Information Sciences

Simulating Crowd Behaviour Combining Both Microscopic and Macroscopic Rules

--Manuscript Draft--

Manuscript Number:	INS-D-21-753R3
Article Type:	Full length article
Keywords:	Crowd Simulation; pedestrians flow; Agent-based Model; Path Planning; steering behaviours; virtual environments
Corresponding Author:	Raza Saeed University of Cagliari: Universita degli Studi Di Cagliari CAGLIARI, ITALY
First Author:	Raza Saeed
Order of Authors:	Raza Saeed
Abstract:	We propose a new method for simulating pedestrian crowd movement in a virtual environment. A crowd consists of groups of different number of people with different attributes such as gender, age, position, velocity, and energy. Each group has its own intention used to generate a trajectory for each pedestrian navigating in the virtual environment. Additionally, an $agent-based$ model is introduced to simulate pedestrian behaviours in the groups, where various steering behaviours are introduced and combined into a single steering force to allow pedestrians in each group to walk toward their destination point. Based on the proposed method, every single pedestrian in each group can continuously adjust their attributes. Moreover, pedestrians optimize their path independently toward the desired goals, while avoiding obstacles and other pedestrians in the scene. The proposed method was implemented for several simulation scenarios under various conditions for a wide range of different parameters. Statistical analysis is carried out to evaluate the performance of the proposed method for simulating the crowd movement in the virtual environment. Results indicate that our method can generate each pedestrian's trajectories in each group independently to reach several goal points within a reasonable computational time.

Simulating Crowd Behaviour Combining Both Microscopic and Macroscopic Rules

R. A. Saeed[a], Diego Reforgiato Recupero[a,b], Paolo Remagnino[c]

[a]Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy
[b]R2M Solution, Via Fratelli Cuzio, 42, 27100 Pavia, Italy
[c]The Robot Vision Team, Kingston University, London, UK

Abstract

We propose a new method for simulating pedestrian crowd movement in a virtual environment. A crowd consists of groups of different number of people with different attributes such as gender, age, position, velocity, and energy. Each group has its own intention used to generate a trajectory for each pedestrian navigating in the virtual environment. Additionally, an $agent - based$ model is introduced to simulate pedestrian behaviours in the groups, where various steering behaviours are introduced and combined into a single steering force to allow pedestrians in each group to walk toward their destination point. Based on the proposed method, every single pedestrian in each group can continuously adjust their attributes. Moreover, pedestrians optimize their path independently toward the desired goals, while avoiding obstacles and other pedestrians in the scene. This method takes into account the safety-space around each pedestrian to avoid collisions among pedestrians. The proposed method was implemented for several simulation scenarios under various conditions for a wide range of different parameters. Statistical analysis is carried out to evaluate the performance of the proposed method for simulating the crowd movement in the virtual environment. Results indicate that our method can generate each pedestrian's trajectories in each group independently to reach several goal points within a reasonable computational time. Moreover, the obtained results reveal that the mean value of the computational time is not increased significantly with the increasing of the number of pedestrians in the crowd.

Email addresses: raza.saeed@unica.it (R. A. Saeed), diego.reforgiato@unica.it (Diego Reforgiato Recupero), p.remagnino@kingston.ac.uk (Paolo Remagnino)

Keywords: crowd simulation, pedestrians flow, $multi-group$ model, agent-based model, path planning, obstacle avoidance, pedestrian collision avoidance, steering behaviours, virtual environments, flocking, Boundary Node Method BNM.

1. Introduction

The study of crowd dynamics and person behaviour in crowded spaces has been the focus of research of various disciplines, such as computer graphics, virtual reality, social science, statistical physics, robotics, pedestrian and evacuation dynamics. Developed research can be grouped in two main categories: $macroscopic$ [1] and $microscopic$ [2]. $Macroscopic$ techniques focus on the aggregate behaviours of crowds. These techniques predict pedestrian motion and other characteristics such as $density$, $speed$, and emerging behaviours. However, $microscopic$ techniques have been used to find each pedestrian's path that avoids the static or dynamic obstacles and other pedestrians in the environment [3, 4]. Crowds being simulated can be $homogeneous$ or $heterogeneous$ [3, 5]. In $homogeneous$ crowds, each pedestrian has a very similar behaviour or goal. Many models have been proposed for $homogeneous$ crowd simulation. For example, the $continuum$ crowd model [6] allows a small, fixed number of goals and aggregate dynamics for dense crowd simulation [7, 3]. The $continuum$ models involve treating the crowd as a whole, and their application in large crowds composed of a hundred thousand agents [8]. Other examples include $flow-based$ models [9], governed by differential equations, which uniformly dictate the flow of crowds. In $heterogeneous$ crowds, each agent's motion in the crowd maintains a distinct, observable identity [10]. This identity is observed in the agent's goals, desired speed, cooperation, and many other factors that affect each agent [3, 5]. In the crowd simulation literature, many techniques have been proposed for simulating $heterogeneous$ crowds [11, 12]. In the $agent-based$ model, each agent's motion in the crowd is computed separately, and it is possible to simulate crowds with a different motion for different agents [3, 5]. For large-scale analysis and prediction of crowd behaviour, $continuum$ models are more efficient than the $agent-based$ models [8], where the computational time does not increase significantly with the number of agents.

However, $continuum$ models suffer from different drawbacks, such as a reduced validity range [12]. In $agent - based$ models, agents follow some predetermined rules of behaviour, which allow agents in the model to behave naturally and autonomously. This characteristic makes $agent - based$ modelling suitable for the study of agents' behaviour in complex environments [13].

In $agent - based$ models, agents (pedestrians) continuously adjust their behaviour in the simulated environment [14], and different simulation parameters can be defined for each agent [6]. These models often result in more realistic crowd movement behaviours and detailed simulations with each agent making independent decisions. However, the $agent - based$ models also have drawbacks, i.e., global path planning for each agent becomes computationally expensive, particularly in real-time contexts. Most existing path planners fail [15] to simulate agent movement using $agent - based$ models. As a result, most $agent - based$ models address only local motion planners for collision avoidance. Moreover, interactions of an agent with other agents or with the environment are often performed at a local level. Sometimes, it can result in undesirable $macroscopic$ behaviours [16]. The $boids$ algorithm [17] generates dynamics that resemble commonly observed animal motion [5] and crowds in a virtual environment [10]. Furthermore, it has been widely used in games and to generate special effects in movies [3].

An agent's motion computation is split in two distinct tasks: $global$ and $local$ navigation [3, 5]. $Global$ navigation aims to compute a long-term collision-free path towards a goal position that only the static obstacles consider. In contrast, $local$ navigation techniques take into account the motion of dynamic obstacles and other agents in the environment [3]. Examples of these methods include potential-based methods [18], field-based methods [19, 20, 16], Data-driven methods [19], Least effort crowds [21, 22, 23, 24, 25]. Despite the diversity, motion planning can be classified into two kinds of approaches: $centralized$ and $decoupled$ approaches [3]. The $centralized$ approaches [26] consider all the agents and treat the resulting system as a single composite system. In these approaches, agents' configuration spaces are combined in a composite space, and the resulting algorithm searches for a solution in this combined configuration space [3]. In contrast, the $decoupled$ planners proceed in a distributed manner, and coordination among them is often handled by exploring a coordination space, representing the parameters along each specific path, or computed some local rules. *Decoupled* approaches [27] are much faster than *centralized* approaches but may not be able to guarantee theoretical completeness. Some of these approaches have been used to generate group behaviours [28] and real-time navigation of large numbers of agents among obstacles [3, 14]. The most widely-used techniques for handling a large number of human-like agents are based on decentralized methods (see [29]). In real-world applications, crowd movement simulation provides valuable tools to planners and designers for improving efficiency and safety in public places such as airport terminals [13], shopping malls [30], train stations [29], and theatres [31]. The design of buildings can be improved if people simulations are employed to better place shops, entrances, corridors, emergency exits, and seating [31]. In new architectural or urban domains, crowd simulation is used to predict crowd flows analysis. Architectural analysis benefits from exploring the environment as quickly as possible [5]. Additionally, the crowd movement simulations have been successfully applied to various scenarios and case studies [32], i.e., crowds associated with transport systems, sporting and general spectator occasions, holy sites, political demonstrations, fire escape [8], emergency evacuation [5, 4, 31, 13], and trading port [10]. Moreover, the crowd movement simulations have been used to develop a level-of-service concept, design elements of pedestrian facilities, planning guidelines [4], safety planning [5] and support transportation planners or managers to design timetables [31]. Furthermore, crowd modelling and simulations allow the safer and more efficient design of civil structures such as big buildings, markets, and stadiums [23]. Realistic pedestrian behaviour simulation in crowded places has diverse applications in architecture design, emergency evacuation, urban planning, personnel training, education, and entertainment [14]. Additionally, realistic simulations based on pedestrian behaviours such as obstacle or other pedestrians avoidance and stress response have many applications such as computer game designers, movies, and virtual environments [5]. Considerable effort has been made on locomotion, realistic pedestrian movement behaviours, path planning, and navigation in large virtual environments [33]. The experimental studies of pedestrian movement behaviours, observations, and data recording are currently extensively available by the most sophisticated techniques. Many human walking attitudes have been pointed out [34].

Recently, simulating groups and their behaviours in a virtual environment have been studied in several scientific disciplines [35, 36]. The authors in [23] investigated the effect of groups on crowd movement, and they showed that the groups' formation has an essential effect on the crowd movement. In addition, grouping plays an important role in affecting crowd behaviour [37]. For modelling different groups' structure aspects in pedestrian crowds, the authors in [30] developed a $agent-based$ crowd simulation system. To evaluate the impact of group dynamics on the crowd behaviour, an $agent-based$ model is proposed by [13, 32]. To simulate diverse groups in crowd simulation, the authors in [2] presented a new $microscopic$ algorithm using the $agent-based$ model. The authors in [31] suggested a $multi-group\ microscopic$ model based on the interacting particle system coupled with the eikonal equation for describing groups' behaviour. Furthermore, [31] discussed a $multi-behaviours$ $microscopic$ model combining with the social force model and optimal path computation. Authors in one recent work [38] developed a new $microscopic$ model by integrating walking preferences of pedestrians into the collision-free speed model. Authors in [1] introduced a generic Smoothed Particle Hydrodynamics (SPH) framework for solving $macroscopic$ pedestrian models. Authors in [39] proposed and validated different mathematical models at the $microscopic$ and $macroscopic$ levels to study the influence of both effects. Furthermore, one recent survey [40] reviews the research developments on crowd simulation algorithms since the year 2010 focusing on the $microscopic$ or agent-based algorithms. There are several non linear modeling approaches which proved to be successful in modeling systems in various fields. In order to generate $heterogeneous$ crowd behaviour, the pedestrians' characteristics and interactions among pedestrians need to be simulated [11]. Moreover, simulating a crowd with a wide variety of pedestrians' type is essential, where the characteristics of pedestrian and surrounding environment are governor of the crowd behaviours. For example, the authors in [10] investigated crowd simulation of different groups with $heterogeneous$ behaviours. The problem of generating $heterogeneous$ crowd behaviours by adjusting the stimulation parameters is investigated by [11]. In crowd animation, the problem of directing and controlling virtual crowds addressed by [16]. Moreover, the authors is in [41] combined steering behaviours to make the virtual characters navigate their environment. To perform multiple virtual agents' path planning, the authors in [14] present a $Multi - agent$ navigation graph for each agent (characters, pedestrians, entities, etc.) in real-time. In the literature, several techniques are proposed to animate crowds [3, 29].

Despite the remarkable contributions in the literature, several essential gaps and limitations still need to be addressed. In particular:

- A mechanism to adjust and adapt the movement behaviours of each pedestrian independently in real-time is needed. Many researchers used $microscopic$ models for simulating crowd movement. However, they avoid the process of independently animating pedestrians [10].
- In a large-scale simulation with a high number of pedestrians, the path planning problem for each pedestrian can become very challenging [14]. Accordingly, crowd simulation methods with a strong computational performance to predict the crowd movement are desirable [5]. Apart from the pedestrian movement, a crowd model needs to address the dynamic interactions between pedestrians [6].
- Interaction between pedestrians can significantly influence crowd behaviour. Limited research exists that considers group dynamics. Most of the proposed methods presented in the literature treat pedestrians as individual agents and neglect group dynamics [3, 13, 31, 32, 30].
- Generally, a crowd tends to be $heterogeneous$, the requirement for having variety and $heterogeneous$ crowd is very necessary [10]. Moreover, different parts of the crowd may be behaving very differently. In this case, it is hard to represent the crowd's dynamics using a single global model [3].

The main contributions of this study can be summarized in the following points:

- We propose a method for simulated $heterogeneous$ crowd dynamics, where the crowd consists of multiple groups with a different number and various types of pedestrians with different intentions.

- Pedestrians in a crowd have attributes and follow steering behaviours. Pedestrians in each group continuously adjust their paths and update their attributes independently in real-time.
- The method we devised can simulate a large number of pedestrians in a very complex environment. Moreover, the method can be adapted to include other pedestrians' attributes in the crowd for more rich and complex simulations.
- The path planning method proposed in our previous work [42, 43] was adapted to implement pedestrian collisions.

The rest of the paper is organized as follows: Section 2 describes the problem formulation for simulating crowd movement in a virtual environment. Section 3 demonstrates the construction and configuration of the proposed method for simulating crowd movement in a virtual environment. Section 4 presents the implementation of the proposed method. Different scenarios are examined, and the results are presented and discussed. Finally, the main conclusions drawn from this study are provided in Section 5.

2. Problem Formulation

In this study, we make the assumption that a crowd of pedestrians with different group sizes and various types of pedestrians moves through the defined virtual environment for a limited amount of time. Each group of pedestrians passes through several way points, stopping at each way point for different length of time. Moreover, the number of each type of pedestrian in each group is different, with individual attributes such as gender, age, position, velocity, and energy. In the virtual environment, there are several obstacles at different locations; pedestrian must avoid such obstacles. The simulation starts with groups of pedestrians with a large variety of characteristics entering the environment from a given starting point. Once inside, they are free to navigate inside the virtual environment seeking their goals through given way points, until the simulation stops. Pedestrian can repeat their routine. In this study, pedestrians have to adjust their attributes and optimize their paths continuously in the virtual environment, while avoiding stationary obstacles and other pedestrians.

To demonstrate the groups' formulation in the crowd, let us consider a very simple scenario, as shown in Figure 1(a and b) and Table 1. For simplicity, we consider only two groups for simulating pedestrians' movement. The first group consists of "3" pedestrians, and the second group consists of "5" pedestrians. For simplifying this scenario, we assumed only "2" types of pedestrians, namely $agent1$ and $agent2$, in each group. The first group has "3" pedestrians of $agent1$ and "0" pedestrians of $agent2$, and in the second group, "2" pedestrians of $agent1$ and "3" pedestrians of $agent2$ are considered. We assume that the first group ($Group-1$) visit $goal_2$, $goal_6$, $goal_4$, and $goal_7$, and the intention of the second group ($Group-2$) is different, in which they visit $goal_8$, $goal_5$, and $goal_1$ (see Table 1). Initially, pedestrians ($Group-1$ and $Group-2$) are distributed randomly in front of the virtual environment around the starting point (see Figure 1(b)). After that, all created groups in the virtual environment need to be appropriately introduced into the working environment based on the specific schedule. In this example, groups ($Group-1$ and $Group-2$) start to move from their starting points. Then, all groups of pedestrians enter the virtual environment through the $Entrance$, and they navigate inside the virtual environment to visit several goal points ($Goals_{group}[1]$ and $Goals_{group}[2]$). Afterwards, they move toward the entrance/exit to leave the virtual environment, and they continue to move until they reach the last destination point located at the $End-point$. In this study, the number of groups, pedestrians in each group, types of pedestrians in each group, and the goal points are created randomly. The variables are illustrated in Table 1, and the given scenario graphically is illustrated in Figure 1(b).

3. Proposed Method for Crowd Simulation

In this study, pedestrians in many groups moving through the navigable area inside a virtual environment with different directions to reach their destination points. Pedestrians in a group stay together, while keeping a given distance. Pedestrians in the same group have the same goal points to visit. Each group is assumed to have a different

Table 1: Simple scenario for formulating groups of pedestrians in the virtual environment.

Variables	**Values**
Number of groups	$nGroups$=2 ($Group-1, Group-2$)
Number of pedestrians in $Group-1$	3 $pedestrians$
Number of pedestrians in $Group-2$	5 $pedestrians$
Number of pedestrian's type	$nTypes$ = 2 ($agent1_1, agent1_2$)
$nTypes$ in $Group-1$	$nPedestrians_{Types}$[1]=[3, 0]
$nTypes$ in $Group-2$	$nPedestrians_{Types}$[2]=[2, 3]
$Pedestrians_{group}$[1]	$[agent1_1, agent1_2, agent1_3]$
$Pedestrians_{group}$[2]	$[agent1_1, agent1_2, agent2_1, agent2_2, agent2_3]$
Number of the goal points for $Group-1$	$Goals_{group}$[1]= [$Start\_point$, $Entrance$, $Goal_2$, $Goal_6$, $Goal_4$, $Goal_7$, $Exit$, $End-point$]
Number of the goal points for $Group-2$	$Goals_{group}$ [2]= [$Start\_point$, $Entrance$, $Goal_8$, $Goal_5$, $Goal_1$, $Exit$, $End-point$]

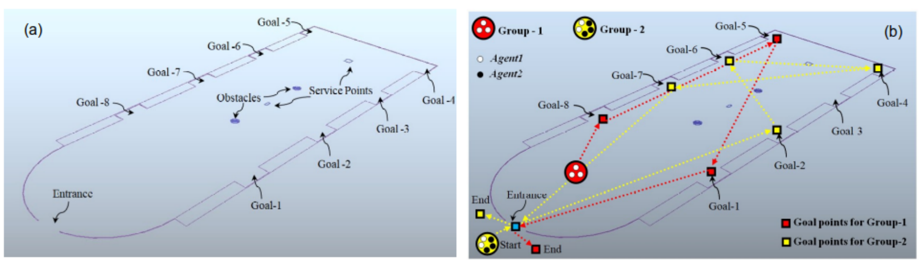

Figure 1: (a) The layout of the virtual environment with obstacles, and (b) graphical representation of the given simple scenario.

list of goal points, where each goal point corresponds to a specific region in the virtual environment. The virtual environment consists of walls, obstacles, or other areas that may not be accessible to pedestrians in addition to the goal points.

The proposed method uses the $multi-group$ model for generating a real-time trajectory for each pedestrian while navigating in the walking area. Moreover, an $agent-based$ model is introduced to model pedestrian behaviours. Several steering behaviours are introduced into the proposed method to help pedestrians to behave and interact with the other pedestrian in the virtual environment. The steering behaviours include separation, cohesion, alignment, obstacle avoidance, pedestrian collision avoidance, flocking, and goal-directed steering behaviour. Additionally, some techniques combine the steering behaviours to a single steering force that allows pedestrians to move toward their destination points independently. The combinations of steering behaviours are used to generate $heterogeneous$ crowd behaviours. The desired goal point for each group of pedestrians changes dynamically, and the desired goal point is defined at each step from the list of goal points. In the proposed method, pedestrians carry out various crowd activities, such as adjusting their attributes, avoiding collisions, optimizing their paths, and changing their behaviour. In this study, pedestrians use the local knowledge for local collision avoidance and for interacting with other pedestrians. Also, pedestrians use global knowledge for long-term planning and to provide goal directing capability. The proposed method is illustrated by the flowchart shown in Figure 2. The steps followed in the flowchart are explained in detail in the following subsections.

To provide a realistic simulation of the virtual environment and the virtual pedestrians, all elements of the simulation are imported into the $3D$ animation software package $Autodesk\,Maya$ [44] using a custom $Python$ script. Then the $Maya$'s key-frame system is used to visualize the simulated scenario. In the proposed method, we created $Python$ script models and embedded them into the $Autodesk\,Maya$ for generating real-time trajectories.

The method consists of three main steps: environment setting, motion computation, and steering behaviours, and locomotion, described in the three following subsections.

1. Environment setting: this step explains the construction of the virtual environment, obstacles, pedestrians, goal points, and the initial state of the simulated scenario (see Section 3.1).

2. Motion computation and steering behaviours: this step introduces the proposed method for simulating the crowd and computes the pedestrians' movement. Several steering behaviours are described, and several techniques are used for combining steering behaviours to a single steering force to direct pedestrians toward their destination points. Furthermore, this step explains the obstacles and pedestrians' collision avoidance (see Section 3.2).

3. locomotion: this step concentrates on the virtual pedestrian's animation in the virtual environment. The locomotion converts the steering force to a control signal motion, which guides the pedestrians to move in the virtual environment toward their destination point (see Section 3.3).

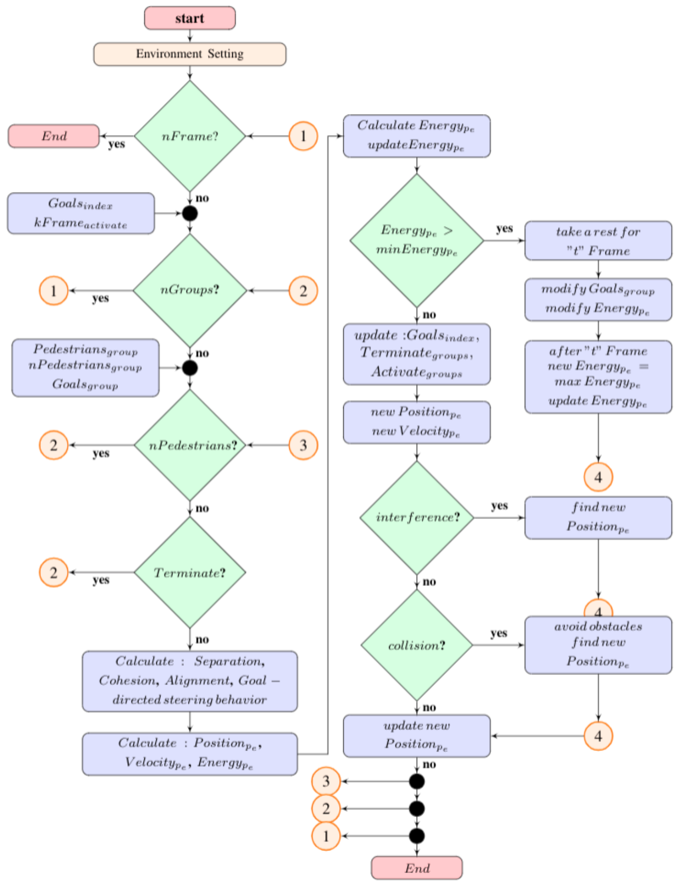

Figure 2: Flowchart illustrating the steps of the new proposed method for crowd simulation.

3.1. Environment Setting

The first step of the proposed method (see Figure 2) introduces the working environment, obstacles, pedestrians, and goal points. Moreover, this step identifies the initial state of the pedestrian's attributes: gender, age, position, velocity, and energy. Also, the interactions between pedestrians with the environment have to be described. The detail of all these steps is provided in the following subsections.

3.1.1. Virtual environment, pedestrians, obstacles and goal points

Figure 1(a) illustrates the used virtual environment with an entrance and a number of goals and obstacles inside the scene. A more detailed description of how the environment can be created can be found in [43]. In this study, obstacles are distributed at different locations in the free space. Pedestrians keep a certain distance from the obstacles. A safety zone is created around the obstacles to avoid the possibility of overlapping the paths traced by pedestrians with the obstacle boundaries. A circle represents a safety zone that pedestrians can not enter while they move. The safety zone radius is constant, R, where the value of R depends on the size of obstacles. Pedestrians are then instantiated in the virtual environment. In the simulated environment, each group is assumed to have different goal points to visit, and the goal points ($nGoals$) corresponding to a specific region, as shown in Figure 1(a).

3.1.2. The initial state of the pedestrian

Pedestrians $Pedestrians_{crowd}$ in each created group $Pedestrians_{group}$ are distributed randomly outside the working environment, with a random velocity $Velocity_{p_e}$. Then they start to move to reach the goal point in the list of the goal points ($Goals_{group}$).

The entrance of pedestrians and groups ($Pedestrians_{crowd}[g_r]$, $g_r = 1, ..., nGroups$) is staggered. Groups in the crowd will decide whether and when (key-frame $kFrame$) they start to move from the starting point. Based on their decision, each group g_r, ($g_r = 1, ..., nGroups$) is activated to move at a randomly created key-frame ($kFrame_{activate}[g_r]$) between (1, $nFrames$*c_{a1}), where $nFrames$ represent the total number of frames and c_{a1} is a constant. In the first key-frame, the value of $Activate_{groups}[g_r]$ is set to *"off"* for each group (g_r), this value will switch to "on" when the key-frame reaches $kFrame_{activate}[g_r]$, then the group $Pedestrians_{group}$ starts to move. In the same fashion, at the first key-frame, the value of $Terminate_{groups}[g_r]$ is set to *"off"* for each group (g_r). The value of $Terminate_{groups}[g_r]$ will switch to "on“ when the group $Pedestrians_{group}$ reaches its final destination $Goals_{group}[last]$ outside the simulated environment.

The groups' initial configurations in terms of the number of the pedestrian, type of pedestrian, number of goal points and their sequence, pedestrian's attributes, and most of the other variables are generated randomly. Additionally, it is essential to simulate each pedestrian independently and introduce interactions between pedestrians and the simulated environment. In the proposed method, a set of functions is created to get and set the values of a pedestrian's attributes in the scene in real-time.

3.2. Motion Computation and Steering Behaviours

This section presents the procedure to compute each pedestrian's movement in the simulated environment. It directs the groups through the walking area until they terminate the simulation. This section also describes various types of steering behaviours that help pedestrians to move in the virtual environment. The pedestrians' behaviours include activities such as flocking with neighbouring pedestrians, goal-directing, obstacle avoidance, and pedestrian collision avoidance. The pedestrians' initial state, goals, obstacles, and interaction between pedestrians with an environment of the simulated crowd are defined previously. At each key-frame ($keyframe$) pedestrians are moving forward from the current position to the new updated position ($Position_{p_e}$) with the walking velocity $Velocity_{p_e}$. The new updated $Position_{p_e}$ and $Velocity_{p_e}$ is calculated based on the steering behaviours with the current pedestrian's position and velocity. Algorithm 1 explains pedestrians' movement in the virtual environment from the starting point to the ending point (see Figure 2), and this process is illustrated in the following steps.

Step 1: First, this algorithm starts with the previously created input data: number of groups $nGroups$, groups' activate list $Activate_{groups}$, groups' terminate list $Terminate_{groups}$, pedestrians in the crowd $Pedestrians_{crowd}$, number of the pedestrians in the crowd $nPedestrians_{crowd}$, list of the goal points for each group in the crowd $Goals_{crowd}$, list of the pedestrian's Position in the crowd $Positions_{crowd}$, and number of frames $nFrames$. Algorithm 1-Line(1)

Step 2: At each key-frame ($keyframe = 1, ..., nFrames$), the activate group's value $Activate_{groups}[g_r]$ set to "on" when $keyframe$ reach to $kFrame_{activate}[g_r]$ (see Algorithm 1-Line(3)). Then, for each activate group we will define:

1. Index of the goal point $Goals_{crowd}[g_r]$ (goal index, $Goal_i$). Algorithm 1-Line(7)
2. Pedestrians in the group $Pedestrians_{crowd}[g_r]$. Algorithm 1-Line(9)

3. Number of pedestrians in the group $nPedestrians_{crowd}[g_r]$. Algorithm 1-Line(10)

4. The goal points $Goals_{crowd}[g_r]$ for the group $nPedestrians_{crowd}[g_r]$. Algorithm 1-Line(11)

Step 3: For each pedestrian ($p_e = 1, .., nPedestrians_{group}[g_r]$), the current pedestrian's position $(Position_{p_e})$ is obtained from the scene in real-time, and the value of pedestrian's velocity $Velocity_{p_e}$ and pedestrian's energy level $Energy_{p_e}$ is obtained from the attributes of pedestrian $Pedestrians_{group}[p_e]$. Algorithm 1-Line(14)

Step 4: At each key-frame ($keyframe = 1, ..., nFrames$), the new $Velocity_{p_e}$ is calculated for each pedestrian $Pedestrians_{group}[p_e]$ in activated groups. In this study, pedestrian' velocity is limited to $mSpeed$ ($unite/timestep$), and the constant c_{s1} influence the speed of the pedestrians in the scene. Algorithm 1-Line(23)

Step 5: For each pedestrian $Pedestrians_{group}[p_e]$ in activated group $Pedestrians_{crowd}[g_r]$, a new position $Position_{p_e}$ inside the simulated environment is computed based on the steering behaviors that we will explain in the following subsections. Algorithm 1-Lines(23)

Step 6: During this simulation, pedestrians need to keep a certain distance $(2 \times r)$ between themselves. Therefore, the new updated $Position_{p_e}$ of each pedestrian has to check for collision with neighbouring pedestrians, as illustrated in Figure 4. Algorithm 1-Lines(24)

Step 7: The proposed method for crowd simulation uses the BNM method to check the new updated $Position_{p_e}$ for collision with obstacles. Suppose a collision happens between pedestrians and obstacles. In that case, the BNM method computes another new position $Position_{p_e}$ for the pedestrian to avoid collision with obstacles. Algorithm 1-Lines(26)

Step 8: At each key-frame ($keyframe = 1, ..., nFrames$) the energy level of every single pedestrian is decrease as they walk through the environment as follow: $Energy_{p_e} = Energy_{p_e}$ - c_{s1} (see Algorithm 1-Line(15)), where the constant c_{s1} represent the reduced value at each $keyframe$. Suppose the value of $Energy_{p_e}$ reaches below the minimum $Energy$ level. In that case, all pedestrians in the group will visit the service point in the walking area after finishing the current task (see Algorithm 1-lines(29-30)). Meanwhile, the group stays in the service point, their energy will increase again, and they stay in the service point until their energy reaches the maximum level. Then they will start again to move to visit the rest of the goal points. Algorithm 1-lines(31-36).

Step 9: At each key-frame, if the distance between any pedestrian $Position_{p_e}$ in the group $Pedestrians_{group}$ and $Goals_{group}[Goal_i]$ is reached below c_{s2}, it indicates that this group is arrived to the current goal point. Therefore, this group has to get a new goal point from the goal points list and update the goal's $index$ ($Goals_{index}$). Algorithm 1-lines(17-22)

Step 10: At each key-frame, pedestrian changes its position to the new position at $Position_{p_e}$ in real time, afterward, pedestrians update their attributes include $Velocity_{p_e}$ and $energy_{p_e}$ in $Pedestrians_{group}[p_e]$ with the new calculated attributes. Algorithm 1-lines(38,39))

Step 11: When the group of pedestrians $Pedestrians_{crowd}[g_r]$ reach to last goal point $Goals_{group}[last]$, the value of $Terminate_{groups}[g_r]$ is switch to "on" and also the value $Activate_{groups}[g_r]$ is switch to "off" as well. Algorithm 1-Line(19-20)

In addition to the classic steering behaviours (see [41]), goal-directed behaviour was implemented to control pedestrian movements and direct them to the goal points. The obstacle avoidance for our simulation is based on an extension of our Boundary Node Method (BNM) [45, 42, 43]. Using the global path planning in real-time contexts is difficult for modelling pedestrians' behaviours by using the $agent-based$ model because it becomes computationally expensive. Therefore, the $agent-based$ models used in this study separate local collision avoidance from global path plan-

Algorithm 1 Motion computation and steering behaviors

Inputs:
$nFrames,\ nGroups,\ Activate_{groups},\ Terminate_{groups},\ Pedestrians_{crowd},\ nPedestrians_{crowd},\ Goals_{crowd},\ Positions_{crowd},\ c_{s1}$
for $keyframe$ = 1 to $nFrames$ **do**
  if $keyframe = kFrame_{activate}$ **then** $Activate_{groups}[g_r] \leftarrow "on"$
  end if
  $p_c \leftarrow 0$, $Goals_{index}([1] * nGroups)$
  for g_r = 1 to $nGroups$ **do**
    $p_c \leftarrow p_c + 1$, $Goal_i \leftarrow Goals_{index}[g_r]$
    if ($Terminate_{groups}[g_r] = "off"$ and $Activate_{groups}[g_r] = "on"$) **then**
      $Pedestrians_{group} \leftarrow Pedestrians_{crowd}[g_r]$
      $nPedestrians_{group} \leftarrow nPedestrians_{crowd}[g_r]$
      $Goals_{group} \leftarrow Goals_{crowd}[g_r]$
      for p_e = 1 to $nPedestrians_{group}$ **do**
        $Compute\ Separation,\ Cohesion,\ Alignment$
        $(Position_{p_e},\ Velocity_{p_e},\ Energy_{p_e}) \leftarrow Pedestrians_{group}[p_e]$
        $Energy_{p_e} \leftarrow Energy_{p_e} - c_{s1}$
        $Pedestrians_{group}[p_e] \leftarrow Energy_{p_e}$
        if $Energy_{p_e} > minimum\, Energy_{p_e}$ **then**
          if $dist((Position_{p_e}, Goals_{group}[Goal_i]) < c_{s2}$ **then**
            $Goal_i \leftarrow Goal_i + 1, update\, Goals_{index}[g_r] \leftarrow Goal_i$
            if $goal_i = len(Goals_{group})$ **then**
              $Terminate_{groups}[g_r] \leftarrow "off"$
            end if
          end if
          $compute\, new\, Velocity_{p_e}\, and\, Position_{p_e}$
          if $ind\, collision = "True"$ **then** $Position_{p_e} \leftarrow Algorithm$ 2
          end if
          if $Obstacle\, collision = "True"$ **then** $Position_{p_e} \leftarrow BNM$
          end if
        else
          $stay\, in\, service\, point,\, modify\, Goals_{group}\, and\, Energy_{p_e}$
          $update\, Pedestrians_{group}[p_e]\, with\, new\, Energy_{p_e}$
          if $staying\, time\, is\, finished$ **then**
            for p_e = 1 to $nPedestrians_{group}$ **do**
              $new\, Energy_{p_e} \leftarrow maximum\, Energy_{p_e}$
              $update\, Pedestrians_{group}[p_e] \leftarrow new\, Energy_{p_e}$
            end for
          end if
        end if
        $update\, Pedestrians_{group}[p_e] \leftarrow new\, (Position_{p_e}, Velocity_{p_e})$
        $Positions_{crowd}[p_c] = Position_{p_e}, update\, position_{p_e}$
      end for
    end if
  end for
end for

ning. Based on the extended BNM method, each pedestrian in the crowd is simulated by a nine-node quadrilateral element (see Figure 3a). The nodes are denoted by $p(q), (q = 1...9)$, the centroid node $p(5)$ represents the pedestrian's location. As illustrated in Figure 3a and c, the nodes $p(1 \rightarrow 4)$ with $p(6 \rightarrow 9)$ represent the eight-boundary nodes that help pedestrians move forward and avoid obstacles. As shown in Figure 3b, pedestrian and boundary nodes $p(q), (q = 1...9)$ are restricted to move in eight-possible directions $e(u), (u = 1...8)$ in the workspace.

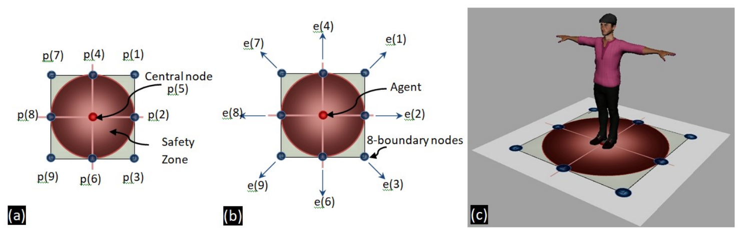

Figure 3: A nine-node quadrilateral element with a safety zone (a) along with its motion directions (b) and simulated pedestrian in a virtual environment (c).

3.2.1. Obstacle Avoidance

Using the extended BNM, a pedestrian and boundary nodes implement steering to avoid obstacles and changing their motion direction by selecting a new position in the free space C_{free}. The obstacles avoidance method considers all obstacles in the walking area at each key-frame when a pedestrian moves near the safety zone around obstacles. If the $2D$ distance between the pedestrian's position and centre of the obstacle is less than (r+R), then the pedestrian interferes with the safety zone around obstacles, where R and r represent the radius of the safety zone around the obstacle and pedestrian, respectively. Based on the BNM method, the pedestrian has to change the motion direction to avoid collision with the obstacle.

3.2.2. Pedestrian Collision Avoidance

Each moving pedestrian represents a dynamic obstacle for all other pedestrians in the scene and influence the pedestrian's motion in the crowd. Pedestrians should not reside within the personal space requirement, the so-called private sphere, of the other pedestrians. Usually, each pedestrian has a safety zone with a $radius, r$ (see

Figure 4(c)), pedestrians should keep a certain distance from the other neighbouring pedestrians located too close.

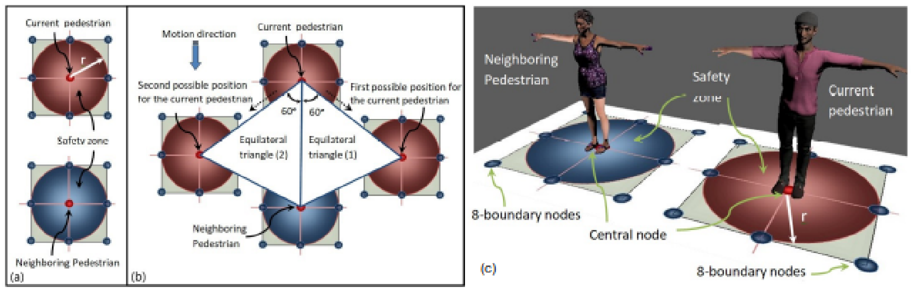

Figure 4: pedestrian collision avoidance (a and b) with personal space requirements (c).

For instance, as the current pedestrian tends to move forward, it should keep a certain distance from other neighbouring pedestrians in the walking area, as demonstrated in Figure 4a. This study assumes that the distance between pedestrians should not be less than ($2 \times r$). Suppose that the distance between current and neighbouring pedestrian drops below a specific value ($2 \times r$). In that case, the current pedestrian changes its motion direction slightly to prevent the collision. Pedestrians usually choose the fastest route to reach their next destination, but not the shortest one [4]. The Principle of Minimum Energy to handle such situations can also be used [37]. In general, pedestrians consider detours and the comfort of walking, thereby minimizing their efforts [4]. We have proposed an appropriate reactive behaviour, called $pedestrian\, collision\, avoidance$, for preventing collision between pedestrians walking closer as illustrated in Figure 4b. Suppose that the current and neighbouring pedestrian move closer, and the distance between them drops below $2 \times r$. In that case, the current pedestrian steers to either left or right for a new calculated position $Position_{p_e}$ which is located on the third vertex of an equilateral triangle (see Figure 4b). The overall process of pedestrian collision avoidance is illustrated in Algorithm 2.

3.2.3. Flocking and Goal-directed Behaviour

The flocking behaviour follows the methods in [41, 17]. Goal-directed behaviour models pedestrians' motivation in each group walking towards a well-defined goal at a certain point in the virtual environment. The computation of the goal-directed steering

Algorithm 2 Pedestrian collision avoidance

Inputs:
  $Positions_{crowd}$,
  $newPosition_{p_e}$,$currentPosition_{p_e}$,$nPedestrians_{total}$
for p_e = 1 to $nPedestrians_{total}$ **do**
  find $distance\, between\, new\, Position_{p_e}\, and\, Positions_{crowd}[p_e]$
  if $distance < (2 \times r)$ **then**
    $\Delta x = newPosition_{p_e}[0] - Positions_{crowd}[p_e][0]$
    $\Delta z = newPosition_{p_e}[2] - Positions_{crowd}[p_e][2]$
    if $random.random() < 0.5$ **then** $rotate_{angle} \leftarrow 60.0$
    else $rotate_{angle} \leftarrow -60.0$
    end if
    $\alpha \leftarrow rotate_{angle}/180 * math.pi$
    $new_x \leftarrow newPosition_{p_e}[0] + math.cos(\alpha) * \Delta x + math.sin(\alpha) * \Delta x$
    $new_y \leftarrow 0$
    $new_z \leftarrow newPosition_{p_e}[2] + math.sin(-\alpha) * \Delta x + math.cos(\alpha) * \Delta z$
    $newPosition_{p_e} \leftarrow [new_x, new_y, new_z]$
    find $distance\, between\, new\, Position_{p_e}\, and\, Positions_{crowd}[p_e]$
  else if $distance = 0$ **then**
    $newPosition_{p_e} \leftarrow currentPosition_{p_e}$
  end if
end for

behaviour ($goal_{vector}$) is explained in the Algorithm 3. As shown in this algorithm, the distance between the current position of the pedestrian ($current\, Position_{p_e}$) and the current active goal ($current\, Goals_{group}$) is calculated. Subject to the distance between these two points being less than a certain value (C_{g1}), then a new goal point is assigned for the current group. Suppose the current goal point $Goals_{group}$ is the last destination point. In that case, the group will terminate the simulation, and this group is no longer active. Otherwise, the $goal\_directed$ steering behavior calculates the $goal_{vector}$ and normalizing the $goal_{vector}$ by applying a weighting $goal_{weight}$, as illustrated in Algorithm 3.

3.2.4. Combining Behaviours

Combining behaviours can happen in two ways: (1) switching and (2) blending [41]. When circumstances change in the simulated environment, a pedestrian may "switch" between behavioural modes. Alternatively, these behaviours, unfolding simultaneously, are commonly "blended" together. For example, in normal situations, as

Algorithm 3 Goal-directed behavior

Inputs:
$current\,Goals_{group}, current\,Position_{p_e}(p_e \leftarrow index), goal_{weight}, C_{g1}$
$Goal_i \leftarrow Goals_{index}[g_r]$
if $(distance\,between\,current\,Position_{p_e}\,and\,current\,Goals_{group}) < C_{g1}$ **then**
  $Goals_{index}[g_r] \leftarrow Goals_{index}[g_r] + 1$
  if $Goals_{index}[g_r] \leftarrow length(Goals_{group})$ **then**
    $terminate[g_r] \leftarrow "on", active_{group}[g_r] \leftarrow "off"$
  end if
end if
$goal_{vector} = \leftarrow subtracting(current\,Goals_{group}, current\,Position_{p_e})$
$goal_{vector} \leftarrow normalizing(goal_{vector}, goal_{weight})$
....
$Combining\,with\,other\,behaviors,\ if\,there\,are$
....

the pedestrians move toward their destination, flocking and goal-directed behaviour are blended into a single steering force vector to allow the group to walk toward their goal. Pedestrians in the group always must keep on moving with the group toward their destination, and they cannot afford to ignore either component behaviour. Furthermore, suppose that a group of pedestrians move in a given environment, and they approach an obstacle or other pedestrians in the same or different groups. This situation leads to a behavioural switch from moving to collision avoidance. All pedestrians then try to avoid collisions with obstacles and pedestrians, and pedestrians might split from the group while avoiding obstacles. As previously discussed, multiple steering behaviours guarantee the shortest collision-free paths for pedestrians, and pedestrians never get stuck behind obstacles. The proposed method uses $Python$ scripted models to evaluate pedestrians' behaviours and computed a new state of pedestrians. Scripted models are also used to generate $3D$ scenes and viewed in the animation software package $Autodesk\ Maya$ [44], as illustrated in the following subsection.

3.3. Realistic rendering of the 3D animation

$Maya$'s key-frame is used to create a $3D$ animation. We used seven different categories of pedestrian, we then imported into $Maya$ using $Python$ scripted models. Subsequently, $Maya$ allows the scripted models to control the pedestrians. Based on the proposed method, the position ($Position_{p_e}$) and the velocity ($Velocity_{p_e}$) of each virtual pedestrian in the scene are updated in real-time.

4. Crowd Simulation

The used virtual environment is shown in Figure 1(a). The proposed method, described in Section 3, is used to generate trajectories for groups of pedestrians. Two different scenarios were implemented. The first scenario has four groups of pedestrians (see Subsections 4.1). In the second scenario, two examples of the real-life crowd situation scenario are considered (see Subsections 4.2 and 4.3). In this study, the proposed methods are implemented in $Python$, and "$example\,1$" was carried out on the laptop with Intel(R) core(TM) i5-8300H CPU 2.30GHz, 8RAM, and for "$example\,2$" we used PC with Intel(R) core(TM) i5-5200U, CPU 2.20GHz $\times$ 4, 7.5GB RAM. Moreover, the effect of the crowd's size on the computational time is investigated for different simulations with various configurations, and statistical analysis for the obtained data was carried out (see Subsections 4.4). Additionally, the performance of the proposed method for simulating the crowd movement is compared with the related techniques (see Subsections 4.5). In this simulation, pedestrians of different groups' sizes entering the virtual environment through the entrance (see Figures 1(a) and 16.

4.1. Simulation of a Simple Scenario

In the simple crowd simulation scenario, we consider only four groups of pedestrians ($nGroups$ ="4"). Each group contains a different number of pedestrians with "7" different types ($nTypes$="7"). The number of each type of pedestrian in each group is generated randomly between $[0, N_c]$, it is assumed that $N_c = 1$. Pedestrian's type and group size distributions illustrated in Table 2. As shown in the table, the total number of pedestrians in each group is determined simply by summation of all type of pedestrians in the same group $nPedestrians_{group}$, which are $[4, 4, 5, 3]$. Therefore, the total number of pedestrians in the crowd is the sum of all pedestrians in the scene, which is equal to "16" pedestrians ($nPedestrians_{total}$="16").

In the simulated scenario, the the entrance's width is "15" $units$, and the width w and the length l of the simulated environment are set to "105" and "330" $unit$, respectively. Inside the simulated environment, there are "8" regions ($goal\,points$) that

Table 2: Pedestrians' distribution with different group size: presents the number of groups in the simulation ($nGroups$ ="4"), number of pedestrians of each type in the same group ($nPedestrians_{Types}$), number of pedestrians in each group ($nPedestrians_{group} = [4, 4, 5, 3]$), and the total number of pedestrians (NOP) in the crowd ($nPedestrians_{total}$ ="16")

Groups	***Type*** $_1$	***Type*** $_2$	***Type*** $_3$	***Type*** $_4$	***Type*** $_5$	***Type*** $_6$	***Type*** $_7$	NOP
$group_1$	1	0	1	0	1	0	1	4
$group_2$	1	0	1	1	0	1	0	4
$group_3$	1	1	1	1	1	0	0	5
$group_4$	0	1	0	1	0	1	0	3
$total$	3	2	3	3	2	2	1	16

the groups of pedestrians want to visit ("4" goal points on both the right and left-hand side). Each goal point is corresponding to a specific region in the simulated environment. Moreover, there are "2" general service points. These points are considered a temporary goal point. We add them to the goal points when the group needs them (therefore, $nGoals$="10"). The x and $z - coordinates$ of the goal points are set to [37.5, 37.5, 37.5, -37.5, -37.5, -37.5, -37.5, 37.5] and [69, 10.5, -51, 10.5, -51, 69, -109.5, -109.5], respectively, and the services points are located at (0,0,0) and (0,0,-135). Additionally, the values of the parameters related to the steering behaviours such as [$cohesion\, radius$, $separation\, radius$, $alignment\, radius$, $cohesion\, weight$, $separation\, weight$, $alignment\, weight$] are set to [6, e, 1.8, 0.75, 0.55, 0.02].

During the simulation, each pedestrian is considered as a dynamic obstacle for other pedestrians. Every single pedestrian has a personal space requirement with a $radius, r$, where other pedestrians should not share the space. In this simulation, the safe space's radius around pedestrians r is set to 0.9 $units$. The minimum distance between pedestrians is set to $2 \times r$ $units$, such that pedestrians do not interfere with the other neighbouring pedestrians. We assume that there are four static obstacles ($n3DObstacles$="4") located $[(-20, 0, 80), (20, 0, 120), (-20, 0, 120)$, and $(20, 0, 80)]$ and a circular safety zone is created around each obstacle with a radius of $4\, units$. Moreover, there are other two static obstacles located at $[(-30, 0, -35)$, $(30, 0, -35)]$, where a circular safety zone is created around these obstacles as well with a radius of $7\, units$. While pedestrians move close to the obstacles and other pedestrians, they should keep a certain space for safety. The detail of the simulation results is presented in the following subsections.

In this scenario, pedestrians' attributes and their movement are investigated at each frame $keyframe$, where the maximum number of key-frames ($nFrames$) set to "500". Based on the proposed method, initially groups of pedestrians g_r, ($g_r = 1, ..., nGroups$, $nGroups$ ="4") with different types ($nTypes$="7") created in the front of the virtual environment around the starting point. The initial position of each pedestrian $Position_{p_e}$ ($p_e = 1, .., nPedestrians_{group}[g_r]$) in each group g_r is generated randomly. Afterward, the initial walking velocity for each pedestrian $Velocity_{p_e}$ ($p_e = 1, .., nPndividuals_{group}[g_r]$) generated randomly between $(0, c_{f3})$. The constant c_{f3} represents the maximum walking speed ($mSpeed$), and it has been limited to "2" $units/timestep$. In the simulated environment, groups of pedestrians move in different directions to reach their desired goal points while avoiding obstacles and other pedestrians in the scene.

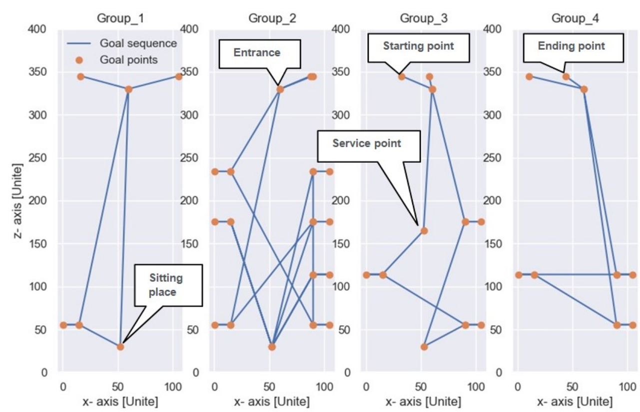

Figure 5: Goal points for each group of pedestrians: each *goal point* represents by two-points, the first point is the entrance of the goal, and the second point represents the inside of the goal area.

A list of goal points has been created for each group, as shown in Figure 5. Both starting and ending points are created randomly and added to the list of the goal points. The value of the $x - coordinate$ for the starting and ending points are generated randomly between [0, w]. In addition, the value of $z - coordinate$ for the starting and ending points are fixed to $l + c_{s1}$, it is assumed that the constant c_{s1} is set to "15". Moreover, the goal point at the centre of the scene (w/2,l/2) represents the first service point. The x and $z - coordinates$ of the entrance are set to $w/2 + 7.5$ and l, respectively. As shown in Figure 5, the origin point of the virtual environment transformed from the centre of the scene to the bottom left corner. The proposed method updates the current group's goal point if the distance between the current pedestrian and the current goal point drops below a certain value C_{g1}. It is assumed that the value of C_{g1} is set to $1.5units$.

In this simulation, each group g_r, ($g_r = 1, ..., nGroups, nGroups =$"4") starts to move to enter the simulated environment at a ($kFrame_{activate}[g_r]$) randomly generated key-frame between (1, $nFrames$*c_{a1}), where $nFrames$ represent the maximum number of frames and the value of constant c_{a1} is set to "0.5". The groups $1,\ 2,\ 3\ and\ 4$ are activated at $keyframes\ 41,\ 137,\ 181,\ and\ 203$, respectively. The red circles in the graph represent the groups' size in terms of the number of pedestrians, and the bigger circle represents the group with a higher number of pedestrians. Afterward, at each key-frame ($keyframe = 1, ..., nFrames$, $nFrames =$"500"), pedestrians' position and velocity are computed by using the proposed method. The simulation results of the pedestrians' positions and velocity for all groups are shown in Figures 6 and 7.

The achieved results shown in Figure 6 represent pedestrians' position of each group after the simulation runs for the "500" key-frame. The red circle objects represent the pedestrians' positions. From the obtained results shown in Figure 6, it is observed that the proposed method can create the trajectories for each pedestrian to move from the starting point to the final destination point in the virtual environment. This study considers the safety-zone around obstacles to avoid the possibility of overlapping the trajectories traced by pedestrians with obstacle boundaries. The proposed method provides a collision-free path for pedestrians to reach their goal points (see Figure 5). Each new position is allocated after the current pedestrians' position, and pedestrians start to change their direction as they move closer to obstacles or other pedestrians in the scene.

Figure 7 presents the simulation result of the calculated pedestrians' velocity based on the proposed method. The left-hand side of the graph represents the pedestrians' velocity in x and $z-directions$, and the right-hand side represents the absolute velocity of all pedestrians in the scene. We obtained "4235" values of pedestrians' velocity from

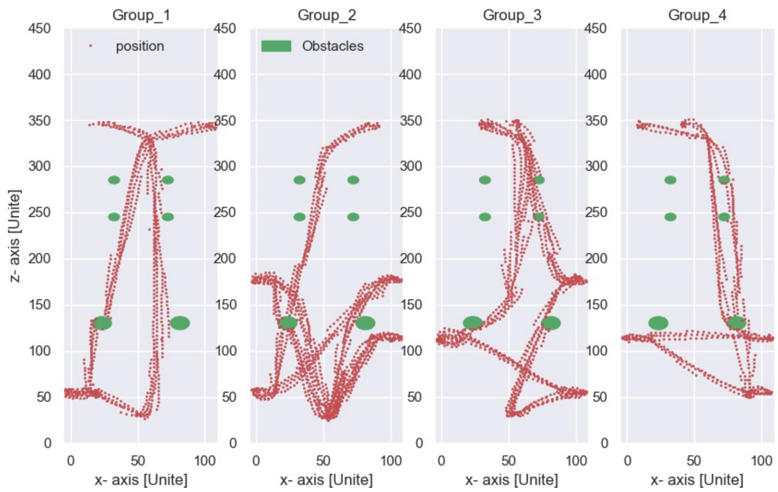

Figure 6: Pedestrian' trajectory: showing the pedestrians' movement towards their goal points in a simulated environment.

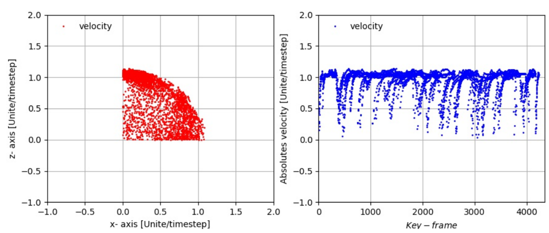

Figure 7: Pedestrian' velocity: change the walking velocity of all pedestrians.

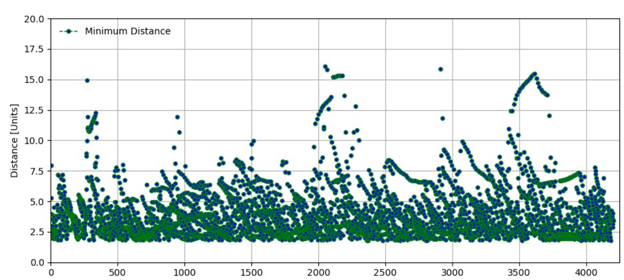

Figure 8: The minimum distance between pedestrians.

the simulation results, where the maximum value of the pedestrian's velocity reaches 1.139 $units/timestep$. The mean and the standard deviation of the calculated velocity are equal to 0.929 $units/timestep$ and 0.205, respectively. It can be observed from the simulation results that the pedestrian's velocity decrease as the pedestrians come closer to the goal points. As shown in Figure 7, the minimum velocity is achieved near the goal points.

Each pedestrian has a personal space requirement with a $radius, r$ (see Figure 4(c)) to avoid pedestrians' collisions. The radius of the personal space r set to $0.9\,unit$. Pedestrians should not interfere with the space of the other neighbouring pedestrians. The minimum distance between pedestrians is set to $2 \times r$ $units$. At each key-frame, the minimum distance between the current pedestrian and all other pedestrians is calculated, and the obtained results are presented in Figure 8. The mean and the standard deviation of the calculated minimum distance are equal to 3.915 $units$ and 2.038, respectively. The obtained results showed that the minimum distances do not fall below 1.8017 $units$. The study of group behaviours can help to better understanding and respecting different cultures' personal space. Even identify the minimum personal space requirements needed to protect health and limit the spread of the disease.

In this simulation, each group consists of several pedestrians' type, where each type of pedestrian has a different energy level $energy_{p_e}$, as illustrated in Table 3. Based on the proposed method, the pedestrians' energy level is calculated, and the obtained results are presented in Figure 9. As illustrated in the figure, each raw represents the change in the energy level of each pedestrian. For example, the first group consist of 4 type of pedestrians $Type_1$, $Type_3$, $Type_5$, and $Type_7$ (see Table 2). Each pedestrian in this group has the initial level of energy $energy_{p_e}$ (maximum $energy_{p_e}$ level) as starts to move to enter the simulated environment at randomly created key-frame ($keyframe$ 41). The maximum energy level for the $Type_1$, $Type_3$, $Type_5$, and $Type_7$ are equal to $8 * C_e$, $12 * C_e$, $16 * C_e$, and $14 * C_e$, respectively (see Table 3), the value of C_e is set to 25 in this simulation. As the pedestrians walk through the environment toward their goal points, the energy level will decrease for all pedestrians, as shown in Figure 9. The energy level of $Type_1$ will reach below the minimum $energy_{p_e}$ level first (see Figure 9) because $Type_1$ has a lower energy level than other pedestrians in the group. In this situation, all pedestrians in the group will visit the service point in the walking area after finishing the current task (see Figures 5 and 6). Meanwhile, the pedestrian stays in the service point; their energy will increase again. They are staying in the service point until their energy reaches the maximum level, and then they will start again to move to visit the rest of the goal points.

Table 3: Maximum max_e and minimum min_e energy level for each type of pedestrian

Types	$Type_1$	$Type_2$	$Type_3$	$Type_4$	$Type_5$	$Type_6$	$Type_7$
max_e	$8 * C_e$	$10 * C_e$	$12 * C_e$	$20 * C_e$	$16 * C_e$	$18 * C_e$	$14 * C_e$
min_e	$4 * C_e$	$5 * C_e$	$6 * C_e$	$10 * C_e$	$8 * C_e$	$9 * C_e$	$7 * C_e$

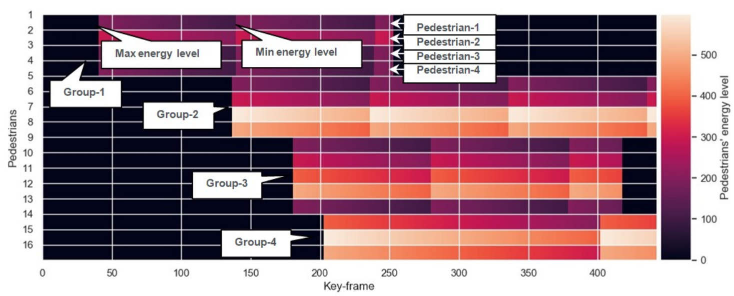

Figure 9: Pedestrian' energy level: change of the pedestrians' energy level at each key-frame.

The next simulated scenario is a public spaces such as a shopping mall. The simulation results with a higher number of small groups are discussed and presented in the following subsection.

4.2. Simulation of Real-Life Crowd Movements: example 1

This section investigates the implementation of the proposed method for simulating pedestrian crowd movement in a large and complex virtual environment of public spaces such as a shopping mall for a limited period of time. To demonstrate a realistic pedestrian movement through a virtual environment, we consider different groups with various types of pedestrians ($family$, $friends$, etc.) in the crowd. Whereas each type of pedestrian has its own attributes such as $gender$, age, $position$, $velocity$, and $energy$. In this scenario, many groups of pedestrians are appropriately introduced into the shopping mall. Each group's intention is different for visiting the number of

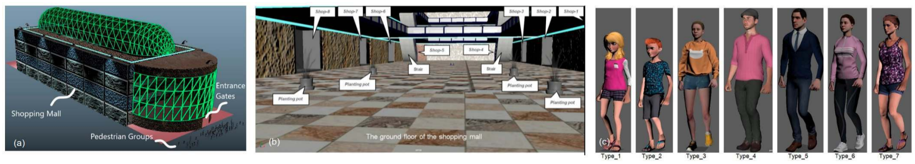

Figure 10: (a) A large-scale $3D$ model of the commercial mall populated by virtual groups of pedestrians [46]. (b) Screen-shot of the shopping mall with obstacles: shows the walking area on the ground floor of the shopping mall with the $3D$ obstacles. (c) Shows different $3D$ characters animation obtained from $Mixamo$ [47]

shops in the walking area with different visiting sequences. At each step in the simulation, virtual pedestrians adjust their attributes and optimize their paths independently in real-time. Moreover, they avoid stationary obstacles and other pedestrians in the virtual environment when they move closer. In this study, a multi-level shopping mall environment is considered, as shown in Figure 10 (a). For simplicity, we assume that the pedestrians are moving within a walking area on the ground floor of the shopping mall, as shown in Figure 10 (b). The other floor of the mall is not accessible by pedestrians. Pedestrians' activities are categorized into walking through the large environment model and shopping activities. The shopping mall consists of several shops (see Figure 10 (b)), and the shops are located on both sides of the shopping area to emulate the real-world scenario.

In this scenario, pedestrians formulated in groups before entering the shopping mall. Each group consists of multiple pedestrian types ($male$ and $female$) of different ages (old, $young$, and $child$) to establish the range of personality variation. Several examples of virtual pedestrians [47] that has been used in this simulation are shown in Figure 10 (c). Each pedestrian has a different energy level, i.e., an old pedestrian has a lower energy level than a $young$ pedestrian. The energy level will decrease as they walk through the environment. Pedestrians rest at the sitting place in a service point when their energy ($energy_{p_e}$) reaches below the minimum energy level. Furthermore, during the visit, sometimes pedestrian considers undertaking other activities, and it can happen between two sequential visits. Examples of activities in the proposed model include having food and using other mall services. In the simulated scenario, pedestrians in the same group have the same goal points ($shops$) to visit, and each group is assumed

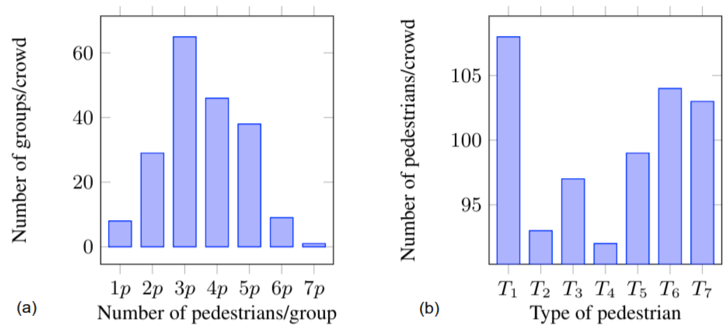

Figure 11: Distribution of pedestrians in the scene, where pedestrian is denoted by p and type of pedestrian is denoted by T. (a) Group size distributions in the crowd, and (b) Pedestrians type contribution in the crowd.

to have a different list of goal points. Each goal point ($nGoals$) corresponding to a specific region ($shop$) in the shopping mall environment.

Apart from pedestrians and shops, the simulated environment consists of walls, obstacles, and other regions that not accessible by pedestrians in the crowd. For example, there are several vases in the walking area of the shopping mall at different locations (see Figure 10 (b)). In this study, the vases are used as obstacles to prevent pedestrians from walking through. Pedestrians need to pay attention and keep a certain distance. Nevertheless, in reality, the planting pot maybe uses for decorating the mall. Many other things can be used for preventing pedestrians from accessing a particular area and making pedestrians change their motion direction, such as barriers, signs, etc. In this study, we defined the positions of the obstacles at different locations inside the shopping mall. Then a safety zone around each obstacle is created to avoid the possibility of overlapping the paths traced by pedestrians with obstacle boundaries. An abounding circle represents a safety zone that the pedestrians can not enter during their motion. The safety zone radius (R) is constant, and the value of R depends on the obstacles' size.

In this simulation, we investigated the pedestrians' movement in the crowd with "200" different small groups, where each group has distinct goal points at known locations. The total number of pedestrians in the crowd is the sum of all pedestrians in all groups, which is equal to "696" pedestrians ($nPedestrians_{total}$ ="696"). Pedestrians' group size distributions in the crowd are illustrated in Figure 11a. As shown

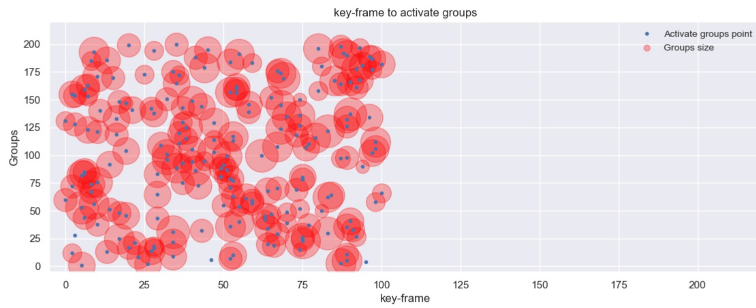

Figure 12: key-frames for activating groups of pedestrians to move in the virtual environment.

in the figure, the groups' size is different in terms of the number of pedestrians. For example, in the condition of pedestrian walking alone, "8" groups (4% of the whole number groups ($nGroups$), where $nGroups$ ="200") are seen in the crowd. The number of each pedestrian's type ($nTypes$="7") in each group is generated randomly between $[0, N_c]$, we assumed that $N_c = 1$, and the obtained results are presented in Figure 11b. From the figure, it is observed that the total number of each type of pedestrian in the crowd is different. For example, the total number of the first type of pedestrian (T_1) is "108" pedestrians (15.5% of the total number of pedestrians in the crowd ($nPedestrians_{total}$), where $nPedestrians_{total}$=696).

In this simulation, seven different types of characters (see Figure 10 (c)) are used to generate "200" groups with "696" pedestrians. Initially, all groups of pedestrians scattered in the front of the virtual environment around the starting point, as shown in Figure 10 (a). In this scenario, the $x - coordinate$ for the starting and ending points for pedestrians is generated randomly between $[(w/2) - 50, (w/2) + 50]$. Moreover, the value of $z - coordinate$ is set to $(l/2) + c_{s1}$, where c_{s1} is determined randomly between $[10, 60]$. Afterwards, the groups of pedestrians were appropriately introduced into the shopping mall environment based on the schedule models. Each group starts to move to enter the mall at randomly created key-frame ($kFrame_{activate}[g_r]$) between $(1, nFrames{*}c_{a1})$, where c_{a1} is set to 0.5. Figure 12 illustrates the activation's' $keyframes$ of all groups; the blue points represent the $keyframes$ for activating groups to move. The red circle represents the groups' size in terms of the number of pedestrians; a larger circle represents a group with a higher number of pedestrians.

Figure 13: Screen-shot of pedestrian flow: pedestrians' trajectories in the final stage of the simulation.

At each key-frame, pedestrians' attributes and their positions are calculated and updated independently in real-time. The maximum number of frames $nFrames$ is set to "200" $keyframes$. The simulation results of the pedestrians' trajectories for all groups at each key-frame are shown in Figure 13. The figure shows that the proposed method allows pedestrians to navigate in the walking area from the starting point to the final destination point. The results have shown that the proposed method prevents pedestrians from colliding with the obstacles and pedestrians in the scene by using the obstacle and pedestrian avoidance method. As the pedestrians move closer to the obstacle and other pedestrians, they need to change their motion direction to avoid collision problems. The proposed method uses BNM for collision avoidance because this method has a high computational performance. Moreover, pedestrians never get stuck in the local minima behind the obstacles, and the BNM method guarantees the collision-free path.

Different screen-shot of the simulation results from different viewpoints presented in Figure 14. As shown in figure, pedestrians attempt to walk in different directions to reach their desired goal points while avoiding obstacles and other pedestrians in the scene. All pedestrians remain with the group during the simulation. Diffident types of walking states are demonstrated in the zoom-in Figure 14b of the particular part of the scene. For example, a pedestrian (A) who has walked alone and changing his motion direction to avoid stairs on his right. Moreover, a pedestrian (B) has a limited space to move after leaving the $shop-6$. At the same time pedestrian (C) has a free moving space to move. At the end of the simulation, all groups terminated in front of the shopping mall, as illustrated in Figure 14a. The figure showed that the most crowded

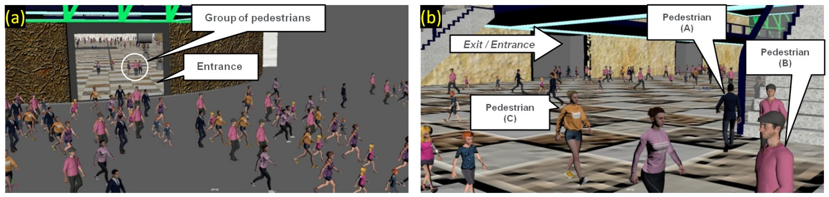

Figure 14: Screen-shot of the simulation: pedestrian's movement towards the goal points in a scene.

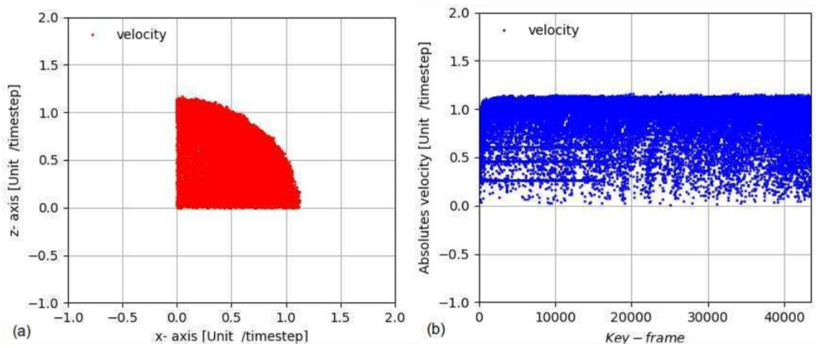

Figure 15: Pedestrian' velocity: change the walking velocity of all pedestrians.

area in the virtual environment is found in front of the mall, where the distance between pedestrians is very small. The calculated results of the pedestrians' velocity of all groups of pedestrians using the proposed method are shown in Figure 15. The left-hand side of the graph represents the pedestrians' velocity in x and $z - directions$, and the right-hand side represents the absolute velocity of all pedestrians in the scene. After the simulation runs for "200" $keyframes$, the computed mean and standard deviation is reached to "0.905" and "0.216", respectively. By comparing the obtained results with the pedestrians' velocity in Subsection 4.1, it can be concluded that the increasing number of groups has the reverse effect on the mean velocity of the pedestrians in the crowd; this is agreed with [37]. In this investigation, we conclude that the groups' formation has an essential effect on the crowd movement. Increasing the groups of pedestrians slow down the pedestrians' movement in the simulated environment.

In this simulation, pedestrians change their position and velocity based on various steering behaviours. In each key-frame, the new position of each pedestrian is calculated based on the proposed method. If the pedestrian does not interfere with the obsta-

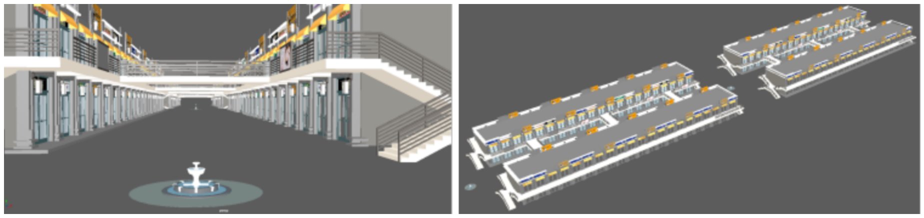

Figure 16: Different screen-shot of the $3D$ model of the shopping street [46] from different viewpoints.

cles and other pedestrians, then the pedestrian's current position will update to the new determined position (see Figure 2). Simultaneously, if the pedestrian interferes with the obstacles or pedestrians, the proposed method will find another pedestrian position based on the obstacle and pedestrian avoidance methods. Then the current pedestrian's position will update with the newly calculated position.

4.3. Simulation of Real-Life Crowd Movements: example 2

This example presents the simulation of pedestrian movement in a large and complex outdoor environment such as a shopping street (see Figure 16). In this scenario, groups of pedestrians are appropriately introduced into the virtual environment through the two entrances. Pedestrians continuously move forward in the navigable area. When they visit all desired places, they leave the simulated environment out through one of the entrances. Entry and exit points are selected randomly for each group. The simulated environment's width w and length l are set to "52" and "330" $unit$, respectively.

In the simulated scenario, pedestrians in the same group have the same goal points ($shops$, $regions$) to visit, and each group is assumed to have a different list of goal points. Inside the simulated environment, there are "12" ($goal\,points$) and "2" general service points that the groups of pedestrians want to visit. The coordinates of the goal points are set to [(7,0,-117),(7,0,-74),(7,0,-31),(7,0,117),(7,0,74),(7,0,31),(-7,0,-117),(-7,0,-74),(-7,0,-31),(-7,0,117),(-7,0,74),(-7,0,31)], respectively, and the services points are located at (-15,0,0) and (15,0,0). We assume that there are two static obstacles $fountain$ ($n3DObstacles$="2") located [(0,0,150), and (0,0,-150) (see Figure 16), and the radius (R) of obstacles are set to "6" units.

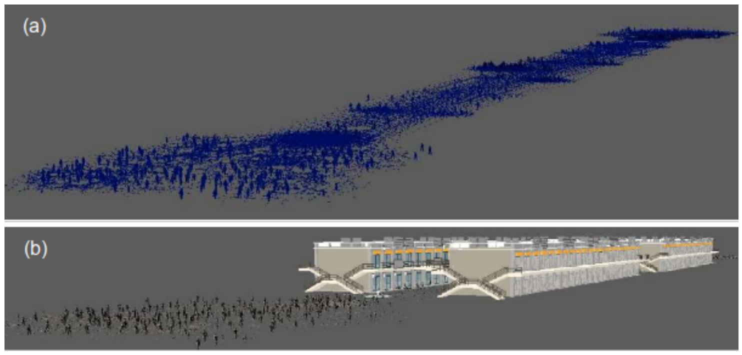

Figure 17: Screen-shot of pedestrian flow: shows the pedestrians' trajectories in the final stage of the simulation.

In this simulation, we investigated the pedestrians' movement in the crowd with "200" different small groups, where each group has distinct goal points at known locations. The total number of pedestrians in the crowd is equal to "696" pedestrians ($nPedestrians_{total}$ ="696"). In this simulation, seven different types of characters (see Figure 10(c)) are used. Pedestrians' group size in the crowd and the number of each pedestrian's type ($nTypes$="7") in each group are generated randomly. Initially, all groups of pedestrians scattered in the front of the virtual environment around the starting point, as illustrated in Figure 17b. Afterward, as explained in the previous subsection, pedestrians are introduced into the simulated environment based on the schedule models.

At each key-frame, pedestrians' attributes and their positions are calculated and updated independently in real-time. The maximum number of frames $nFrames$ is set to "200" $keyframes$. The simulation results of the pedestrians' trajectories for all groups at each key-frame are shown in Figure 17(a). The figure illustrates that the proposed method allows pedestrians to navigate in the walking area from the starting point to the final destination point. Based on the proposed method, every pedestrian in each group in the crowd can constantly adjust their position and optimize their path toward the desired visiting points. During the simulation, pedestrians stay together and maintain the group while pedestrians keep a certain distance. Also, pedestrians can avoid obstacles and other pedestrians in the scene when they move closer. As shown in Figure 17(b), pedestrians walk in different directions to reach their desired goal points while avoiding obstacles, and other pedestrians in the scene when they move closer. During the simulation, pedestrians remain with the group while pedestrians keep a certain distance. At the end of the simulation, all groups terminated the virtual environment in entrance/exit, and they continue to move until they reach the last destination point. As illustrated in the figure, the most crowded area in the virtual environment is found at the entrance, the goal points, and the service point in the walking area.

4.4. Statistical Analysis

To evaluate the performance of the proposed method, we implemented the proposed method for simulating the crowd movement in the virtual environment with different groups (from $1 \rightarrow 10\, groups$) and different scales. For each simulation instance, we perform "200" independently runs for "200" $frames$. For each pedestrian in each group, the proposed method is used to generate pedestrians' trajectories in the virtual environment shown in Figures 10 (a) and 16. The mean and the standard deviation (Std) of the pedestrians' number of each group is calculated. The obtained results are presented in Table 4 and Table 5. The graphical representation of pedestrians' number of each group is illustrated in Figures 18a and 19a.

Table 4: The mean and the standard deviation Std of the pedestrians' number (NOP), and the computational time (CT) (in $seconds$, $[S]$) required to find the trajectories for the pedestrians in each group in $example$ 1.

nGroups	**No. of Pedestrians (NOP)**		**Computational Time (CT)**	
	Mean$_{NOP}$	**Std**$_{NOP}$	**Mean**$_{CT}$	**Std**$_{CT}$
1	6.101	2.405	3.540	1.348
2	12.681	3.384	6.910	1.860
3	18.703	4.332	10.475	2.349
4	24.920	4.996	13.860	2.607
5	31.231	5.375	17.400	2.900
6	36.775	5.581	21.320	3.170
7	42.475	6.532	24.335	3.272
8	49.077	7.175	27.830	3.906
9	56.590	7.667	31.810	4.006
10	61.838	7.701	35.375	4.105

For each simulation scenario, as shown in tables, the generated pedestrians in different groups are located randomly at the front of the simulated environment. Then,

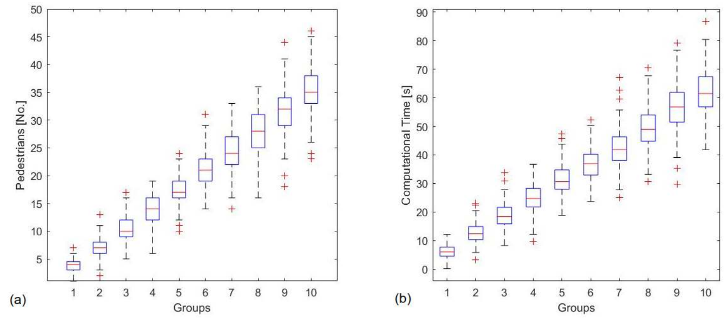

Figure 18: Simulation and performance evaluation ($example$ 1): (a) the number of pedestrians, and (b) the computational time required to generate the pedestrians' trajectories in each group using the proposed method.

each group g_r, ($g_r = 1, ..., nGroups$) starts to move to enter the simulated environment at a specific key-frame number ($kFrame_{activate}[g_r]$=1). After that, pedestrians continuously move forward to reach their desired goal points, where the number of goal points and their sequence are defined are provided in Subsections 4.2 and 4.3. In these scenarios, pedestrians' attributes and their movement are calculated in each key-frame, where the maximum number of $keyframes$ ($nFrames$) is set to "200" $keyframes$. All other parameters are the same as the previous simulations.

Table 5: The mean and the standard deviation Std of the pedestrians' number (NOP), and the computational time (CT) (in $seconds$, $[S]$) required to find the trajectories for the pedestrians in each group in $example$ 2.

nGroups	**No. of Pedestrians (NOP)**		**Computational Time (CT)**	
	Mean$_{NOP}$	**Std**$_{NOP}$	**Mean**$_{CT}$	**Std**$_{CT}$
1	3.465	1.33	1.875	0.656
2	6.53	1.836	3.574	0.927
3	10.725	2.238	5.955	1.176
4	13.95	2.69	7.95	2.241
5	20.93	3.287	11.815	5.021
6	20.93	3.287	11.815	5.021
7	24.62	3.404	13.646	1.793
8	27.59	3.964	15.367	3.977
9	31.485	4.06	17.455	4.124
10	34.865	4.577	19.932	2.956

At each independent run, the proposed method calculated the total computational time required to generate the pedestrians' trajectories while avoiding obstacles and other pedestrians in the scene. Tables 4 and 5 presents the calculated mean and the

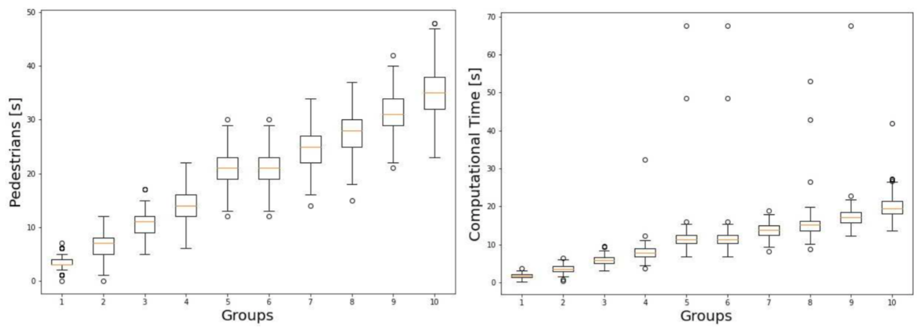

Figure 19: Simulation and performance evaluation ($example\,2$): (a) the number of pedestrians, and (b) the computational time required to generate the pedestrians' trajectories in each group using the proposed method.

standard deviation (Std) of the total computational time, and the simulation results is illustrated in Figure 18b and 19b. The obtained results reveal that the proposed method can generate the trajectories for the groups of pedestrians navigating in the virtual environment to visit several goal points within a reasonable computational time. Moreover, the obtained results reveal that the mean value of the computational time is not increased significantly with increasing the number of pedestrians in the crowd. The computational time depends on the number of pedestrians, the number of goal points and their ordering, the complexity of geometric problems, pedestrian avoidance, etc.

4.5. Comparison with Different Methods

This section presents the comparison between the proposed method and the related methods based on the evaluation criteria described in [48]. However, setting uniform evaluation criteria for different categories of models is quite difficult. The authors in [48] collected and analyzed the information provided by the papers describing different models to simulate crowds in the years 2000-2020. This table shows the group behaviour of the models that have been used previously in the field of crowd simulation. In order to compare the proposed method with these models, the group behaviours of the proposed method for simulating the crowd movement are defined and presented in Table 6. In this study, each pedestrian in each group is influenced by other pedestrians in the same group. The pedestrian-to-pedestrian influencing relationships inside a group are referred to as intra-group structure. However, a group of pedestrians in

Table 6: Ability of simulating group dynamics with the proposed method for the crowd simulation (GA: Group Avoidance, GI: Group–pedestrian Interaction, InterE: Inter-group Emotion Contagion. LF: Leader–Follower, GF: Group Formation, GS: Group Structure, GC: Group Cohesion, GCo: Group Cooperation, PB: Peer Behavior, IntraE: Intra-group Emotion Contagion). A similar table is described by [48].

Method	**Inter-Group**			**Intra-Group**						
	GA	**GI**	**InterE**	**LF**	**GF**	**GS**	**GC**	**GCo**	**PB**	**IntraE**
Ref [36]	×	×	×	×	✓	✓	✓	×	×	×
Ref [49]	✓	✓	×	×	×	✓	✓	×	×	×
Ref [35]	✓	✓	×	✓	×	×	✓	×	×	×
Ref [37]	✓	✓	×	✓	×	×	✓	×	×	×
Ref [30]	×	×	×	✓	×	✓	✓	×	×	×
Ref [50]	✓	✓	×	✓	×	×	✓	×	×	×
Ref [2]	✓	✓	×	✓	✓	✓	✓	✓	×	×
Proposed Method	×	×	×	×	✓	✓	✓	✓	✓	✓

the crowd is not influenced by other groups, and the relationships s group-to-group are referred to as inter-group relationship [30]. As shown in Table 6, the proposed method focus on intra-group interactions such as group formation, group structure, group cohesion, group cooperation, peer behaviour, Intra-group Emotion Contagion. Moreover, this method pays less attention to the inter-group relationship. By comparing the obtained results based on the proposed method with the other models that have been used previously (see Table 6), one can conclude that the proposed method can simulate different scales of crowds (small scale, pedestrians < 200, medium scale, 200 < pedestrians < 500, large scale, 500 < pedestrians < 1000). In contrast, other $agent-based$ models cannot simulate large-scale crowds.

5. Conclusion

In this study, a new method is developed for simulating pedestrian crowd movement in a virtual environment. In this study, the virtual pedestrians in many groups navigated in the virtual environment with different directions to reach their distinct destination points. The developed method uses the $multi-group$ model for generating a real-time trajectory for each pedestrian in the crowd. Additionally, an $agent-based$ model is introduced into the developed method for modelling the pedestrians' behaviours. Each pedestrian in the crowd has a particular set of data that represents the characteristics of the pedestrian. Moreover, various steering behaviours are introduced. Several techniques have been presented for combining steering behaviours to a single steering force to allow the pedestrians to walk toward their goal points. The developed method demonstrates how the pedestrians choose their path with their group in a virtual environment toward their goal points while avoiding static obstacles and other pedestrians in the scene. Based on this method, each pedestrian in each group constantly adjusts his/her paths toward the desired goal point and updates his/her attributes independently in real-time. The proposed method focuses on intra-group interactions and this method pays less attention to the inter-group relationship.

The obtained results demonstrate that the developed method has been well applied to simulate pedestrian crowd movement in a virtual environment. Moreover, the obtained results reveal that the proposed method can generate each pedestrian's trajectories in each group in the virtual environment independently to reach several goal points within a reasonable computational time. Moreover, the obtained results reveal that the mean value of the computational time is not increased significantly with the increasing of the number of pedestrians in the crowd. From the simulation results, it can be observed that the maximum value of the pedestrian's velocity reaches 1.139 $units/timestep$. In contrast, the minimum velocity is achieved near the goal points. In this study, the minimum distance between the current pedestrian and all other pedestrians is calculated, and the obtained results show that the minimum distances do not fall below 1.8017 $units$. The study can help to better understanding and respecting different cultures' personal space. Even identify the minimum personal space requirements needed to protect health and limit the spread of the disease. By comparing the obtained results based on the proposed method with the other models that have been used previously, one can conclude that the proposed method can simulate different scales of crowds. In contrast, other $agent - based$ models cannot simulate large-scale crowds. The developed method provides a collision-free path for pedestrians to reach their goal points, and pedestrians start to change their direction as they move closer to obstacles or other pedestrians in the scene. Furthermore, it is concluded that the developed method had achieved good results in terms of safety and accuracy. As far as the future works are concerned, we would like to test the proposed method and strategies in real environments. A first possibility will be to apply the proposed techniques to wheel robots. Then, to slightly increase the complexity we will employ several NAO humanoid robots as crowd and different settings for the obstacles. One final direction will be to assign certain priorities to each robot and take them into account when devising their paths to each destination.

Acknowledgements

This research was partially funded by the H2020 project STAR - Nove AI technology for dynamic and unpredictable manufacturing environments (grant number 956573).

References

[1] Y. Yuan, B. Goni-Ros, H. H. Bui, W. Daamen, H. L. Vu, S. P. Hoogendoorn, Macroscopic pedestrian flow simulation using smoothed particle hydrodynamics (sph), Transportation Research Part C: Emerging Technologies 111 (2020) 334–351.

[2] Z. Ren, P. Charalambous, J. Bruneau, Q. Peng, J. Pettré, Group modeling: A unified velocity-based approach, in: Computer Graphics Forum, Vol. 36, Wiley Online Library, 2017, pp. 45–56.

[3] S. Ali, K. Nishino, D. Manocha, M. Shah (Eds.), Modeling, Simulation and Visual Analysis of Crowds - A Multidisciplinary Perspective, Vol. 11 of The International Series in Video Computing, Springer, 2013.

[4] D. Helbing, I. J. Farkas, P. Molnar, T. Vicsek, Simulation of pedestrian crowds in normal and evacuation situations, Springer Berlin, 2002, pp. 21–58.

[5] S. J. Guy, Geometric collision avoidance for heterogeneous crowd simulation, Ph.D. thesis, USA, aAI3545519 (2012).

[6] A. Treuille, S. Cooper, Z. Popović, Continuum crowds, ACM Trans. Graph. 25 (3) (2006) 1160–1168.

[7] R. Narain, A. Golas, S. Curtis, M. C. Lin, Aggregate dynamics for dense crowd simulation 28 (5) (2009) 1–8.

[8] R. L. Hughes, A continuum theory for the flow of pedestrians, Transportation Research Part B: Methodological 36 (6) (2002) 507–535.

[9] R. L. Hughes, The flow of human crowds, Annual Review of Fluid Mechanics 35 (1) (2003) 169–182.

[10] C. K. Lim, K. L. Tan, A. A. Zaidan, B. B. Zaidan, A proposed methodology of bringing past life in digital cultural heritage through crowd simulation: a case study in george town, malaysia, Multimedia Tools and Applications 79 (5) (2020) 3387–3423.

[11] S. J. Guy, S. Kim, M. C. Lin, D. Manocha, Simulating heterogeneous crowd behaviors using personality trait theory, in: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 43–52.

[12] P. Degond, C. Appert-Rolland, M. Moussaïd, J. Pettré, G. Theraulaz, A hierarchy of heuristic-based models of crowd dynamics, Journal of Statistical Physics 152 (6) (2013) 1033–1068.

[13] L. Cheng, V. Reddy, C. Fookes, P. K. V. Yarlagadda, Impact of passenger group dynamics on an airport evacuation process using an agent-based model, in: 2014 International Conference on Computational Science and Computational Intelligence, Vol. 2, 2014, pp. 161–167. doi:10.1109/CSCI.2014.111.

[14] A. Sud, E. Andersen, S. Curtis, M. Lin, D. Manocha, Real-time path planning for virtual agents in dynamic environments, in: 2007 IEEE Virtual Reality Conference, 2007, pp. 91–98.

[15] C. Cao, P. Trautman, S. Iba, Dynamic channel: A planning framework for crowd navigation, in: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019, IEEE, 2019, pp. 5551–5557.

[16] S. Patil, J. Van Den Berg, S. Curtis, M. C. Lin, D. Manocha, Directing crowd simulations using navigation fields, IEEE transactions on visualization and computer graphics 17 (2) (2010) 244–254.

[17] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, SIGGRAPH Comput. Graph. 21 (4) (1987) 25–34.

[18] D. Helbing, P. Molnár, Social force model for pedestrian dynamics, Phys. Rev. E 51 (1995) 4282–4286.

[19] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, S. Donikian, Experiment-based modeling, simulation and validation of interactions between virtual walkers, in: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 189–198.

[20] B. Yersin, J. Maïm, P. Ciechomski, S. Schertenleib, D. Thalmann, Steering a virtual crowd based on a semantically augmented navigation graph, in: Proc. The First International Workshop on Crowd Simulation (V-CROWDS'05), Lausanne, Switzerland, 2005, pp. 169–178.

[21] G. K. Zipf, Human behavior and the principle of least effort., Addison-Wesley Press, 1949.

[22] I. Karamouzas, P. Heil, P. van Beek, M. H. Overmars, A predictive collision avoidance model for pedestrian simulation, in: A. Egges, R. Geraerts, M. Overmars (Eds.), Motion in Games, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 41–52.

[23] S. Sarmady, F. Haron, A. Z. H. Talib, Modeling groups of pedestrians in least effort crowd movements using cellular automata, in: 2009 Third Asia International Conference on Modelling & Simulation, IEEE, 2009, pp. 520–525.

[24] G. K. Still, Crowd dynamics, Ph.D. thesis, University of Warwick (2000).

[25] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (2) (1968) 100–107.

[26] S. M. LaValle, Planning Algorithms, Cambridge University Press, 2006. doi:10.1017/CBO9780511546877.

[27] S. Leroy, J. Laumond, T. Siméon, Multiple path coordination for mobile robots: A geometric algorithm, in: T. Dean (Ed.), Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages, Morgan Kaufmann, 1999, pp. 1118–1123.

[28] O. B. Bayazit, J.-M. Lien, N. M. Amato, Better group behaviors in complex environments using global roadmaps, in: Proceedings of the Eighth International Conference on Artificial Life, ICAL 2003, MIT Press, Cambridge, MA, USA, 2002, p. 362–370.

[29] W. Shao, D. Terzopoulos, Autonomous pedestrians, Graphical Models 69 (5) (2007) 246–274, special Issue on SCA 2005.

[30] F. Qiu, X. Hu, Modeling group structures in pedestrian crowd simulation, Simulation Modelling Practice and Theory 18 (2) (2010) 190–205.

[31] N. Mahato, A. Klar, S. Tiwari, Particle methods for multi-group pedestrian flow, Applied Mathematical Modelling 53 (2018) 447–461.

[32] L. Manenti, S. Manzoni, Crystals of crowd: Modelling pedestrian groups using mas-based approach, in: G. Fortino, A. Garro, L. Palopoli, W. Russo, G. Spezzano (Eds.), Proceedings of the 12th Workshop on Objects and Agents, Rende (CS), Italy, Jul 4-6, 2011, Vol. 741 of CEUR Workshop Proceedings, CEUR-WS.org, 2011, pp. 51–57.

[33] N. Pelechano, J. M. Allbeck, N. I. Badler, Controlling individual agents in high-density crowd simulation, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '07, Eurographics Association, Goslar, DEU, 2007, p. 99–108.

[34] B. Piccoli, A. Tosin, Pedestrian flows in bounded domains with obstacles, Continuum Mechanics and Thermodynamics 21 (2) (2009) 85–107.

[35] N. Jaklin, A. Kremyzas, R. Geraerts, Adding sociality to virtual pedestrian groups, in: Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology, VRST '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 163–172.

[36] I. Karamouzas, M. Overmars, Simulating and evaluating the local behavior of small pedestrian groups, IEEE Transactions on Visualization and Computer Graphics 18 (3) (2011) 394–406.

[37] J. Bruneau, A. Olivier, J. Pettre, Going through, going around: A study on individual avoidance of groups, IEEE Transactions on Visualization & Computer Graphics 21 (04) (2015) 520–528.

[38] S. Zhang, J. Zhang, M. Chraibi, W. Song, A speed-based model for crowd simulation considering walking preferences, Communications in Nonlinear Science and Numerical Simulation 95 (2021) 105624.

[39] M.-T. W. Michael Fischer, Gaspard Jankowiak, Micro- and macroscopic modeling of crowding and pushing in corridors, Networks & Heterogeneous Media 15 (3) (2020) 405–426.

[40] W. van Toll, J. Pettre, Algorithms for microscopic crowd simulation: Advancements in the 2010s, Computer Graphics Forum 40 (2) (2021) 731–754. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.142664.

[41] C. W. Reynolds, Steering behaviors for autonomous characters, in: Game developers conference, Vol. 1999, Citeseer, 1999, pp. 763–782.

[42] R. Saeed, D. R. Recupero, P. Remagnino, A boundary node method for path planning of mobile robots, Robotics and Autonomous Systems 123 (2020) 103320.

[43] R. Saeed, D. R. Recupero, Path planning of a mobile robot in grid space using boundary node method, Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2019 - Volume 2 (2019) 159–166.

[44] Autodesk, Maya, https://www.autodesk.com/maya/ (2018).

[45] R. A. Saeed, D. Reforgiato Recupero, P. Remagnino, The boundary node method for multi-robot multi-goal path planning problems, Expert Systems (2021) e12691.

[46] kofta55, Commercial mall 3d model, https://free3d.com/3d-model/commercial-mall-689388.html (2019).

[47] Adobe, Mixamo, https://www.mixamo.com/ (2018).

[48] S. Yang, T. Li, X. Gong, B. Peng, J. Hu, A review on crowd simulation and modeling, Graphical Models 111 (2020) 101081.

[49] L. He, J. Pan, S. Narang, D. Manocha, Dynamic group behaviors for interactive crowd simulation, in: Symposium on Computer Animation, 2016.

[50] S. Lemercier, J.-M. Auberlet, Towards more behaviours in crowd simulation, Computer Animation and Virtual Worlds 27 (1) (2016) 24–34.

Declaration of interests

☑ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Credit Author Statement

R A Saeed: Investigation, Write and edit source code, Writing original draft preparation.

Diego Reforgiato Recupero: Supervision, Reviewing and Editing

Paolo Remagnino: Conceptualization, Methodology, Writing-Reviewing and Editing