frac{\partial \left( pg(p)f^2(p)\frac{\partial \varphi}{\partial x} \right)}{\partial x} + \frac{\partial \left( pg(p)f^2(p)\frac{\partial \varphi}{\partial y} \right)}{\partial y} = 0 \tag{1}$$

$$g(p)f(p) = \frac{1}{\sqrt{\left(\frac{\partial \varphi}{\partial x}\right)^2 + \left(\frac{\partial \varphi}{\partial y}\right)^2}} \tag{2}$$

where  $p$  is the crowd density and  $f(p)$  is the function of the speed for an individual related to the density.  $g(p)$  is a function related to the discomfort of the crowd.  $\varphi$  points to the walking time ahead, and is considered to be one of the special parameters of each individual.

Choosing suitable speeds and discomfort functions makes the equations easier to apply in 2D spaces, regardless of their time dependence and non-linearity. The detailed derivation of these equations can be found in [18]. This model is very efficient to deal with extreme high densities of pedestrians, where fluid dynamics model can provide more accurate simulation than other models. Furthermore, recent studies suggested that new forms of this model can represent some characteristics of the crowd behavior, such as blocking of corridors and obstacle avoidance.

### 3.2.2. Route Choice and Regression Models

Routes are continuous trajectories in time and space in the route choice model. In [19], the simulation starts with the predefined destinations for the individuals. After that, they choose different routes based on the utility function that indicates the benefits of the route, individual comfort and the travel time. The final route is selected by maximizing the utility.

In contrast, the regression model can be used for the prediction of the flow operations of pedestrians under certain circumstances. This depends on the statistically established relations of the flow variables. In [20], the flow features are based on the environment components such as stairs and doors, etc.

### 3.3. Microscopic Models

Microscopic models describe a group of pedestrians at the micro level. This means every individual in the group has its own characteristics. Models under this classification focus mainly on the behavior and the decision making of the individual, and the interactions between different individuals.

#### 3.3.1. Queuing Model

This model describes the environment as a network of walkway nodes. The pedestrian movement in this network can be modeled as a queuing network process. Each individual can be treated as an independent flow object moving through a queue of nodes. This model has the ability to model the interactions between different objects [21].

#### 3.3.2. Cellular Automata Models

The cellular automata model proposed in [22] has become a very popular simulation method, i.e., the micro-simulation approach. The dynamic system in the space of simulation was determined by the interactions between the various units. It has a better capacity for simulating the natural phenomena and different physical systems, and commonly used in biological, environmental and social areas [17]. The concept of cellular automata has been a good option among simulation methods for crowd dynamics simulation.

Based on defined cellular states and cellular space, a set of rules was studied in [17] to simulate the sheep mentality in the evacuation process. In [15], cellular automata is used to simulate evacuations with obstacle obstruction at the exit. In fact, a cellular automata system consists of four main components, namely Cell, Grid, Rule and Neighbor. In the CA method, the cellular space and the functions are responsible for the transactions in this space. A cellular automata system can be illustrated by the following formula [17]:

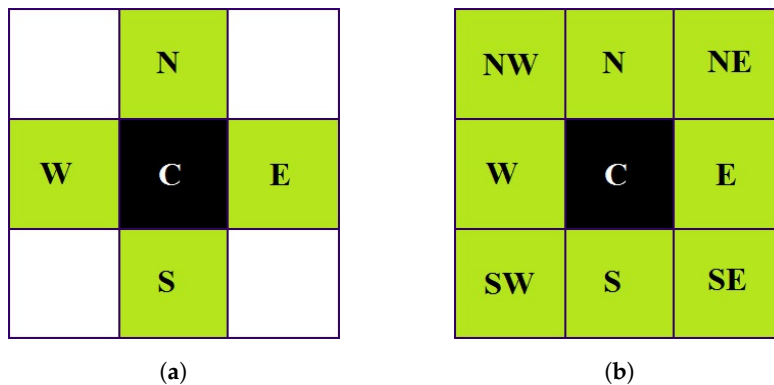
$$C = (d, S, N, f) \quad (3)$$

where  $C$  represents a cellular automata system.  $d$  is the system dimension and a cellular automata system in majority works in 2-dimensional space.  $S$  represents this set of states.  $N$  illustrates the state of the cells located in the neighbourhood space,  $N = s_1, s_2, \dots, s_i$ , where  $i$  is the number of cells. Then,  $f$  represents the set of rules controls the transfer from state to state for a cell.

To start using a cellular automata system for a crowd simulation, we could assume that the initial location for an individual is randomly generated or pre-defined. As shown in Figure 4, the movement direction for each person is from its current location to a neighbor cell. More specifically,

- The four Neumann movements of a cell  $C$  are shown in Figure 4a.
- The eight Moore movements for a cell  $C$  are given in Figure 4b [18].

Conflicts between two pedestrians can also be represented in this model by following set of characters such as the strength of the cell/pedestrian to decide which is going to occupy this empty cell.



**Figure 4.** Cell and its neighbors describing the available movements for a cell [18]. (a) Neumann movements; (b) Moore movements.

Movement direction for each person to walk from its current location to a neighbor cell can be defined according to one of the neighbor sets shown in Figure 4. The original version of cellular automata defined the neighbor as a cell which shares one side at least. A Moore neighborhood added more flexibility for the movement available and its current cell shares at least a corner with the neighboring cells [23]. Conflicts between two pedestrians also can be represented in this model by following set of characters such as the strength of the cell/pedestrian to decide which is going to occupy this empty cell.

A cellular automata system has various characteristics, such as the grid type that has three options: Triangular, rectangular and hexagonal grid. Each type of grid has its special use. For example, the hexagonal grid is suitable to represent complex obstacles, but not the walls. Moreover, the rule update type is another characteristic that has to be defined for each cellular automata system. The way for the rules to be updated has a significant impact on the crowd dynamics. Rules can be updated in two basic ways, namely the sequential and the parallel update ways.

- In parallel update, the rule system updates the individual locations at the same time, and may result in conflicts.
- In sequential update, no conflict problem will be generated since the occupied cells are forbidden for a moving pedestrian.

### 3.3.3. Social Force Model

Human behavior was traditionally considered to be irregular or almost “chaotic”, which is especially true in dangerous situations. Von Neumann proposed the social force model in 1951, in which social forces affect the change of human behaviors [22]. It is an alternative to the cellular automata model mentioned above. The social forces are treated as a set of determinations of a pedestrian in [24], in which a pedestrian can be affected by three main factors below:

- Pedestrians try to follow the shortest way available in order to reach the destination as comfortably as possible.
- Each pedestrian has an influence on the others, and wants to keep a safe distance from other pedestrians and the obstacles such as the walls. This depends on the crowd density and the desired speed.
- In some cases, pedestrians can be attracted by the objects such as exit doors or by other individuals such as family members or friends, or even rescue members in the case of an emergency evacuation.

Since each pedestrian is trying to keep a distance from the exit door and the other pedestrians, the social force model has its limitations such as deadlocks and blockage of an exit that may happen in some cases.

### 3.3.4. Agent-Based Model

Agent-based modeling has been used in many different areas, from stock market agent behavioral simulation to many other kinds of simulations such as medicine, nature studies, and many more. Recently, agent-based modeling started to play a key role in simulating the different human behaviors. The agent-based model is able to simulate and model complicated systems, in which multiple autonomous agents could interact and influence each other.

Agents can also be modeled individually with different characteristics to make them act independently and enhance the reliability of the system as a whole [25]. These characteristics are necessary to identify these agents as independent components of an application. These essential characteristics can be short-listed below [26]:

- Agents have to be independent, objective and distinguishable individuals. It is necessary to allow every individual/agent to be identified by others through special set of attributes.
- Agents are autonomous. This characteristic allows the agents to act independently within the space they are applied to, and sensed data can be analyzed using the corresponding behaviors to produce suitable actions and decisions. Behaviors can be built up using any chosen method from simple if-else statements to genetic algorithms if required.
- Every agent has a state that consists of sets of attributes related to the current status for the agent, and can change over time. Its whole status can be taken from the states of all the agents and the environment.
- An agent is an interactive component of the system with the ability to interact with other agents that affect its behavior. Interactions between agents work according to a rule base that governs all the activities of the agent such as communications, movements and collision avoidance.

In addition to the above characteristics there are some useful other characteristics:

- The ability to be adaptive, so that the behavior of the agent can be modified or upgraded. This can be achieved by using suitable rules. Agents can collect information from different experiences and use it to adapt or learn by modifying their behaviors accordingly and this is on the individual level. Furthermore, at the system level, the process of agent-type selection can be adaptive to the environment. The agent with better suitable properties should be used to complete the mission.
- An agent can be goal guided, so a comparison between the objective and the behavior related to the current situation can take place in order to enhance the resulting response.
- Agents may be “heterogeneous”. In other words, attributes and behaviors of different agents can diverge in their range and complexity. This diversity may depend on multiple factors such as the size of memory responsible of storing the previous events or experiences and the number of attributes will be considered in the decision making process.
- In an agent-based model, all the data related to an agent can be divided into two sections: attributes and method. Figure 5 illustrates a typical agent. Attributes such as the name, weight, and height are all to be considered as static. Other attributes are dynamic and can change through the running time. Agent methods are mostly describing the behaviors of the agent and the rules that relate the current state with different available responses, and even the functions responsible for modifying the existing behaviors and rules.



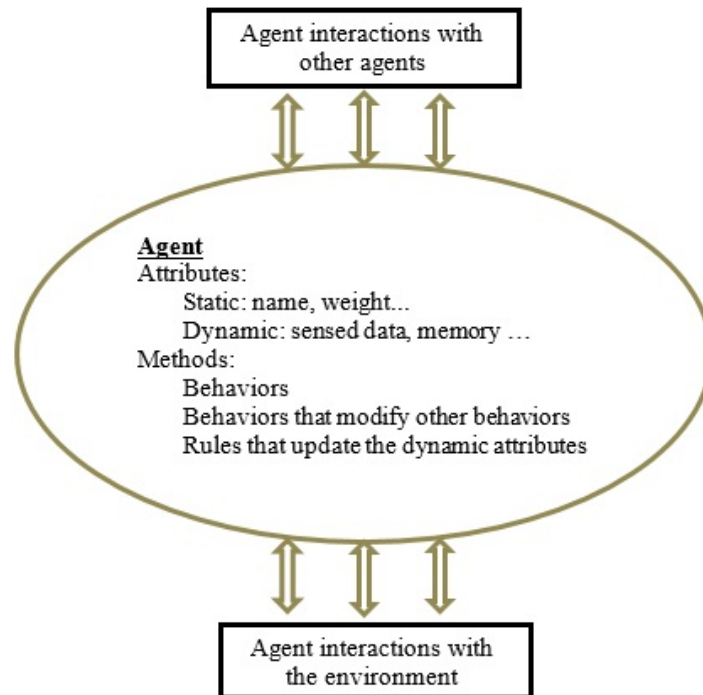


Figure 5. A typical agent [26].

As mentioned above, an agent-based system is concerned about the interactions between different agents. Therefore, the system will be evaluated as a whole through the group of agents interacting with each other. Information available to each individual is limited to the local surroundings. An agent can use the available behaviors and methods to collect data by interacting with other agents and with the environment. There is no global information broadcasting. Agent-based simulations are used to evaluate the results of interactions among agents, and also be combined with other models such as fluid model previously mentioned. Different agent-based models can be combined in order to minimize the required computational power. Some simulations can use a combination of models in order to model different parts of the system, i.e., creating a bridge between macroscopic and microscopic simulation methods [27].

The agent-based simulation model is suitable for large scale of systems that contain many components. Working with this model is easier than other models from the prospective of simplicity since dealing with rules and attributes is much easier than the complicated equations that other models depend on. However, this model needs a huge amount of computing power to process the large numbers of individual agents simultaneously. In contrast, the crowd in other methods is being processed as one flow. Fortunately, rapid advances in hardware capabilities make the computing power required by agent-based model less than a problem.

#### 3.4. Data-Driven Model

Other models choose a different way of simulating a crowd depending on real crowd movement analysis. Combining the realism of data-driven methods and the interactivity of multi-agent techniques a simulation method can be formed to give movement patterns based on real life interactions. Standard tracking algorithms can be used to obtain 2D trajectories from videos of real life crowd moving in different situations, from normal walking to emergency evacuation panic scenario. After that, these trajectories can be used in adaptive data-driven crowd simulation, where agents will be affected by the data extracted from the observed crowds. Velocities and directional movements will be given according to the collected data, which will govern the behavior of each agent. Multiple agents can exhibit similar variables in terms of space and time, resulting in similar movement trajectories [28].

Combining data-driven pedestrian dynamics with other agent-based models can give agents different behavior and trajectory depending on the observed surroundings in the environment. Different factors can influence the resulting behavior such as the density of agent, which refers to the number of agents per square meter. Collision avoidance can play a major role in changing the trajectory and speed for moving agents. On the other hand, changes in the situation can also result in behavioral changes. In [29], authors proposed stressor-based behavior model, where the behavioral changes are modeled by updating the simulation parameters of the affected agent. Agents tend to move away from stressors, the amount of change in trajectory and speed depends on the nature of the stressor. Data-driven adaptive crowd simulations show higher similarity to the observed crowd movements. However, the resulting trajectories may not give similar movements as real crowd in some cases, where extra calculations are involved because of stressor effect or high density environments. The performance of this model depends on accuracy of the input trajectories. Where high density crowds can be a challenge to observe and extract accurate trajectories to be used in this model. Also, the layout of the environment is a main factor in this model, any changes to the environment could result in different agent behavior [28].

#### 3.4.1. Combined Models

As discussed previously, different modeling approaches have their strengths in specific areas of simulation [30], but also have their weakness as well. To enhance the strengths and eliminate the limitations, many works have suggested the use of the combined modeling methods, namely “hybrid models”. The following models are examples of hybrid models:

- Agent-based CA model: This model introduced two new characteristics using CA as environment representation. The first was the shape of the cell, original shape was a square shaped cell. The new model suggested a triangle-shaped cells, which gives new ways of movements. The other new addition was the neighboring cells list, which was extended to include cells that have a distance larger than one. This model shows that advanced agent-based models could be built using simple CA model through combining the intelligent agents with the simple movement approach by CA. However, this model suffers from the imprecise position implementation just like the CA models [31].
- Force-based CA model: Few studies considered combining the CA modeling approach with force-based model. In [32] authors proposed a model that combines these two approaches, where the new model can benefit from the computational efficiency of the CA model, and the ability to perform complex behavior interactions that can be provided by the force-based model. The results showed similar results to the social force model. Because the new model is based on the CA model it has the computational advantage. However, due to the different representation of the individual movements, these two models are rarely used in a combination [32].
- Force-based agent model: In this model, the agent structure and the decision making process are similar to the agent-based model. The resulting decided behaviors will then be combined according to formulas defined using the force-based modeling method. There are two layers for this model: The top layer represents the agent-based model, responsible for the behaviors the agents picks to perform. On the bottom layer, the force-based model takes responsibility of combining the resulting behaviors. Combining these two models sums the strengths of both of them, and the weaknesses are reduced. The force-based model has the precise movement representation, but lacks the precision in decision making. On the other hand, the agent-based model focuses on decision making and intelligent behaviors, but lacks the movement representation. In result, the combination of the two models will benefit from combining the needed components of each model with the other. However, it was noted that combining different behaviors could be difficult due to the non-standard methods of measuring the effects of agent parameters within the force-based formulas [33].

### 3.4.2. Discussion

In this section, we have outlined a number of the crowd models that belong to microscopic models, namely queuing models, cellular automata models, social force models, agent-based models and the combined models. Each of these models has its advantages and disadvantages. A combination of both models could achieve better results. More specifically, it is clear to see the following factors:

- The force-based models take a lot of computing time in order to provide a continuous location of the pedestrian.
- The cellular automata model is discrete in time and space, and makes the crowd simple, with and fewer characters for individuals.
- Agent-based models enable agents to communicate with each other although the computation cost is high. These models have many rules and some constrains on the moving speed.
- The hybrid force-based agent model can almost eliminate their weaknesses. However, there is no general method to link the parameters of the agent with the movement computations from the force-based model.

## 4. Existing Work for Crowd Simulation

This section outlines some of the existing work on crowd simulation applications according to the categorization. Some of these methods could be greatly enhanced by adding more factors that can give the human agents different abilities. The decision making process can include different effects related to the sociological, physiological, and psychological models.

### 4.1. Helbing's Model

Helbing proposed a model to simulate the pedestrian behaviors in different panic situations [22]. It is based on physics and sociological and psychological forces affecting the movements of a pedestrian in a crowd. Each individual of a crowd will be treated as a particle in this model, and all these particles are given a mass and a velocity. The observations from real crowd panic determine the given characteristics. The distance, velocity, mass and different forces are concluded from the analysis.

More specifically, every individual will be defined as a particle  $i$  of the mass  $m_i$ , a predefined speed is given by  $v_i^0$  and a given direction as  $e_i^0$ . The particle will try to adjust its practical speed  $v_i$  in a given time interval  $\tau_i$  to achieve the desired velocity and direction. This is described in the first part of the equation below.

$$m_i \frac{dv_i}{dx} = m_i \frac{v_i^0(t)e_i^0(t) - v_i(t)}{\tau_i} + \sum_{j \neq i} f_{ij} + \sum_w f_{iw} \quad (4)$$

However, other forces, i.e., interaction forces  $f_{ij}$  and  $f_{iw}$ , influence the entities, as they will try to keep a suitable distance from other entities  $j$  and walls  $w$ . The change in speed at time  $t$  for every particle  $i$  is given by the acceleration Equation (4).

Three types of social forces compose the interaction force  $f_{ij}$  between different entities. The first one is the psychological effect and described by a repulsive interaction to keep two pedestrians away of each other. The second force is the body force to oppose the body compression, and the last one is the sliding friction force to stop the random motion [22].

This model has been deployed to simulate and test crowd movements in 2D environments. Different scenarios have been tested such as passing through a narrow passage and exit blocking patterns. In addition, many social forces have been proposed to be used with this model to simulate various interactions between two pedestrians. The calculation cost of this model is too high due to the large number of factors that need to be modeled.

#### 4.2. Virtual Reality LAB (VRLab)

Researchers working on VRLab have achieved a remarkable progress in crowd simulation and modeling, in particular animating and visualizing virtual crowds of pedestrians. These technologies have been used in computer games and training systems. Different simulations have been done in common environments such as terminals and urban areas. They used an agent-based model called ViCrowd [24], which depends on a hierarchical modeling system where a crowd can be described as a various smaller groups and each group consists of few individuals. ViCrowd can be used to model different social groups (friends, family, etc.) in an easy way, which gives the simulation more flexibility in modeling social factors within the crowd simulation [25].

As shown in Figure 6, agents in VRLab are represented by 3D entities. In contrast, Figure 7 shows the components of VRLab. Each individual is given some knowledge about the environment, in order to be able to perform path planning, obstacle avoidance. This knowledge can be achieved by defining all objects in the environment and the free walk areas. There are three types of behaviors can be used in this system, programmed, reactive, and guided behaviors, where a script language is developed in ViCrowd for the user to define the desired behavior. The decision making process depends on finite state machines (FSMs). Although the Virtual reality Lab has a good crowd animation, the downside is the computational costs of 3D rendering are high, especially when it is used for high density crowds.



Figure 6. Virtual Reality LAB (VRLab) 3D environment [24].

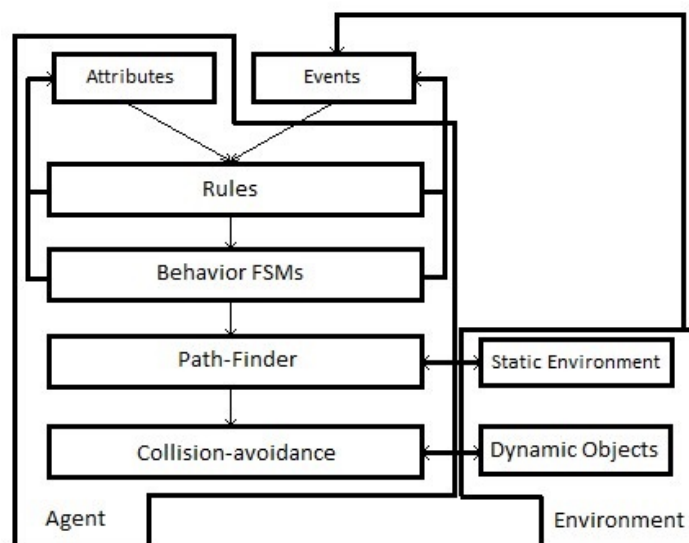


Figure 7. VRLab system [25]. FSM: Finite state machines

### 4.3. High-Density Autonomous Crowds

HiDAC [34] combines a social forces model and rule-based model. It enables a variety of behaviors and can simulate different types of crowds, ranging from extreme panic situations such as fire evacuation to high-density crowds such as leaving a cinema. Moreover, it allows a number of different behaviors to be exhibited simultaneously for heterogeneous crowds.

Unlike cellular automata and rule-based models, HiDAC can realistically simulate an individual trying to push others and force its way through a crowd. Also, HiDAC agents can exhibit more respectful behaviors when desired and make decisions and queuing when necessary, which differs from social forces models. These emergent behaviors are driven by the combination of psychological and physiological rules together with a social forces model. Impatience has been integrated in HiDAC to avoid the sheep-like behavior that many crowd simulation models exhibit.

Figure 8 shows the system architecture for HiDAC. As can be seen, it shows a wide range of emergent behaviours, e.g., from agent line formation to physical reaction behavior and its consequences. These behaviors are relative to the current situation, personalities of the individuals and perceived social density.

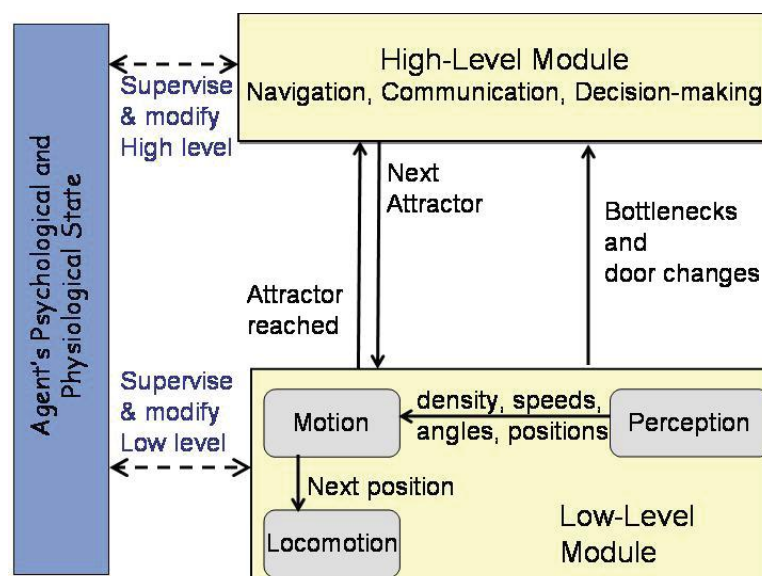


Figure 8. High-density autonomous crowds (HiDAC) architecture overview [34].

### 4.4. MACES and PMFs

In addition to dealing with low level motion based on social forces, a crowd simulation system must be able to explore unknown environments and obtain a cognitive map for navigation purposes. It adds communication and roles to achieve individualistic behaviors and a realistic way to spread information about the environment.

Performance moderator functions (PMFs) can be used to expand the range of realistic human behaviors, e.g., stress, emotion, decision theoretic and motivation sources. The proposed architecture that integrates MACES and PMFs can add validated agent behaviors to crowd simulations. A psychological model is deployed into a crowd simulation system in order to obtain believable emergent behaviors [34].

### 4.5. Crowd Dynamics

Crowd Dynamics Ltd has developed a set of tools used in simulating different scenarios for crowd behavior in an evacuation. The main focus is on the observation and analysis of events involving large

crowds. Their tools can support police and security forces in case of emergency situations, and can also be used for planning events for large numbers of people.

On the other hand, Legion is another agent-based modeling and simulation tool used mainly for the study of crowd dynamics, e.g., a global path planning method in order to control the agents. Since this method is expensive in computations, a less demanding algorithm was introduced to solve the computational problem. The path planning is the only focus, thus no more decision making is required. Social and psychological factors are not much of concern in this model, where it is used in operational analysis only [35].

This tool uses some 2D structures and paths to describe the environment for visualization. Agents are represented by simple 2D icons or just simple dots. A new version of the tool has been introduced recently, namely Myriad, which is an evacuation model in 3D. This tool aims at understanding the spatial aspect of the building evacuation. An example of the 3D tool Myriad is given in Figure 9.

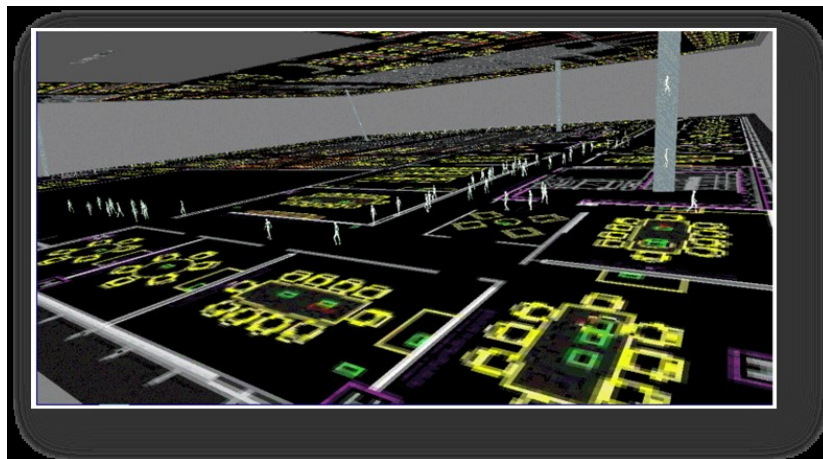


Figure 9. Crowd dynamics simulation environment [35].

## 5. Robots in Evacuation Simulation

Many types of software can be used to model a crowd of people in an evacuation situation, including the simulation systems mentioned in Section 3. STEPS and Fire Dynamics Simulator with Evac (FDS+Evac) are specialized software tools provide an easy way to model and simulate a crowd of people in panic situations and environments. These tools are free and available, providing the ability of behavioral and hazard modeling. The basic requirements for evacuation simulation should have the following 4 characteristics [36]:

- The simulator should be able to perform real-time interactions between the objects in the environment.
- The simulator should have the ability of performing numerical operations stably and in real time mode.
- The simulator should be able to model the rigid and soft bodies in a realistic manner.
- The simulator should be able to add new models of sensors, communication packages and actuator systems.

Apart from the traditional software that is available for simulating an evacuation, some new useful tools can be used for simulating the robot-assisted evacuation and satisfy two main requirements listed below:

- The crowd of people should react to the panic situation in a realistic way, and collisions have to be taken into consideration.
- The robot models should be able to move and react in a realistic way, and sensed information and control methods have to be independent for each robot.

The survey in [37] covered many tools used for dynamics simulation of robots based on user feedback in different stages of study (BS, MS, and PhD) and shows that the most important standards or requirements are as follows:

- Realistic simulation ( or close enough)
- Open source.
- Code generated can be used for both simulation and real application.
- Light and quick.
- Ability of modifying.
- No physical interpenetration between simulated objects.

Zhang and Guo used potential fields to model the physical environment during a fire emergency and simulated evacuees in a similar manner [38]. In [39] the authors proposed a panic propagation model to describe how crowds behave in panic situation. Combining this panic representation with the social force model gives a self-driven force that can interact with other evacuees and affect the crowd mobility. Utilizing robotic help, an algorithm was developed to give the pedestrians the best exit with the shortest escape time. The robot calculates all the data collected from the surrounding environment, providing the best exit available. An evaluation shows the improvement in evacuation times. Boukas et al. developed a new crowd modeling technique using cellular automata and tested it by having a robot provide guidance to human participants in a simulated evacuation [40]. The participants trust in the robot was not evaluated and the authors did not report it motivating participants to find an exit quickly. Instead, the authors used this experiment to validate their simulations on crowd movement using robot guidance. The simulation was carried out in real-world conditions, and the evacuation times showed remarkable performance in terms of speed and ease of use. In [41] authors used Unity game engine to create a crowd evacuation simulation. The crowd model used a combination of both the social force model and the agent-based model, and the resulting crowd model is able to simulate a crowd of people evacuating under different set of circumstances, panic, social bonds and injuries could be applied to the model. Results showed the difference between the evacuation times when the crowd performed the evacuation without robot assistance and when a robot provided alternative routes to less crowded exits. Results showed the benefits of the robotic assistance in emergency evacuation.

It should be noticed that the Robot Operating System (ROS) and Gazebo, Bullet, ARGoS, Webots, V-Rep, MRDS, and MATLAB with Simulink are widely used by researchers involved in this field of study.

### 5.1. ROS and Gazebo

Gazebo is 3D multi-robot simulator; simulated objects can behave realistically in Gazebo as each object has its mass, shape, friction and many other attributes. In order to render 3-D objects including robots and other dynamic objects in the simulated environment, Gazebo utilizes the Object-Oriented Graphics Rendering Engine (OGRE) [42] and Open Dynamics Engine (ODE) [43]. Gazebo is well integrated with the Robot Operating System (ROS). It works as a ROS node and interactions between Gazebo and other ROS nodes can be carried out using mixture of message through ROS plug-ins provided by Gazebo simulator. These plug-ins are dynamically loaded into Gazebo, providing other nodes with the ability to create and control objects in the simulation having full access to all the available APIs the 3D simulator provides. Gazebo and ROS have many re-usable components that can be used in robotics research. In [40] the authors used ROS to implement an evacuation study scenario depending on the CA model for crowd behavior simulation and for robot navigation implementation.

Both simulated and real environment experiments were conducted and Gazebo simulator was used for the visualization purposes. The results of this study showed a custom-made robotic platform combined with both real-world and simulated evacuation implementation, showing remarkable evacuation performance using robots in an evacuation scenario. Also, ROS has many related packages connected to this study, one of them is the social force model based on Helbing studies [5]. This package offers the necessary abilities to perform a crowd simulation from individual walking and social interactions, with the ability to connect with Gazebo for physics reasoning and 3D visualization.

### 5.2. Webots

Webots [35] can be used for simulating different types of robots with various types of sensors. It has the following features:

- Based on ODE for rigid body simulation, Webot is a commercial library used for simulating the dynamics of rigid body.
- Its advanced user interface makes its user interacting with the environment and objects very easy.
- Its programs can be written with different programming languages such as C, C++, MATLAB, and Python.
- Webots can interact with different other tools and applications through TCP/IP.
- It can work with different types of operating systems, such as Windows, Linux or MAC.

### 5.3. MRDS

Microsoft Robotics Developer Studio (MRDS) is a windows-based simulation tool and can be used to create models for many types of robotic systems. It has the following features:

- MRDS uses a visual programming language (VPL) to program a robotic application or behaviour, which can be done by connecting multiple blocks to form a process.
- MRDS can work with a 3D visual simulation environment or a real life robot.
- Its visual simulation environment uses the NVIDIA PhysX to give the physics behaviour and XNA framework to enable 3D real-time graphics rendering [44].

### 5.4. Other Simulation Tools

Other open source simulators include ARGoS, Bullet, and V-Rep, that are freely available. More specifically, they have the following features:

- ARGoS is an advanced robot simulator with a modular and multi-threaded architecture, working with Linux and MAC [45].
- Bullet is designed for robotics, allowing high quality physics simulations, especially for animation purposes and research projects [46].
- V-Rep is designed to work on both Linux and Windows systems. It uses C++ and LUA for programming and is able to add new modules and sensors in a flexible way.

Table 1 shows the most important features for the popular robot simulation tools.

### 5.5. Unity3D Game Engine

Another option is to use one of the game engines, which provides a combination of physical engine and rendering engine. One of the engines considered is the USARSim, which was launched in 2003, built upon Unreal Engine 2.0, used to produce many simulations and even games. This engine lacks for proper documentation and the engine editor suffers from crashes.



**Table 1.** Simulation tools characteristics.

	ROS and Gazebo	MRDS	MATLAB	Webots	ARGoS	V-Rep	Bullet
<b>Simulator Type</b>	3-D	3-D	3-D	3-D	3-D	3-D	3-D
<b>Documentation</b>	Low Level	High Level	High Level	Limited	High Level	High Level	High Level
<b>Programming Language</b>	C++, Python Lisp, Java, LUA	VPL, C# JScript, Python	MATLAB C, C++, Java	MATLAB, C/C++ Java, Python	C++	C++, LUA	C++
<b>Sensors</b>	Various	Camera Range F Odometry	Various	Various	Various	Various	Various
<b>GUI</b>	Yes	Yes	Can be Built	Yes	Yes	Yes	Yes
<b>Realistic Modelling Level</b>	High	High	Medium	Medium	High	High	Medium
<b>Operating System</b>	Linux	Windows	Win/MAC /Linux	Win/MAC /Linux	Linux/MAC	Linux /Windows	Win/MAC /Linux

As a game engine, the Unity3D engine was developed by the Unity technology based in Denmark. Unity3D engine combines the NVidia PhysX engine and a custom-made rendering engine with Mono. The PhysX engine is able to simulate complex effects like 3D movement 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 language to write scripts to control every part of the simulation. The reason for using the Unity engine is clarified in this section in detail; we considered key features to show that Unity engine can play a very important role as a robot simulation engine.

- **Extensive Documentation:** Complete documentation with wide range of examples is available. The API is explained in details within these examples. This is a big advantage for beginners where it leads to faster productivity when compared with other tools we mentioned earlier.
- **Environment and Physics:** the renderer in the Unity engine can provide the needed texturing that will affect the visualization of the scene, with dynamic shadows for the moving objects. It is specialized to create games, so it will have no problem supporting large scale simulation. The physics properties are not less effective than the visuals. Different objects are supported, with the ability to add mass, drag bounciness and different forces. It also provides a collision detection system. The C language provided by Mono is provided with a Unity Class that simplifies the physics scripting.
- **Multi-platform and Developer community:** the Unity engine runs on different platforms OSX, Windows, Android, iOS and different game consoles, and each one of these distributions can build the application being developed for all other supported platforms. New users can benefit from an active on-line developer community. Many features in the Unity engine have been added to support realistic simulations.
- **Comprehensive API:** all objects in the environment are listed in a tree. Multiple script codes can be assigned to each object. Drag-n-Drop technique is used to add contents to the environment and objects Figure 10 shows the Unity editor interface. We can pass parameters to the C scripts through the editor window which makes it easier to change the parameters during the simulation.

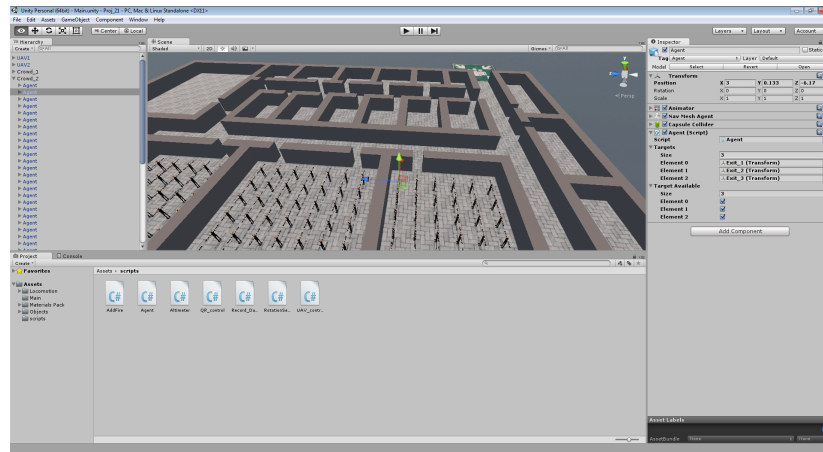


Figure 10. Unity3D Interface.

## 6. Conclusions

This survey reviews the research and development of the crowd modeling and crowd simulation. Firstly, the main hierarchy of the crowd behavior analysis is introduced, where behavior of the individual, as well as interactions between individuals and the crowd have been summarized. Then the main categories of the crowd modeling are introduced and discussed in order to give a thorough idea about the crowd modeling methods that have been recently developed. The advantages and disadvantages of each model have been discussed with a main focus on the microscopic approaches in particular. Furthermore, the existing work on crowd simulation are analyzed based on the mentioned modeling categories, including Helbing and HiDAC models. Finally, the importance of using robots to aid in emergency evacuations, as well as search and rescue missions is demonstrated, and some of the popular simulators that allow simulating robots and crowds of pedestrians are discussed.

**Author Contributions:** Ibraheem Sakour collected the related literature and conducted the analysis and comparison as well as completing the draft under the supervision of Huosheng Hu, who has contributed to the revision of the paper structure and the presentation style, as well as the proofreading of the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Liu, M.; Lo, M. The quantitative investigation on people's pre-evacuation behaviour under fire. *Autom. Constr.* **2011**, *20*, 620–628.
2. Kobes, M.; Helsloot, I.; De Vries, B. Building safety and human behaviour in fire: A literature review. *Fire Saf. J.* **2010**, *45*, 1–11.
3. Ansonge, R.; Thomas, S. *Mathematical Models of Fluid Dynamics*; Wiley-VCH: Berlin, Germany, **2009**.
4. A New Kind of Science. Available online: <http://www.wolframscience.com/nksonline/toc.html> (accessed on 5 April 2017).
5. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Phys. Rev.* **1995**, *51*, 42–82.
6. Luo, L.; Zhou, S.; Cai, W.; Low, M.; Tian, F.; Wang, Y.; Xiao, X.; Chen, D. Agent-based human behavior modeling for crowd simulation. *Comput. Anim. Virtual Worlds* **2008**, *19*, 271–281.
7. Nuria, P.; Ali, M. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Autom. Constr.* **2008**, *17*, 377–385.
8. Dylan, S.; Maja, M. Insights Toward Robot-Assisted Evacuation. *Int. J. Robot. Soc. Jpn.* **2005**, *19*, 797–818.
9. Bhatia, S.; Dhillon, H.S.; Kumar, N. Alive human body detection system using an autonomous mobile rescue robot. In Proceedings of the 2011 Annual IEEE India Conference, Hyderabad, India, 16–18 December 2011.
10. Chikahito, N.; Massimiliano, P.; Bernd, H.; Tomaso, P. Full-body person recognition system. *Pattern Recognit.* **2003**, *36*, 1997–2006.

11. Mohan, A.; Papageorgiou, C.; Poggio, T. Example-Based Object Detection in Images by Components. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 349–361.
12. Cavalcanti, C.; Gomes, H. People Detection in Still Images Based on a Skin Filter and Body Part Evidence. In Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, Natal, Brazil, 9–12 October 2005.
13. Symington, A.; Waharte, S.; Julier, S.; Trigoni, N. Probabilistic target detection by camera-equipped UAVs. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010.
14. Suzuki, T.; Miyoshi, D.; Meguro, J.; Amano, Y.; Hashizume, T.; Sato, K.; Takiguchi, J. Real-time hazard map generation using small unmanned aerial vehicle. In Proceedings of the SICE Annual Conference, Tokyo, Japan, 20–22 August 2008.
15. Hughes, L. The Flow of Human Crowds. *Annu. Rev. Fluid Mech.* **2003**, *35*, 169–182.
16. LaValle, S. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
17. Waharte, S.; Trigoni, N. Supporting Search and Rescue Operations with UAVs. In Proceedings of the International Conference on Emerging Security Technologies EST, Canterbury, UK, 6–7 September 2010.
18. Hughes, L. A continuum theory for the flow of pedestrians. *ACM Trans. Res.* **2002**, *36*, 507–535.
19. Hoogendoorn, S. Pedestrian travel behaviour modelling. In Proceedings of the 10th International Conference on Travel Behaviour Research, Lucerne, Switzerland, 10–14 August 2003.
20. Milazzo, J.; Roupail, N.; Hummer, J.; Allen, D. Effect of Pedestrians on Capacity of Signalized Intersections. *Transp. Res. Rec.* **1998**, *5*, 37–46.
21. Løvås, G.G. Modelling and simulation of pedestrian traffic flow. *Transp. Res.* **1994**, *28*, 429–443.
22. Neumann, J. The general and logical theory of automata. *Cereb. Mech. Behav.* **1951**, *1*, 289–326.
23. Winter, H. Modelling Crowd Dynamics During Evacuation Situations Using Simulation. Lancaster University, 2012, pp. 1–21. Available online: <http://www.lancs.ac.uk/~winterh/CrowdDynamics.pdf> (accessed on 5 April 2017).
24. Zhou, Q. Study on the Evacuation Simulation based on Cellular Automata. In Proceedings of the International Conference on Information Technology and Computer Science, Kiev, Ukraine, 25–26 July 2009.
25. Varas, A.; Cornejo, D.; Mainemer, D.; Toledo, B.; Rogan, J.; Munoz, V.; Valdivia, A. Cellular automaton model for evacuation process with obstacles. *Phys. A* **2007**, *382*, 631–642.
26. Macal, C.; North, M. Towards Teaching Agent-based Simulation. In Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010.
27. Silverman, B.; Badler, N.; Pelechano, N.; O'Brien, K. Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication. In Proceedings of the First International Workshop on Crowd Simulation, Lausanne, Switzerland, 24–25 November 2005.
28. Aniket, B.; Sujeong, K.; Dinesh, M. Online Parameter Learning for Data-driven Crowd Simulation and Content Generation. *Comput. Graph.* **2016**, *55*, 68–79.
29. Kim, S.; Guy, J.; Manocha, D.; Lin, C. Interactive simulation of dynamic crowd behaviours using general adaptation syndrome theory. In Proceedings of the Symposium on Interactive 3D Graphics, Costa Mesa, CA, USA, 9–11 March 2012; pp. 55–62.
30. Zheng, X. Modelling crowd evacuation of a building based on seven methodological approaches. *J. Build. Environ.* **2009**, *44*, 437–445.
31. Bandini, S.; Federici, M.; Vizzari, G. Situated Cellular Agents Approach To Crowd Modelling and Simulation. *J. Cybern. Syst.* **2007**, *38*, 729–753.
32. Song, W. Evacuation behaviours at exit in CA model with force essentials: A comparison with social force model. *Phys. A Stat. Theor. Phys.* **2006**, *371*, 658–666.
33. Qiu, F.; Hu, X. Modelling group structures in pedestrian crowd simulation. *J. Simul. Model. Pract. Theory* **2010**, *18*, 190–205.
34. Still, K. Crowd Dynamics. Ph.D. Thesis, University of Warwick, Coventry, UK, July 2000.
35. www.bespokeinternet.com, B. Myriad II—Evacuation Module—Crowd Dynamics | Crowd Management, Crowd Modelling, Crowd Behaviour. Available online: <http://www.crowddynamics.com/evacuation-demo.php> (accessed on 2 December 2016).

36. Staranowicz, A.; Mariottini, L. A Survey and Comparison of Commercial and Open-Source Robotic Simulator Software. In Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments, Heraklion, Greece, 25–27 May 2011.
37. Ivaldi, S.; Padois, V.; Nori, F. *Tools for Dynamics Simulation of Robots: A Survey Based on User Feedback*; Extended Report; ArXiv e-Prints: New York, NY, USA, 2014.
38. Zhang, S.; Guo, Y. Distributed multi-robot evacuation incorporating human behaviour. *Asian J. Control* **2015**, *17*, 34–44.
39. Tang, B.; Jiang, C.; He, H.; Guo, Y. Human mobility modelling for robot-assisted evacuation in complex indoor environments. *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 694–707.
40. Boukas, E.; Kostavelis, I.; Gasteratos, A.; Sirakoulis, G. Robot Guided Crowd Evacuation. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 739–751.
41. Sakour, I.; Hu, H. Robot Assisted Evacuation Simulation. In Proceedings of the 2016 8th Computer Science and Electronic Engineering (CEECE), Colchester, UK, 28–30 September 2016.
42. Jaillet, L.; Simeon, T. A PRM-based motion planning for dynamically changing environments. In Proceedings of the IEEE/RSJ International Conference on IROS, Sendai, Japan, 28 September–2 October 2004.
43. Open Dynamics Engine—Home. Available online: <http://www.ode.org/> (accessed on 2 December 2016).
44. Products and Services. Available online: <https://msdn.microsoft.com/en-us/library/cc998476.aspx> (accessed on 2 December 2016).
45. The ARGoS. Available online: <http://www.argos-sim.info/publications.php> (accessed on 2 December 2016).
46. Physics Simulation. Available online: [http://bulletphysics.org/mediawiki-1.5.8/index.php/Main\\_Page](http://bulletphysics.org/mediawiki-1.5.8/index.php/Main_Page) (accessed on 2 December 2016).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).