PUCRS

FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

PAULO RICARDO KNOB

SIMULANDO DIVERSIDADE DE COMPORTAMENTOS EM BIOCROWDS

Porto Alegre
2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
GRADUATE PROGRAM IN COMPUTER SCIENCE

SIMULATING BEHAVIOR DIVERSITY IN BIOCROWDS

PAULO RICARDO KNOB

Dissertation presented as partial requirement for obtaining the degree of Master in Computer Science at Pontifical Catholic University of Rio Grande do Sul.

Advisor: Prof. Soraia Raupp Musse

Porto Alegre
2018

Ficha Catalográfica

K72s Knob, Paulo Ricardo

Simulando Diversidade de Comportamentos em Biocrowds / Paulo Ricardo Knob . – 2018.

85 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Soraia Raupp Musse.

1. Simulação de Multidões. 2. Culturalidade. 3. Conforto da Multidão. 4. Ambientes Complexos. I. Musse, Soraia Raupp. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Paulo Ricardo Knob

Simulating Behavior Diversity in BioCrowds

This Dissertation/Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor/Master of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifícia Universidade Católica do Rio Grande do Sul.

Sanctioned on August 22nd, 2018.

COMMITTEE MEMBERS:

Prof. Dr. Márcio Sarroglia Pinho (PPGCC/PUCRS)

Prof. Dr. Léo Pini Magalhães (FEEC-DCA/UNICAMP)

Prof. Dr. Soraia Raupp Musse (PPGCC/PUCRS - Advisor)

SIMULANDO DIVERSIDADE DE COMPORTAMENTOS EM BIOCROWDS

RESUMO

A maioria das técnicas disponíveis hoje em dia para simulação de multidões está focada em uma situação específica, por exemplo, evacuação em eventos perigosos. Poucas técnicas consideram os aspectos culturais e de personalidade presentes em uma sociedade para determinar o comportamento dos agentes. Portanto, este trabalho tem como objetivo construir um framework capaz de lidar com diferentes características culturais e de personalidade como input, traduzindo-as em uma parametrização de grupo, que vai determinar o comportamento de grupos e multidões em ambientes virtuais. Além disso, incluímos no BioCrowds uma resposta de conforto para os agentes, em termos de densidade e características térmicas do ambiente. Os resultados indicam que os mapeamentos culturais / psicológicos parecem promissores, uma vez que os agentes foram capazes de se comportar conforme o esperado. Além disso, os agentes foram capazes de reagir devido ao conforto térmico e de densidade, melhorando sua capacidade de reagir às mudanças do ambiente.

Palavras Chave: Simulação de multidões, Culturalidade, Conforto da Multidão, Ambientes Complexos.

SIMULATING BEHAVIOR DIVERSITY IN BIOCROWDS

ABSTRACT

Most of the techniques available nowadays for crowd simulation are focused on a specific situation, e.g. evacuation in hazardous events. Very few of them consider the cultural and personality aspects present in a society to determine the behavior of agents. Therefore, this work aims to build a framework able to deal with different cultural and personality traits as input, and translate them into a group parametrization, which is going to determine the behavior of groups and crowds in virtual environments. Also, we include in BioCrowds a comfort response for agents, in terms of density and thermal characteristics of the environment. Results indicate that the cultural/psychological mappings seem promising, since agents were able to perform as intended. Additionally, agents were able to react due to thermal and density comfort, improving their ability to react to environmental changes.

Keywords: Crowd simulation, Culturality, Crowd Comfort, Complex Environments.

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

Crowd, as defined by Bon [38], is an agglomeration of individuals. From a psychological point of view, he points out that that these individuals present different behaviors of those they would present if they were alone. Although there is no consensus about the types of a crowd, Momboisse [43] classifies it in four types: casual, conventional, expressive and aggressive. Yet, Berlonghi [1] prefers to treat the crowd as spectator, demonstrator and people that want to escape.

The area of crowd simulation has been the studying focus for several researchers through many years because of its numerous and varied applications in diverse fields. It can be used to simulate large crowds of people [57], crowd's movement [9, 56, 48, 28], to create and test evacuation plans [5], among others. Nevertheless, there are many open challenges to tackle. Although the existence of a large number of techniques in literature [9, 29, 48, 55, 21, 49, 56], most of them are focused on a specific situation to be simulated where agents are endowed with skills to perceive the world, seek goals, avoid collisions and other related behaviors. One existing challenge is to simulate realistic individuals in crowds when considering the possible diversity of behaviors caused by cultural and personality aspects.

Very few of the existing methods for crowd simulation take into account the cultural aspects of a nation or society [37, 39]. These aspects can be very useful in providing specific information about a crowd, which can be translated into different behaviors such as desired speed, group cohesion, gesturing, eye gazing, among others. It can be useful to simulate different crowds from all around the world, like an urban simulation running with Chinese agents or a game where the history is set in Germany and, therefore, has Germanic agents. One example of cultural analysis is proposed by Chattaraj et al. [6] where he measured and compared the Fundamental Diagrams, based on density and speed of population, of Germany and India. The Fundamental Diagrams were originally proposed by Jelic et al. [34] to be used in traffic planning. In short, the diagrams are used to show the relationship between three different characteristics of the crowd: density of people, speed and flow. By this definition, Chattaraj et al. [6] proposed that cultural differences between people can also affect their speed, density and flow. Following the results found in their tests, the authors perceived that the speed of the Indian people was less dependent on density than the German people. Yet, they mention about the indicators that the more unordered behavior of Indians was more effective than the more ordered behavior of Germans. Therefore, it suggest that the differences found in their Fundamental Diagrams are due to cultural differences between both population. So, a cultural simulation should be able to replicate such behavior, not just for these two countries, but for many different cultures.

The goal of this work is to extend the Biocrowds [9] model, making the agent's navigation more realistic including the following contributions:

- to build a framework able to experiment different behaviors and analysis in crowds;

- to consider as input the cultural aspects of a population in order to provide diverse behaviors in virtual humans simulation; and
- to consider as input the personality aspects of population in order to provide behavioral diversity in crowds.

Therefore, to provide cultural diversity, it is proposed to use two different methodologies as input to this framework, namely Hofstede's Cultural Dimensions (HCD) [33] and Durupinar [13]. HCD is a very consolidated methodology to evaluate the cultural dimensions of Countries, while Durupinar model states for a computational method to simulate virtual humans based on their personalities. While HCD is not a simulation methodology, we use it as a way to map characteristics in crowds, as will be discussed in this work. Indeed, our framework should be able to receive one of these two cultural/personalities aspects as input, and use them to define group parameters to control virtual agents in BioCrowds.

In addition, we included in BioCrowds the comfort response in terms of density and the thermal characteristics of the space. Our goal is to be able to simulate agents avoiding uncomfortable places in the environment and seek for cozy locations. It is important to notice that cultural, personality aspects and response to comfort relies on more realistic agents reactions w.r.t. environments and groups in the crowd. Our goal is to include various possibilities to simulate behavioral diversity in BioCrowds.

This work is structured as follows. Chapter 2 presents several works in the area of virtual human simulation, as well elucidates different techniques for crowd simulation and works concerning population cultural aspects and psychological traits. Chapter 3 explains the method proposed in this work, extending the original Biocrowds's method in order to achieve the proposed diversity aspects of people. Chapter 4 presents some preliminary results of this model for the proposed method. Finally, Chapter 5 addresses some conclusions as well the future work that we are yet planning to do.

2. RELATED WORKS

Several ways to simulate crowds were developed in last years. The origin of crowd simulation goes back to Reynolds [48] and Helbing [29] papers, which evolved in time thanks to many contributions. More recently, the cultural aspects started to gain space in some models, like the work of Mascarenhas et al. [39] and Lala et al. [37], which aim to add this new context in crowd methods. This section focus on some papers in crowd simulation area, as well some work focused on gathering society cultural aspects with simulation of agents.

2.1 Virtual Human Modeling and Simulation

In his work, Reynolds [48] simulated flocks of bird-like entities which he called "boids", obtaining a realistic animation using only simple local rules. His goal was to simulate the movement of bird flocks, schools of fishes and herds of animals, where the "boid" could perceive the local environment and react, maintaining his position and orientation in relation to the group (flock), following three rules:

- **Separation**: each "boid" maintains a minimal distance from his neighbors to avoid collision;
- **Alignment**: each "boid" adjusts his velocity vector to keep a coherent trajectory in relation to his neighbors; and
- **Cohesion**: each "boid" maintain his position near the center of his neighbors positions.

Helbing et al. [29] proposed a psychosocial forces based model to reproduce the pedestrian dynamic. This concept is defined based on the assumption that pedestrians adopt behavioral strategies according stimulus from routine situations. Therefore, he says that their actions are often automatic and predictable. Yet, According to Fruin [21], crowd behavior is related with territorial perception of each individual. So, the way people move in the environment and position themselves in relation to others is affected by how the space is seen and measured.

Another work in the area is proposed by Treuille et al. [55], where they define four hypothesis which control the crowd dynamics:

- Each person tries to achieve a spatial goal;
- Each person moves at its maximum possible speed;
- There is a discomfort field g which makes people prefer to be in the position x instead of x' if $g(x') > g(x)$; and
- People will choose paths minimizing a linear combination of path length (α), time to the goal (β) and discomfort on the way (γ), following: $\int_p C ds$, where $C \equiv (\alpha f + \beta + \gamma g)/f$.

Van den Berg et al. [56] named Optimal Reciprocal Collision Avoidance (ORCA). It is a velocity-based method for collision avoidance between multiple agents and it was developed originally for the robot industry. The main idea is to find the velocity obstacle (VO) between two agents, which defines the zones where a collision should occur between them. With this, it is possible to deduce the areas outside this VO between agent A and agent B and, therefore, the optimal reciprocal collision avoidance velocity for them. One of the greatest features of this method is that it runs at O(n), since each agent can perceive all agents in simulation to define its own speed.

Using the ORCA method, He et al. [27] present an algorithm to simulate group behavior, similar with the observed behavior in real life. Such groups are dynamic, meaning each one can have any format and number of agents inside it. The concept used for ORCA of Velocity Obstacles (VO) is used in this work for collision avoidance purposes between groups. Even so, it is possible for a large group to divide itself when passing by another group, just to join again later. Figure 2.1 shows how the groups work.

As can be seen in Figure 2.1, the initial formation of the groups is based solely on the initial position of the agents, using a spatial clustering algorithm. During the simulation, each group tries to maintain its structure as long it is possible. Each agent of the group (except the group leader) chooses another fellow agent to follow during the simulation, so the method does not need to explicitly check for collisions between agents inside the same group. For an efficient collision avoidance between groups, ORCA is used to avoid collision between the groups leader and other groups. Then, a suitable intended velocity is calculated for the other agents of the group.

Besides being able to re-join with its original group, or even join another group, each agent can leave the group at any moment if it will be easy for him/her to achieve the goal. The results of this work show that the proposed method was able to generate group behaviors similar with the ones observed in real life. A limitation of the method is that it does not take into account the personality of the individuals, neither the cultural aspects of the crowd or the concept of personal space.

Krontiris et al. [36] also try to add more realism to the crowd. More specifically, their work aims to develop an activity-centric framework for authoring functional, heterogeneous virtual crowds in semantically meaningful environments. The simulated environment is classified with attractors (for example: food, coffee or vendor) and each agent computes the so-called "influence maps", which is the influence of its surroundings on its behavior. Besides locations, it is possible for an agent be influenced by other agents, which allows the framework to work with group behavior as well. To determine an agent influence map, Equation 2.1 is used:

$$M_\alpha = \langle \{I_l\}, \{I_g\}, I_d \rangle, \tag{2.1}$$

where $\{I_l\}$ are the influences that an agent has from the locations, $\{I_g\}$ are the influences that this same agent has from the other agents inside its group (if any) and I_d is the influence for the final destination of this agent. Therefore, the target location for an agent is the location of the attractor which has the maximum influence over it, being it from environment or group. So, it is possible to simulate a group behavior (for example: a family) just tweaking the group influence to be the

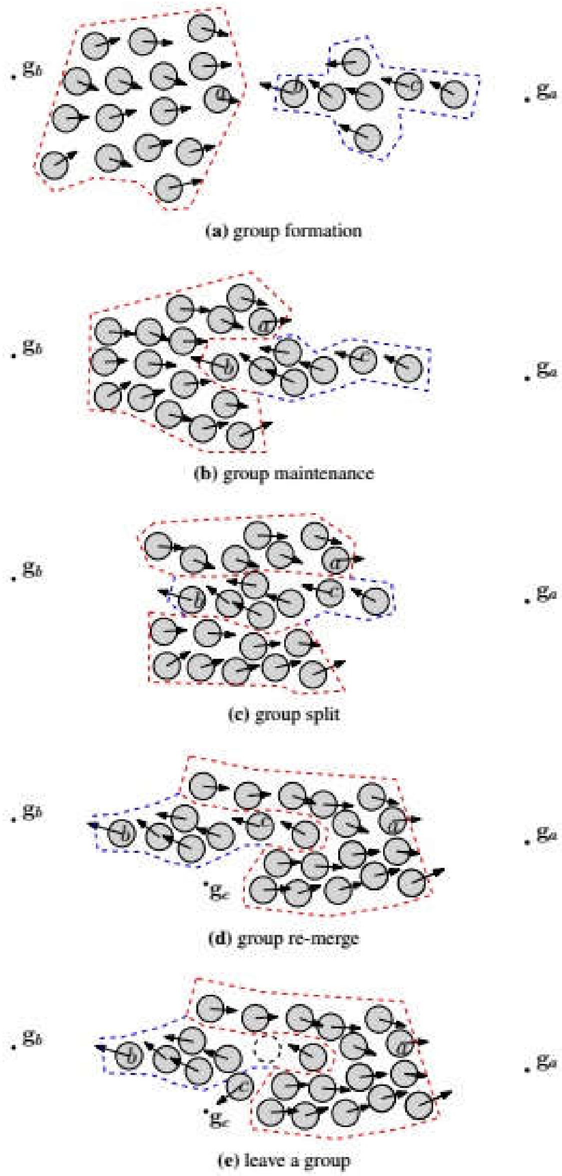


Figure 2.1 – Group behaviors from [27]. In (a) a spatial clustering algorithm for the groups. The groups are maintained as long it is possible (b), but can split (c) when needed. Agents have the capacity to join again with the group when possible (d). If an agent can easily reach its goal, it just leaves the group (e) and goes towards the objective.

highest. Tests were ran with three simulated environments: a shopping mall, an art museum and an airport. The idea was to show that the proposed method is able to show heterogeneous behavior and that it can be applied in varying and different scenarios. Results show that the framework was able to perform as expected, allowing to build independent scenarios and crowds. Also, the method was able to perform as intended in three different scenarios, were agents had different behaviors and desires.

The work of Cassol et al. [5] claims to be the first one in the literature to propose a metric to evaluate quantitatively an evacuation performance, which was validated by a safety expert and tested in a real-life scenario. Despite the complexity involving virtual simulation of an evacuation process, their method was able to create a new metric to measure evacuation performance, considering evacuation time, speed and density. In addition, a framework was created to find the optimal evacuation strategies in complex environments, using both simple brute force (BF) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [25]. Tests were run using CrowdSim [4], as the simulation software, and the results show that both BF and CMA-ES had similar results. Yet, they conclude that the initial crowd configuration can be crucial to determine an optimal evacuation plan.

The crowd behavior is something hard to evaluate and validate, once it is very difficult to measure all behaviors that can influence it. Despite this, it is possible to evaluate and validate its behavior both qualitatively and quantitatively [47, 44]. According to the literature [30, 31, 21, 52, 28, 53], two types of behaviors can be observed: inherent and emerging. Inherent behaviors are the ones owned by each person. Although they may present some irregularities due to a lack of practice or knowledge (for example, children) [28], this type of behavior is used in order to help a person to move around in the most efficient way according to a specific environment. The three main inherent behaviors cited in literature [30, 31, 21, 52, 28] are:

- Goal Seeking: the movement of people is influenced by the willingness of each individual to reach a particular destination;
- Collision Avoidance: the movement of people is, also, influenced by the willingness of avoiding collision with obstacles and other person; and
- Least Effort: people tend to choose a path where they are going to spend less energy, for example, a straight path between their current positions and their goals. The collision avoidance behavior may alter this selected path, since it can be necessary to change a person orientation in order to avoid a collision.

Emerging behaviors are the ones that appear spontaneously in a crowd due to interaction between people inside it. The main ones cited in literature [30, 31, 21, 52, 28, 53] are:

- Lane Formation: when two groups move in opposed directions, lanes of people tend to emerge in both groups. It happens following the least effort rule, since it is easier for a person to follow another one in front of it, instead to try to open space for itself (Figure 2.2);
- Prior organization: when two crowded groups move in opposed directions, it is possible to compact small groups temporally in order to open space for other people to pass, reducing necessary effort;
- Speed Reduction: in a high density crowd, it is logical to presume that each person has low space to move. It affects directly its speed, making people move slower;
- Arc Formation: when a great number of people move towards a small passage (for example, leaving a nightclub), they tend to form a geometrical formation in form of an arc around this passage, due their willingness to stay close to it;

Figure 2.2 – Lanes forming in different groups [52]. It is easier for a person to follow another one in front of it, instead to try to open space for itself.

- Bottleneck Effect: when people need to move toward a narrow passage (for example, a corridor), they tend to stick together, as well reduce speed while inside it. The region right before this narrow passage becomes denser, while the region right after allows to people spread out;
- Corner Effect: in a high density crowd, passages with a corner tend to reduce people speed, since the space is not used effectively; and
- Shockwaves Effect: in a high density crowd, frequently the people movement will push others. It can generate a wave effect since this pushing can be propagated between people.

In this section we presented some general characteristics about crowd simulation, as well some work in this area. Section 2.2 depicts the three main methods used to simulate agents, known as social forces, space subdivision and velocity-based.

2.2 Simulation Techniques

Agents can be simulated in several ways, using various techniques and aiming different purposes like entertainment, evacuation plans, urban planning, among others. In last years, many approaches were developed and improved. In this work, the simulation model known as BioCrowds [9] is used, which has the key idea to explicitly represent free spaces in a virtual environment. This section aims to explain Biocrowds among other known simulation methods for agents as Social forces, based on Space discretization (BioCrowds) and Velocity Based.

2.2.1 Social Forces

The work of Helbing [29] makes pedestrian movement follows what is called "social forces", which are a measure of the motivations of each agent to do something (i.e. move). In other words, this concept is defined from the assumption that pedestrians adopt behavioral strategies according stimulus from routine situations. Therefore, their actions are often automatic and predictable. Figure 2.3 presents a schematic representation of the processes that can lead to behavioral changes. Three force terms are essential:

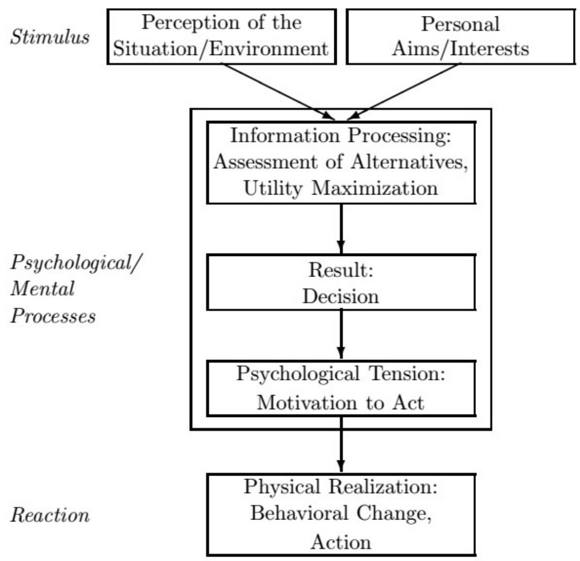


Figure 2.3 – A schematic representation of the processes that can lead to behavioral changes. [29]

- A term describing the acceleration of the agent towards its goal;
- A term describing the distance an agent keeps from other agents and borders; and
- A term describing attractive effects.

The authors define the term *fluctuation* as a measure of behavior variations an agent can present, either because of accidental or willing deviations from its desired movement. Each stimuli an agent can receive is called a *psycho-social force*. So, each *psycho-social force* f_α represents a different influence that an agent α can receive and alter its behavior, either deriving from another agent or the environment itself. This f_α along the *fluctuation* determines the agent's speed modification in time, given by $\frac{dv_\alpha}{dt}$. Therefore, the speed modification by time can be given by Equation 2.2:

$$\frac{dv_\alpha}{dt} = f_\alpha(t) + \textit{fluctuations}, \tag{2.2}$$

where f_α is responsible for keeping the agent away from obstacles and other agents (i.e. avoid collision) as well to keep *known* agents close, like family or friends. Figure 2.4 shows an example of simulation. If one agent can pass through a narrow door, any other agent with the same desired direction can easily follow it. Yet, agents with an opposite desired direction need to wait. The size of the circles represent the motion velocity.

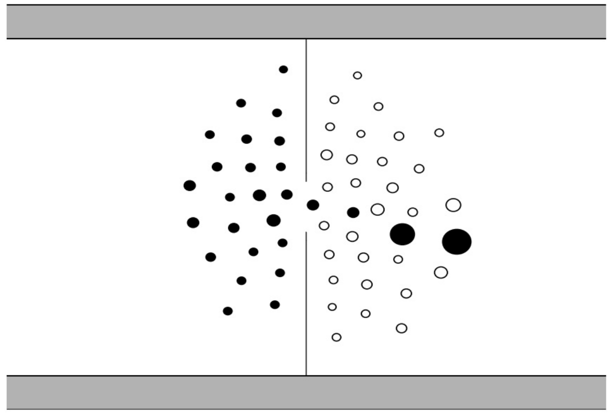


Figure 2.4 – An example simulation. If one agent can pass through a narrow door, any other agent with the same desired direction can easily follow it. Yet, agents with an opposite desired direction need to wait. The size of the circles represent the motion velocity. [29]

2.2.2 Space Subdivision

A simulation technique based on space subdivision aims to subdivide the space where agents can walk. Yersin et al. [57] focus their work in densely populate large-scale environments. Their idea is to be able to simulate environments of any dimension, since small areas to entire cities. The presented method involves the so-called "Crowd Patches", small pieces of precomputed animations which greatly reduces the resources needed to animate virtual humans. So, instead to pre-compute the simulation for all environment, each crowd patch is pre-computed separately. The animations of each crowd patch are cyclic over time and can move to neighbor patches. Two test environments were used: an "infinite" street and a city environment. The authors claim that their method delivers a good trade-off between memory usage, computation needs and motion quality, since the crowd patches can handle with large-scale environments while populating them with credible movement.

In BioCrowds [10], this subdivision is generated by the competition for space among agents, where if a marker belongs to one agent, it must not belong to any other agent but this one. Plus, agents do not see each other, instead, they can only view the markers placed around the scenario, which represents the free space where agents can walk. These markers are a reinterpretation of auxins, which occupy free spaces and stimulate the growth of ribs in the leaf blades, as hypothesized by Sachs [50] and implemented in the geometric model of Runions et al. [49].

According to Runions [49], the most acceptable theory for the veining patterns in plant leaves is the hypothesis of channeling of Sachs [50], which defines that the development of the leaf ribs is controlled by a signal scattered on the leaf blade, from a plant hormone called "auxin". The auxins positions are calculated using an adapted version of the Dart-Throwing algorithm [8, 42], where points are randomly generated in a uniform distribution, since the distance between the new point and its neighbors is less than a threshold. Although it is a computationally expensive algorithm, it was adapted to be used with each model iteration.

The auxins present in leaf blade determine the direction of the rib nodes growth and remain until removed due to the proximity of ribs grown towards them. Each auxin influences the rib node growth closest to it; if there are more than one node at the same distance, it randomly chooses one of them. Figure 2.5 shows how this process works, with auxins in red filled circles and nodes in black empty circles.

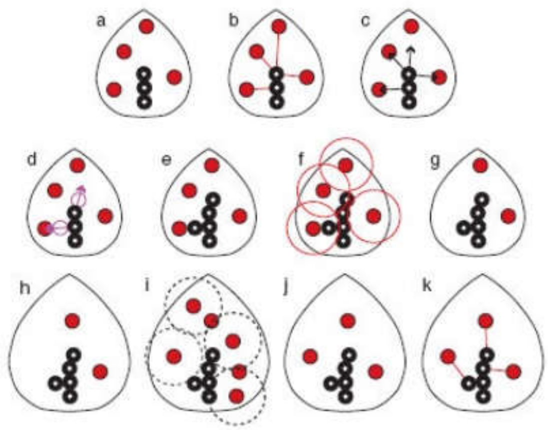


Figure 2.5 – Illustration of algorithm execution for the node pattern generation in leaves. Auxins are the red filled circles and nodes are the black empty circles. [49]

BioCrowds is based on Runions spatial colonization algorithm. While in the modeling of biological patterns ribs can be seen as paths generated by its ends to penetrate a free space toward auxin, in crowd simulation the displacement of the ends can be identified as the movement of agents in a given scenario. To perform this adaptation, some changes are proposed by Bicho[9]:

- **Restricting auxin space**: only auxins contained in the agent's personal space can influence it.
- **Auxins Persistence**: auxins are kept in the virtual environment throughout the simulation, but are available only to the nearest agent. This distance calculation is updated every iteration.
- **Goal Seeking**: Besides being influenced by auxin, the movement of people is also influenced by the willingness of each individual to reach a particular destination.
- **Speed Adaptation**: agents vary their speed according to space availability.

To calculate the agent's motion, it must find its next position. For this, first we compute all the vectors d between the agent and its auxins (closest to this agent than any other):

$$\boldsymbol{d_k} = \boldsymbol{a_k} - \boldsymbol{p}, \tag{2.3}$$

where $\boldsymbol{p}$ is the agent's position and $\boldsymbol{a_k}$ represents the position of the auxins. The set S' is defined by all vectors between the agent and its auxins, as in Equation 2.4.

$$S' = \{\boldsymbol{d_1}, \boldsymbol{d_2}, ..., \boldsymbol{d}_n\}. \tag{2.4}$$

In leaf ribs pattern, these vectors are normalized and summed, resulting in a mean vector. Since agent's goal must be considered, each direction vector of the model receives a weight, calculated and normalized according to its alignment with the objective. This weight shall achieve its maximum value when the alignment angle with the goal is 0° and its minimum value when the angle is 180°, decreasing as the angle increases. Therefore, the motion vectors can be defined as:

$$\boldsymbol{m} = \sum_{k=1}^{N} w_k \boldsymbol{d_k}, \tag{2.5}$$

where the weights w can be found by:

$$w_k = \frac{f(\boldsymbol{g}, \boldsymbol{d_k})}{\sum_{l=1}^{N} f(\boldsymbol{g}, \boldsymbol{d_l})}. \tag{2.6}$$

If the distances $\boldsymbol{d_k}$ increases for marker k, it will have relatively lower weights, preventing a possible dominance of these markers in the calculation of motion vector m. One possible choice for f that satisfies this model would be:

$$f(\boldsymbol{g}, \boldsymbol{d_k}) = \frac{1 + cos\theta}{1 + ||\boldsymbol{d_k}||} = \frac{1}{1 + ||\boldsymbol{d_k}||}(1 + \frac{< \boldsymbol{g}, \boldsymbol{d_k} >}{||\boldsymbol{g}||||\boldsymbol{d_k}||}), \tag{2.7}$$

if $||\boldsymbol{d_k}|| > 0$ where $< \boldsymbol{g}, \boldsymbol{d_k} >$ represents the internal product. If $||\boldsymbol{d_k}|| = 0$, $f(\boldsymbol{g}, \boldsymbol{d_k})$ assumes the value 0.

Finally, the model should allow the agent to move with a maximum desired speed S_{max}. However, in dense crowds, the space available for each agent is smaller, resulting in a speed reduction. Therefore, in the proposed model, the instantaneous motion vector can be defined by:

$$\boldsymbol{v} = s_{min} \frac{\boldsymbol{m}}{||\boldsymbol{m}||}, \tag{2.8}$$

where $s_{min} = min(||\boldsymbol{m}||, s'_{max})$. It implies that if $||\boldsymbol{m}|| > s'_{max}$, the maximum displacement is limited by s'_{max}. Otherwise, it is given by $||\boldsymbol{m}||$. Figure 2.6 shows a preview of the free-collision motion with infinitesimal agents proposed by the model.

2.2.3 Velocity-based

A velocity-based method implies that each agent in the environment can observe the velocity of each other agent and take them into account in order to avoid collision with the other agents. So, each agent changes its velocity keeping it inside a *velocity space*, where areas marked as forbidden can not be selected, since these velocities would generate collisions with other agents.

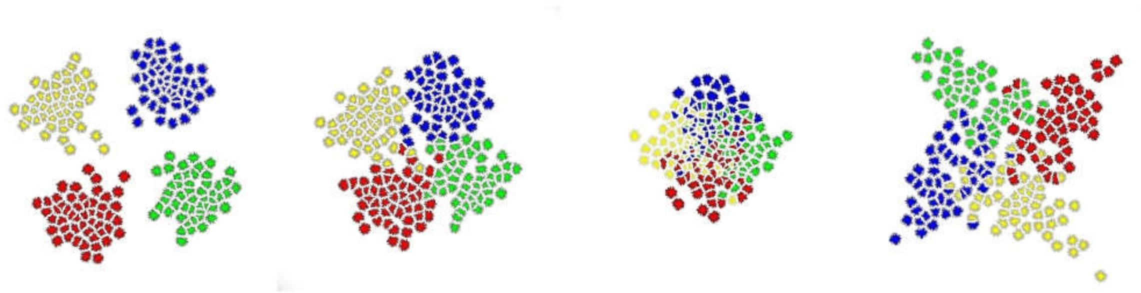


Figure 2.6 – A preview of the free-collision motion with infinitesimal agents. It is one of the results obtained by Biocrowds model. [9]

The Optimal Reciprocal Collision Avoidance (ORCA) is a velocity-based method, proposed by Van den Berg et.al [56], for collision avoidance between multiple agents. It was developed for the robotics industry to make multiple robots avoid collision between them. One of the most important advantages is that it does not need a central control to maintain agents free of collisions, since each agent can perceive other agents positions and velocities in order to define its own velocity. Therefore, this method runs at $O(n)$.

In ORCA method, agents are defined as circle shapes and can move to any direction. To find a possible velocity for an agent α, a velocity obstacle (VO) function between α and other agents β is used, which can be calculated by the following equation:

$$VO^T_{\alpha|\beta} = \{\boldsymbol{v} | \exists t \in [0,T] :: t\boldsymbol{v} \in D(\boldsymbol{p}_\beta - \boldsymbol{p}_\alpha, r_\alpha + r_\beta)\}, \tag{2.9}$$

where $\boldsymbol{v}$ is the relative velocity of α regarding β, r_α is the radius of agent α, $\boldsymbol{p}_\alpha$ is the center position of agent α and $D(\boldsymbol{p}, r)$ is the agent disc with radius r and centered at position p. In short, VO has all $\boldsymbol{v}$ which will result in a collision between α and β in a T interval of time. Figure 2.7 gives a visual example of how it works, with robots A and B, where a) shows an initial configuration of them and b) shows the VO of A regarding: all velocities of A inside the gray area will most likely cause a collision with B.

Then, for any set of velocities V_β, if it is true that $\boldsymbol{v}_\beta \in V_\beta$ and $\boldsymbol{v}_\alpha \notin VO^T_{\alpha|\beta} \oplus V_\beta$, it holds that α and β are collision free for at least T time. It forms the set of collision avoidance velocities, denoted by:

$$CA^T_{\alpha|\beta}(V_\beta) = \{\boldsymbol{v} | \boldsymbol{v} \notin VO^T_{\alpha|\beta} \oplus V_\beta\}, \tag{2.10}$$

where $\oplus$ stands for the *Minkowski* sum of two sets. In short, it is the sum of the elements of the two original sets, at the same index level.

Finally, if $V_\alpha \subseteq CA^T_{\alpha|\beta}(V_\beta)$ and $V_\beta \subseteq CA^T_{\beta|\alpha}(V_\alpha)$, it can be said that they are sets of reciprocally collision avoidance velocities. So, the optimal reciprocal collision avoidance velocity is

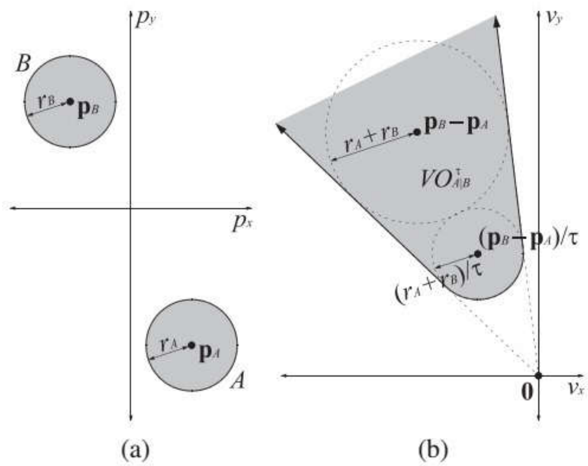


Figure 2.7 – (a) Initial configuration of two robots A and B. (b) The velocity obstacle of A regarding B, represented by the cone with origin at A's position. It can be seen that all velocities of A inside the gray area will most likely cause a collision with B. [56]

the pair that increases the number of velocities closer to the optimum for each agent. This can be achieved following:

$$ORCA^{T}_{\alpha|\beta}(V_\beta) = \{\boldsymbol{v} | (\boldsymbol{v} - (\boldsymbol{v}^{opt}_{\alpha} + \boldsymbol{u}/2))\boldsymbol{n} \geq 0\}, \tag{2.11}$$

where $\boldsymbol{v}^{opt}_{\alpha}$ is the optimization velocity for agent α, $\boldsymbol{u}$ is the vector between $\boldsymbol{v}^{opt}_{\alpha} - \boldsymbol{v}^{opt}_{\beta}$ to the closest point on the boundary of the velocity obstacle and $\boldsymbol{n}$ is the outward normal of the $VO^{T}_{\alpha|\beta}$ boundary at point $(\boldsymbol{v}^{opt}_{\alpha} - \boldsymbol{v}^{opt}_{\beta}) + \boldsymbol{u}$. Since ORCA takes into account both α to β and β to α, only half of $\boldsymbol{u}$ is considered, which means that each agent takes half responsibility to avoid collision between each other. It is commented that choosing the current velocity as the optimization velocity gives the ideal trade-off between zero and preferred velocity, since it adapts to the situation automatically. Figure 2.8 shows an example simulation of 1000 agents evacuating an office.

Figure 2.8 – Example simulation of 1000 agents evacuating an office [56]

2.3 Population Cultural Aspects

Hofstede [32] define culture as "... the collective programming of the mind that distinguishes the members of one group or category of people from others". It is a collective phenomenon where a group (for example, a nation) has striking features which differs from another group, evidenced in each individual inside this group. This kind of behavior can be extended to crowd simulation. It shall be possible to define some expected behaviors in a group of agents and make them follow these rules while part of it. This section presents some studies on cultural dimensions in different aspects: we present the model of Hofstede that proposes six cultural dimensions to characterize a nation, two models that use cultural characteristics in games and crowd simulations, and finally a method that describes the possibility of detect cultural aspects in video sequences.

2.3.1 Hofstede Cultural Dimensions

Hofstede's Cultural Dimensions (HCD)[33] was proposed by Geert Hofstede as a framework for cross-cultural communication. It aims to describe the effects of a society's culture based on the values of its individuals, as well how these values can influence on how people behave. For this end, the author conducted a large survey across the world to find the difference of national values in subsidiaries of IBM. Then, he compared the answers of 117,000 employees samples of the corporation. The goal was to find out data about National Culture, which is related with the difference of values that can be found between groups of nations/regions/societies. For this, six cultural dimensions were defined as percentages values, as follows:

- Power Distance Index (PDI): defined as "the extent to which the less powerful members of organizations and institutions accept and expect that power is distributed unequally".
- Masculinity vs. Femininity (MAS): defined as "a preference for cooperation, caring for the weak and quality of life.", which is denoted by the femininity (high degree) aspect (i.e. women tend to be more cooperative).
- Long-term orientation vs. short-term orientation (LTO): a long term orientation (high degree) tends to take a pragmatic approach, while a short term orientation tends to view societal change with suspicion.
- Indulgence vs. restraint (ING): In short, it is a measure of happiness, where indulgent groups believe to be in control of their own lives.
- Uncertainty Avoidance (UNC): indicates how much the culture feel either uncomfortable or comfortable in novel or unknown situations. A weak UNC level feels comfortable with ambiguity and chaos, while a high level has need for clarity and structure.

- Individualism (IND): Deals with the individualism/collectivism of the culture. Individualist cultures tend to have weak ties between individuals, while collectivist cultures have individuals who look after each other.

Some of this cultural dimensions are going to be used in order to calculate our group parameters, like cohesion and desired speed. In fact, the work of Lala [37] already makes use of such dimensions and is going to be presented in the Section 2.3.2.

2.3.2 Game and Crowd Simulations

The work of Mascarenhas et al. [39] is focused on representing general cultural biases of social conduct in a cognitive model for virtual agents, applying it in a virtual game named Traveller [11], which is centered around human-agent interaction in everyday situations with agents that have their own cultural values. The proposed model is centered around the idea of Social Importance (SI), which quantifies the extent that an agent is willing to act voluntarily in the interest of another. So, a group of agents has three components:

- SI Attribution Rules: each rule represents a specific relational factor that is associated in human culture to a gain or a loss of SI;

- SI Claims: associates a specific SI requirement to a given action, like joining a group to start a conversation; and

- SI Conferrals: represent a behavior that confers a certain amount of SI and is socially expected to be performed in a given context.

The experience performed by the authors consists of an interactive narrative that takes place in a fictitious world and is divided in several scenes. Each of the scenes is designed to evoke mismatches between the cultural biases of users and the cultural biases of agents. Since the user plays the role of a foreigner, agents will believe the user is an outsider. The higher is their collectivism, lower will be the SI attribute. The same applies to the user: he/she loses some SI when s/he performs an action that is perceived as an inappropriate claim in the culture of the agents. The main goal here was to compare users from a collective country, Portugal, and an individualistic country, the Netherlands. The results show that the proposed model is capable of adapt the agents' cultural behavior toward one extreme of the Individualism vs. Collectivism dimension.

Following similar hypothesis, the work of Lala et al. [37] develops a virtual environment which enables the creation of different types of cultural crowds in which the user may interact. For this, the cultural dimensions work of Hofstede [33] are used to describe a measure of culture which influence the behavior of societies. For each country, each of the five used dimensions (Indulgence (ING) was not used) receives a number score based on the results of questionnaires and surveys. In brief, the main goal is to utilize an environment with a simulated crowd from a cultural perspective.

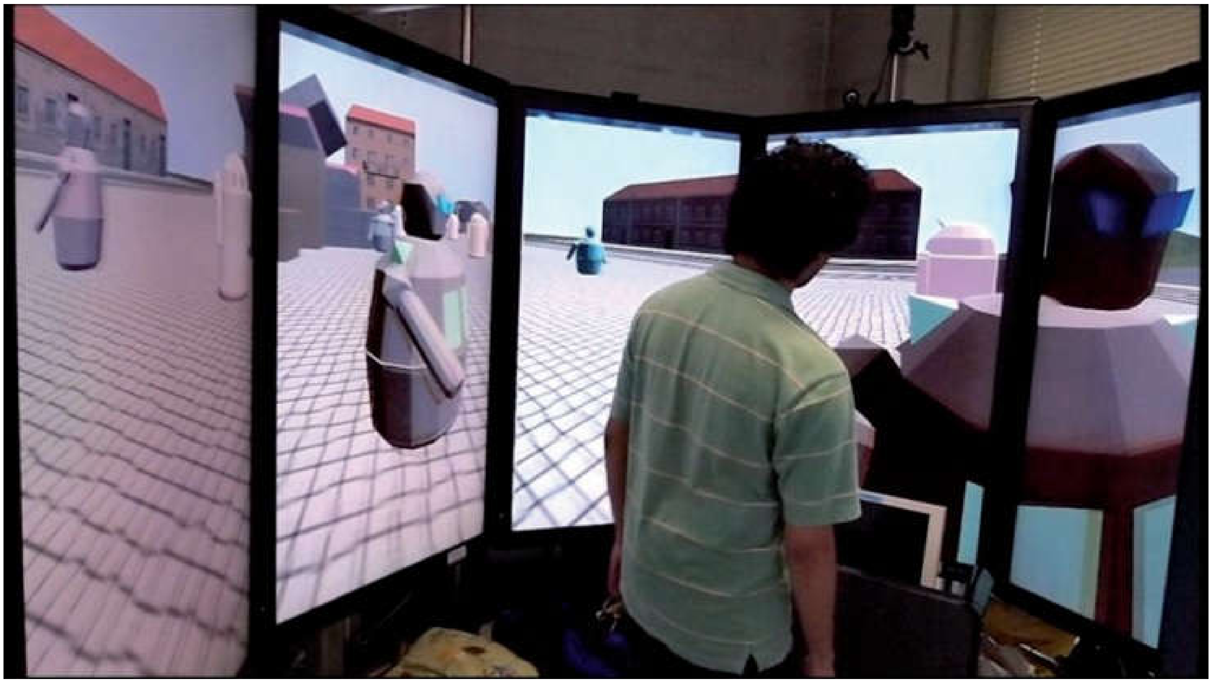


Figure 2.9 – User inside the immersive environment, where s/he may turn in any direction and observe the scene from that viewpoint. [37]

The proposed environment is a 360 degrees immersive one, where the user may turn in any direction and observe the scene from that viewpoint (Figure 2.9). Yet, the user may move around the environment and interact with the cultural virtual agents. The agents are robotic characters, and the primary reason for this is that there should be no preconceptions in regards to culture, once users should find out the culture norms based only on interactions and gestures. The most important characteristics of the agents are:

- Personal Space: a minimum distance between the agent and others around. The agent tries to preserve this space;
- Walking Speed: the speed the agent moves;
- Range of Vision: the maximum distance within the agent considers to try to avoid collision;
- Angle of Vision: represents peripheral vision of agents;
- Maximum Turning Angle: the maximum angle that the agent can change direction without affecting speed; and
- Collision Avoidance Behavior: the agent tries to avoid collision with other agents or obstacles.

With the data gathered from the tests, it was observed that agents in the individualist crowd were faster on average than those in the collectivist crowd for a crowded environment, but there was also more likelihood of individualist agents having to wait more for space to move, once individualist agents have a higher value for personal space. The authors conclude that there is a reasonable difference between an individualist crowd of agents and a collectivist one, especially in how their collision avoidance methods and their different need for space that can influence in how they navigate.

2.3.3 Detecting Cultural Aspects in Video

Favaretto et al. [18] present a method to detect cultural aspects in groups of people, in video sequences. Using computer vision to detect groups, it is proposed to map some observed characteristics of people such as speed, distance between them and occupied space, to Hofstede's Cultural Dimensions (HCD) [33] such as power distance (PDI), masculinity/femininity (MAS) and long/short-term orientation (LTO/STO).

The method is able to identify temporary and permanent group of people, the latter been defined if it keeps a group structure for more than 10 % of the total frames of the video. Results show that their defined equations to map culturality seem to be coherent with psychological literature (for more details please refer to [18]). Figure 2.10 shows a cultural comparison among the three studied Countries: Brazil, China and Austria. As can be seen in this graph, Brazil presented lower individualism level, while Austria presented the higher one. The results were also compared to the values obtained in HCD.

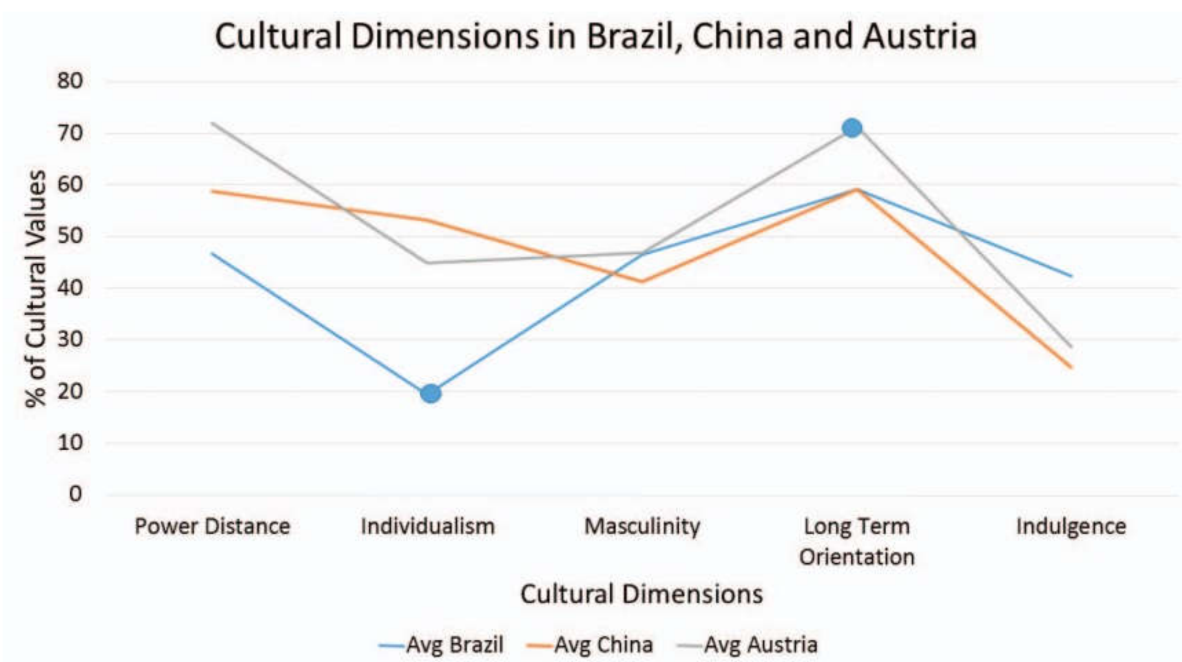


Figure 2.10 – Cultural comparison among Brazil, China and Austria [18]

2.4 Personality traits and emotions

This section presents some simulation methods developed using personality traits and emotion contagion. Firstly, Durupinar's model [13] is explained. The authors proposed a method to represent emotion and emotion contagion based on psychological traits. Then, the approach of Guy [23] is discussed, which tries to relate the personality traits of individuals with the crowd behavior. Finally, the model of Carretero [3] is presented, where they focus on mapping emotional states and contagion in agents through full-body expression, as well the work of Neto et al. [45] where they aim to give emotional contagion ability to virtual agents in the crowd.

2.4.1 Durupinar methodology

Durupinar et al. [13] developed a simulation model based on psychological traits which aims to represent emotions and emotion contagion between agents in an effective way. To this end, the OCEAN (Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism) psychological traits model, proposed by Goldberg [22] is used. In short, the agent's personality, its perception of other agents and the environment can change the agent emotion, which can lead to a behavior change. Figure 2.11 shows the proposed Durupinar's framework.

Due to the orthogonal nature of the OCEAN model, it is possible to make a direct association between its traits and agents characteristics, like walking speed, pushing and agent radius. Figure 2.12 shows a relationship between these agents characteristics and OCEAN traits. About emotion, agents emotional state are defined as a combinations of its appraisal and emotion contagion with other agents.

For appraisal, the OCC (Ortony, Clore, Collins) model [46] is applied. It suggests that an individual emotion has a positive or negative reaction to its goals, standards and attitudes. With these three stimuli, the model builds a hierarchy with 22 possible emotions, like fear, hope, joy, hate, among others.

Concerning the emotion contagion, a generalized model is adopted following the approach proposed in the work of Dodds and Watts [12]. It is a threshold model, where the probability to get infected increases as the individual becomes exposed to more infected agents. Therefore, an individual can be in two states: susceptible or infected. Following this idea, if the emotion value of an agent surpasses the threshold, it becomes infected and affected by that emotion, having its value summed with the contracted emotion one. If this value surpasses the expressiveness threshold, then this agent starts to spread this emotion to others.

Since agents can experience different emotions at the same time, the PAD (Pleasure, Arousal, Dominance) model, proposed by Mehrabian [41], is used to determine the average emotional state that an agent is experiencing. Pleasure defines the predominance of positive over

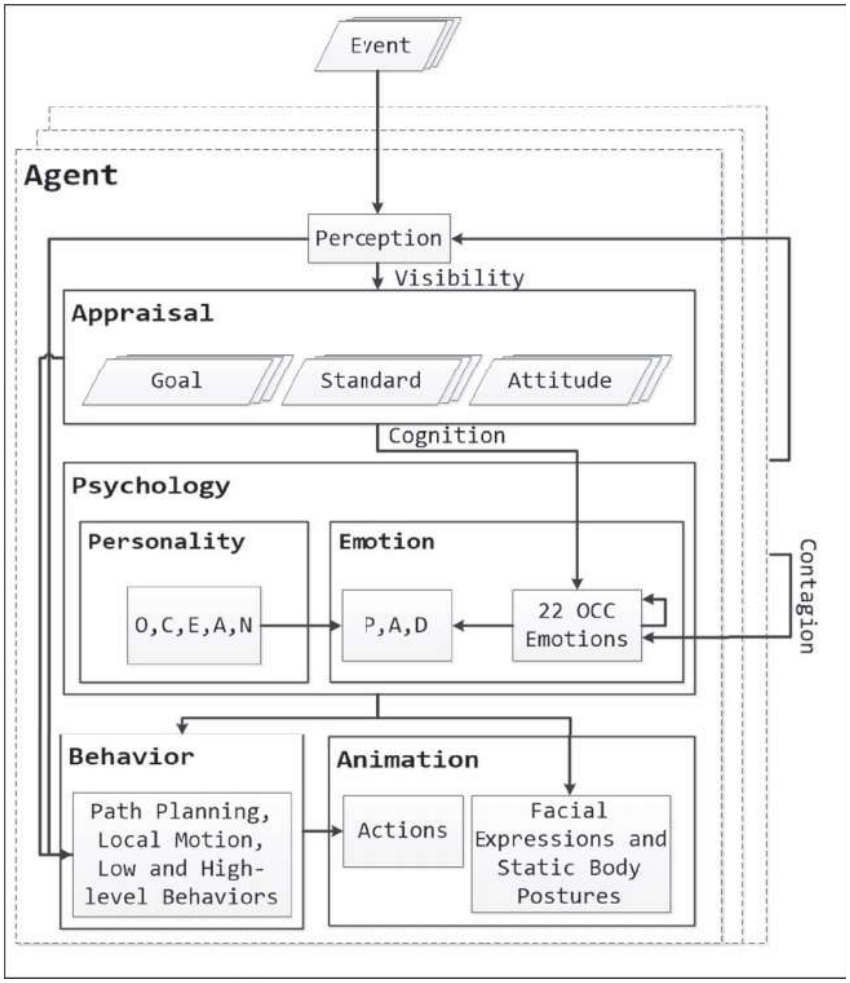


Figure 2.11 – Durupinar's framework. The appraisal of the agent about its environment, along with its personality, can trigger different emotions and behaviors.[13]

Agent behavior	Personality factor	Ocean factor*
Leadership	Assertive, social, unsocial, calm, fearful	E, N
Trained or untrained	Informed, ignorant	O
Communication	Social, unsocial	E
Panic	Oversensitive, fearful, calm, orderly, predictable	N, C+
Impatience	Rude, assertive, patient, stubborn, tolerant, orderly	E+, C, A
Pushing	Rude, kind, harsh, assertive, shy	A, E
Right preference	Cooperative, predictable, negative, contrary, changeable	A, C
Avoidance or personal space	Social, distant	E
Waiting radius	Tolerant, patient, negative	A
Waiting timer	Kind, patient, negative	A
Exploring environment	Curious, narrow	O
Walking speed	Energetic, lethargic, vigorless	E
Gesturing	Social, unsocial, shy, energetic, lethargic	E

The letters in this column stand for openness, conscientiousness, extroversion, agreeableness, and neuroticism.

Figure 2.12 – Relationship between low level parameters and OCEAN traits [14].

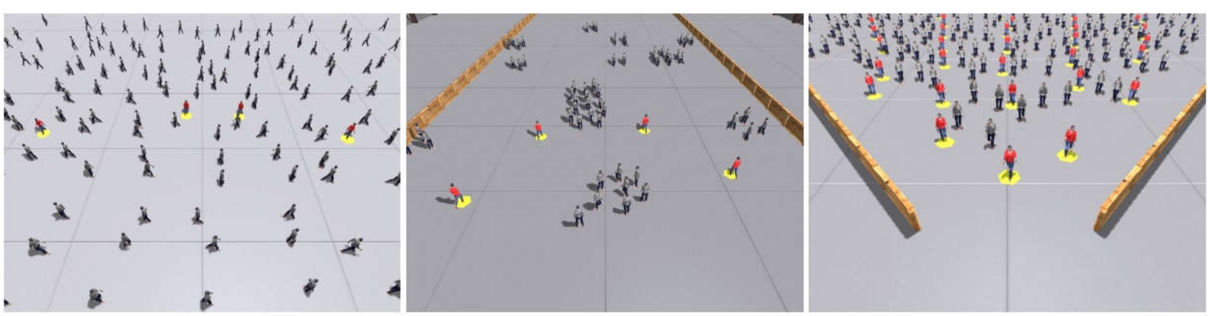
negative states. Arousal shows how easily an agent can be incited. Dominance relates the feeling an individual has to be in control of its life situations with the feeling of being controlled or influenced.

2.4.2 Guy methodology

Guy et al. [23] tackle the problem of generating heterogeneous crowd behavior altering the simulation parameters to mimic personality traits of individuals, in order to be able to evaluate how it can affect the crowd behavior as a whole. To define this personality variation, they use the Eysenck 3-Factor personality model [16], known as PEN, which is based on three personality factors:

- Psychoticism: measures an individual level of aggression and egocentricity.
- Extraversion: measures an individual level of social interest.
- Neuroticism: measures an individual level of emotional instability.

The authors conducted an user study in order to map these personality traits to simulation parameters. To do so, 40 participants were selected, where each one of them was asked to watch 18 videos (six for each of the three generated scenarios) randomly selected from a previously generated pool composed of approximately 100 videos. An example of each scenario is showed in Figure 2.13. In the Pass-Through scenario (a), four individuals move through the crowd composed of 400 agents. In the Hallway scenario (b), four individuals move through a hallway passing in middle of 66 agents, which are still and divided in several small groups. Finally, in the Narrowing Passage scenario (c), 40 individuals move alongside the other 160 agents trying to reach a narrow exit at the bottom.



(a) Pass-Through Scenario (b) Hallway Scenario (c) Narrowing Passage Scenario

Figure 2.13 – a) Four individuals move through the crowd composed of 400 agents. b) Four individuals move through a hallway passing in middle of 66 grouped agents. c) 40 individuals move alongside the other 160 agents trying to reach a narrow exit. [23].

The highlighted individuals were always wearing a red shirt and with a yellow circle under them. The participants were asked to rate them in comparison with the rest of the crowd, as being more or less "Aggressive", "Shy", "Assertive", "Tense", "Impulsive" and "Active". Using these results, the authors were able to derive a linear model in order to map the traits to simulation parameters. Then, another user study is conducted to validate the model, trying to understand how well it performed at producing simulations with the expected behavior. For this, another 19 participants were chosen and new videos were recorded. The results suggests that the chosen adjectives "Aggressive" and "Impulsive" presented a closely behavior, which was expected due to their high correlation found at the first user study. Finally, the authors conclude that their method could generate simulations where agents visibly appear to have different personality traits, as defined by PEN.

2.4.3 Carretero Methodology

Carretero et al. [3] present a simple model for emotional crowd. Their work uses 3D androgynous mannequin as agents and focus on full-body expression. Concerning this, they map three different states: neutral, happy and sad. They use Unity's animation system to create the transition state machine between the expressions.

To define the emotional state, an interval between -1 and 1 is used, where -1 represents sad, 0 represents neutral and 1 represents happy. The change in the mood of an agent emerges from its perception of other agents, therefore, each agent is able to perceive and react according emotions demonstrated by other agents. For this, three modules were created:

- Perception Module: with a defined field of view for each agent, a perception occurs when, for example, agent B enters into the field of view of agent A. So, agent A is aware of the emotional state of agent B.
- Appraisal Module: once agent A is aware of the emotional state of agent B, agent A checks if there is a possibility for an emotion contagion.
- Contagion Module: if it is possible to agent A to be infected, this stage deals with the change of mood for agent A.

Figure 2.14 shows idle (left) and walking (right) agents for both happy (top), neutral (middle) and sad (bottom) states, where the colors reflect the emotion too. Although the simplicity of this model, the authors conclude that it can be used as a basis for complexer approaches. Several possible future works are cited, like to consider different emotional states (like anger or panic), to use a more complex contagion model and to represent the emotional scale in a different way, avoiding to use discrete values.

2.4.4 Neto Methodology

Neto's work [45] aims to introduce a model to give agents the ability of emotional contagion in crowd simulations. His motivation is that, whichever the application of a crowd simulator, the

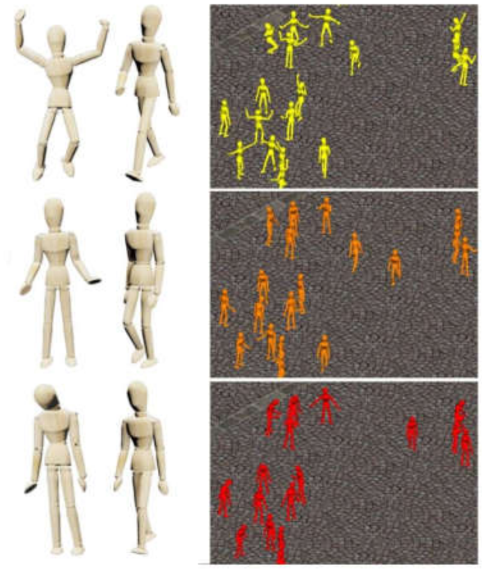


Figure 2.14 – Idle (left) and walking (right) agents for both happy (top), neutral (middle) and sad (bottom) states. Colors are generated automatically according the emotional state. [3]

simulation should be the most realistic as possible in order to obtain the most trustworthy result, especially when this result is going to be used as a basis for critical decisions. His work, also, uses BioCrowds to simulate agents in a given scenario.

For the emotion contagion, Bosse's model[2] is used, where a model for emotion contagion between agents is proposed. In fact, Neto's work extrapolates Bosse's model, considering more than one emotion per agent. Therefore, a happy agent may have a high expressiveness for happiness and a low expressiveness for sadness. Also, each agent may be more or less susceptible to each emotion, which means that a high susceptibility will make an agent more responsible to this given emotion. Finally, the relationship between agents is taken into consideration, being a function of attachment and distance between them.

Some scenarios were tested, as with standing agents, agents moving in the same direction and agents moving in a counterflow scenario. Their results show that the emotion contagion worked as intended, even when agents were still. Some groups were observed to not be affected by the contagion, but that was due to the limitation of distance imposed by the model. Also, connecting emotions with goals made agents change its objectives when their emotional state shifted.

2.5 Thermal models

Besides all works in literature presented in this section, we also include some pioneer method to simulate crowd comfort based on thermal comfort. Thermal comfort denotes how comfortable an agent is in a given place which has a given set of climatic parameters. In other words, it can be seen as how comfortable a person is inside a room with an air conditioner at 20 degrees, or at the street exposed to the sun in a temperature of 30 degrees, humidity of the air at 60% and wearing a coat.

This section presents some concepts concerning thermal comfort models. Firstly, Fanger's work [17] is explained, where he develops the Predicted Mean Vote (PMV) to evaluate a person thermal comfort. Then, the approach of Chen et al. [7] is presented as a way to use Fanger's PMV in order to control agent's behavior in virtual humans simulation.

2.5.1 PMV and PPD

Although environment temperature may be a straight number, people usually feel different levels of thermal comfort. In order to evaluate this thermal comfort, Fanger [17] proposed the Predicted Mean Vote (PMV). In general, thermal comfort is impacted by six major factors, as discussed by [17]:

- Metabolic Rate (M): rate of transformation of chemical energy into heat.
- Clothing Insulation (I_{cl}): quantifies the thermal insulation provided by a specific piece of clothing.
- Air Temperature (t_a): the actual temperature of the air.
- Mean Radiant Temperature (t_r): the uniform temperature of an imaginary enclosure in which the radiant heat transfer from the human body is equal to the radiant heat transfer in the actual nonuniform enclosure.
- Air Speed (v): the velocity of the air in a given environment.
- Humidity (r_h): the humidity of the air.

PMV maps these six major factors into a numerical value between -3 and 3, which represents the thermal sensation of a given person. The desirables PMV values lie between -0.5 and 0.5, where negative values indicate a cold sensation and positive values indicate a hot sensation. In fact, PMV may be calculated following Equation 2.12:

$$PMV = (0.303e^{-0.036M} + 0.028)L(M, I_{cl}, t_a, t_r, v, r_h), \tag{2.12}$$

where L represents the thermal load of the body, defined as the difference between the heat generation and the heat loss. Details about this equation can be found in Fanger [17].

Plus, Fanger proposed the Predicted Percentage of Dissatisfied (PPD), which represents the expected percentage of people thermally uncomfortable. Plus, it can be seen as the chance of a person to feel thermally dissatisfied in a given environment. It can be related with the PMV value, following Equation 2.13:

$$PPD = 100 - 95e^{-0.03353PMV^4 - 0.2179PMV^2}. \tag{2.13}$$

2.5.2 Crowd Simulation Incorporating Thermal Environments

Chen et al. [7] developed a thermal model which can be virtually integrated with any crowd simulator, based on Fanger's PMV and PPD [17]. In their work, agents select their motions and actions in order to improve their comfort level, using thermal and density comfort parameters.

An interesting aspect in their work is that agents are not only affected by the thermal conditions, but can also affect the environment transferring heat into it. In general, human heat generation values can vary from 81W (i.e. at sleeping state) to 1630W (i.e. at a running state).

Their results show plausible agent behaviors, both under varied density and thermal contexts. In a virtual environment, such observed emergent behaviors can improve the sense of presence of the individuals, since simulated agents will spontaneously change their position (or even clothing) in order to feel more comfortable in the environment.

2.6 The context of this work in the State-of-the-Art

Since the goal of this work is to be able to simulate behavioral diversity in crowds, many of the presented methods are going to be useful. As simulation model, Biocrowds [9] was chosen since it is a collision-free state-of-the-art method. Concerning cultural aspects of the crowd, the work of Hofstede [32] is going to be useful whereas it defines several cultural dimensions that can be mapped in group parameters, which can define group behaviors. Besides that, the work of Durupinar et al. [13] uses OCEAN psychological traits [22] to define emotional state and behaviors for their agents. It can be used in this work to relate the defined Durupinar's parameters with group parameters, in order to decide the behavior of agents. Finally, the work of Cheng [7] can be used in order to achieve the desired thermal comfort for agents. Gathering all these methodologies, we intend to re-parametrize BioCrowds with the main goal of providing heterogeneous agents that behave with diversity.

In Section 3 we present the model proposed in this work.

3. PROPOSED MODEL

The main proposal is to extend Biocrowds [9] model, by implementing new features and new parameterization, in order to achieve the behavioral diversity aspects, providing more realistic simulation in the virtual world.

When we think about providing models to endow virtual agents with characteristics that allow the behavioral diversity, we also consider the virtual environment and the way people evolve in such spaces. BioCrowds, as the major part of existent crowd simulators, is a goal based method. It means that virtual agents should "appear" in the simulation knowing their goals and their coordinates in the environment. Having in mind that we wanted to provide diversity of behaviors, we decided to include also an exploratory behavior in the space where agents can be "influenced" by the space and then, even with the same goals, behave in a different way.

Firstly, to achieve the exploratory navigation, we propose to make some changes concerning the goal seeking behavior. In short, instead to have just one goal to achieve, each agent should have a list of desired goals and an intention value for each one of them, representing its willingness to reach them (this factor could be connected with agent personality in a future work). Yet, the exploratory behavior should mimic situations where people do not know where places are located (for example, an agent may be hungry and willing to go to a restaurant, but does not know where it is). Still considering the environment, thermal cells are added in the environment in order to calculate the thermal comfort of the agents, following the proposals of Cheng et al [7] and Fanger [17].

Regarding the goal to provide behavioral diversity in agents according to cultural and personality aspects, we propose to re-parametrize BioCrowds as follows:

- to define some group parameters (for example, cohesion and desired speed) which will guide agents behaviors. These parameters are defined following mentioned models/theories, like Hofstede cultural dimensions [33]; and
- to consider as input the OCEAN [22] of each agent and compute individual and groups characterization, as well as motion parameters (goals, speed, etc). To do so, we use Durupinar model [13], which work mapped OCEAN factors into agent's behavior.

Section 3.1 presents an overview of the proposed method.

3.1 Overview of the Method

Figure 3.1 presents an overview of the method. At the image, it can be seen the three major crowd simulation entities that are going to be tackled in this work: Environment, Groups and Agents. It is important to notice that, as commented in Section 2.2.2, agents in BioCrowds do not see other agents, neither obstacles in the way. All they can perceive are the markers, which guide their movement. Such behavior is maintained in our work, where we merely add new behaviors and attributes which aim to enhance the complexity of how agents behave in the simulations. Section 3.2 details the changes made in the environment, which includes the addition of signs in the environment to help the guidance of the agents and heat/sinks sources which are going to affect the thermal comfort of the agents. Section 3.3 presents the proposed method to deal with groups in the simulation, where each one of them have some parameters which help to guide their behavior. Although these group parameters, which are going to be discussed in Section 3.3.1, can be statically set, it is proposed to define them as function of cultural/psychological methods, like Hofstede Cultural Dimensions [32]. Finally, Section 3.4 presents behaviors and attributes which are going to control the actions of our agents, like exploratory behavior (i.e. meaning they do not know where their goals are located) and comfort.

3.2 Environment

This section details the changes made in the environment. Section 3.2.1 shows how the signs are able to help the guidance of the agents, as well how such signs can communicate with agents and what are the heat/sink sources introduced in the environment.

3.2.1 Signs

As briefly commented in Section 3.1, agents may not know where their goals are located, therefore, they can just wander around the environment for a long time without actually achieving their objective. To indicate the way to go, it is proposed to place signs in the environment. These signs have a goal to which they refer to and an appeal value indicating how much it may induce agents. So, a sign with high appeal could be a big outdoor and another sign with low appeal could be a small poster. Plus, each agent should have a value indicating how much any sign can change its intentions, i,e, how much the agent really wants to go this goal. To do so, a susceptibility value is added to each agent for each sign. All of this three new values - intention and susceptibility (for agents) and appeal (for signs) - are normalized to rely in the interval $[0; 1]$. Finally, since agents may start the simulation with no knowledge about where their goals are located, a temporary state is added on them, implying that they will just wander around randomly until finding some orientation sign. We called this as state Looking_For (LF). An agent will find a sign if it is inside the agent perception radius parameter (empirically defined as 5 meters, but can be easily altered) and this affects its intentions according to the following equation:

$$\gamma_{SA} = \epsilon_S \alpha_{SA} \delta_A, \tag{3.1}$$

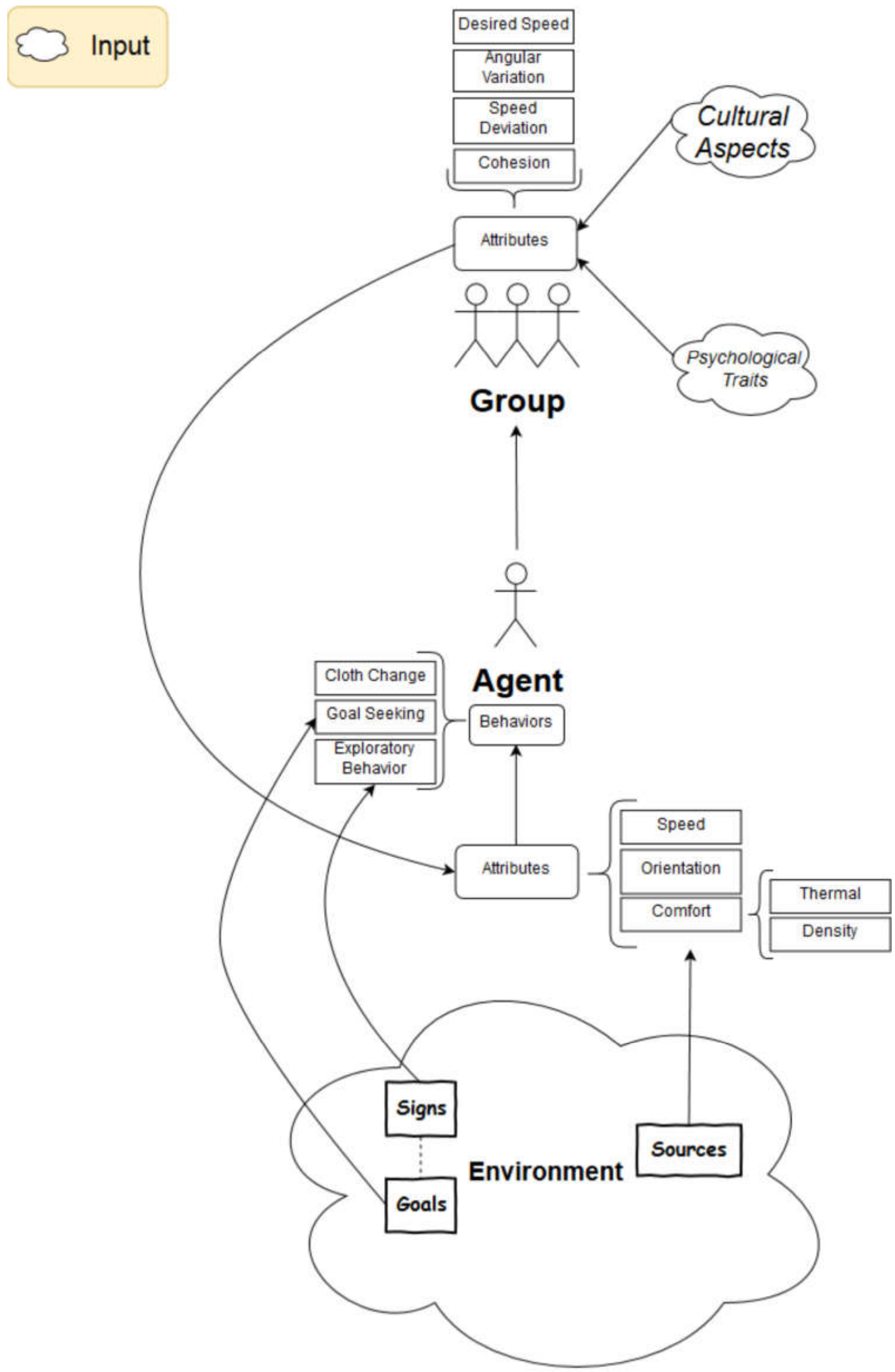


Figure 3.1 – Overview of the method.

where γ_{SA} is the total intention force between the sign and the agent, ϵ_S is the sign's appeal, α_{SA} represents the interaction between the sign and the agent, which is inversely proportional to the distance (the farther away, the less interaction), and δ_A is the agent's susceptibility to that goal.

Having the intention force, it is possible to calculate the intention variation just by multiplying it by the reversed original intention. Finally, this variation is added to the original intention I, resulting in the new intention $I*$:

$$I* = I + (\gamma_{SA}(1 - I)). \tag{3.2}$$

This mathematical model was inspired in Bosse's method [2], which deals with emotion contagion between agents. In this work, instead to deal with emotion, it is used to provide a way of simulating the impact the visual information in the environment (e.g. signs) causes in the agents.

Each agent A will start the simulation with a G_A number of goals it wants to visit and a G_A number of intentions I_A, which values are in the interval $[0; 0.8)$. The state LF is then added, as intention 0 (i.e. the first one) for all agents $I_{A,0} = 0.8$. Indeed it is initialized as a value that makes it the highest intention value for agent A and, consequently, the first "objective" to follow. While agents wander around, if a sign come into their perception spaces, the Equation 3.2 is applied and their new intentions are calculated. If one intention value exceeds the threshold of state LF (0.8), this goal becomes the new target and the agent starts to move towards it. To avoid agents wandering around forever (i.e. in a scenario with no signs), a sign is by default placed at each goal location, pointing to it. Results about this behavior are presented in Section 4.1.

3.2.2 Comfort Sources

In this work, we define sources as something which may affect the comfort of agents in the simulation. As explained later in Section 3.4.2, we are working with two types of comfort: thermal and density. Thermal is related with the ambient temperature and, therefore, is related with, among other factors, thermal sources in the environment. Density is related with the quantity of agents inside a given area and, therefore, its source is related with the amount of other agents around a given agent. More details about comfort are going to be given in Section 3.4.2.

3.3 Groups

This section presents the proposed method to deal with groups in the simulation, where each one of them can have some attributes which help to guide its behaviors. Although these parameters can be statically set, it is proposed to define them as function of cultural/psychological methods, like Hofstede Cultural Dimensions (HCD) [32]. Section 3.3.1 depicts our defined group parameters, which are going to be used to determine how our groups and agents are going to behave. Then, Section 3.3.2 shows the proposed mapping from HCD to our group parameters, while Section 3.3.3 shows the proposed mapping from Durupinar behaviors to our group parameters.

3.3.1 Groups Attributes

One important structure in crowds are the groups. Still more relevant if we are thinking about cultural and personality aspects. There are some literature in this area such as the work of Fridman et al. [20] which discuss cultural parameters and analyzes videos of five different countries. For this, some crowd parameters are studied, like speed, personal space and types of grouping. Yet, Knowles et al. [35] define that a group consists of two or more people interacting for an amount of time. In its turn, McDavid [40] defines a group as a social unit which has a number of individuals and a system of organization. Such individuals have relationships or bonds between each other, which keep them together and guide their behavior.

Therefore, groups in real life are not the same: they can have different goals to achieve, intentions, velocities, group cohesions, etc. For example, a group may be formed by intimate friends, with people staying close to each other and trying to stay together, even in a free space. On the other hand, another group may be formed by people sharing a temporary common goal. This people will probable try to keep larger distances with others, if compared to intimate distances, but maybe they will not achieve that due to the lack of space around. However, if they can maintain a comfortable space for each one, they will try.

In order to be able to simulate such types of behaviors, we propose to add some information in the group level. Firstly, it is important to mention that the concept of groups already exists in BioCrowds, which is the chosen tool to simulate crowds. In original BioCrowds, this concept is implemented just assigning the very same goal for agents which should be in the same group.

Following we discuss some aspects existent in groups of people. The work of Dyaram et al. [15] aims to research and explain the factors which construct the cohesiveness of a group. By their definition, members of a strongly cohesive group tend to stay together, it means do not leave the group, as well to be an active part of it, participating diligently of group's activities.

In our method, a cohesion value ζ_g is set to define how much a group g tends to stay together, in the interval [0,3], where 0 is the lowest cohesion value and 3 is the highest. This interval is defined according to the work of Favaretto et al. [18]. Further, a cohesion distance value μ_g is defined to represent the maximum distance an agent can be away from the rest of the group g, without leaving it. This cohesion distance is calculated as follows in Equation 3.3:

$$\mu_g = Hs - (\zeta_g(\frac{Hs - Hp}{MC})), \tag{3.3}$$

where Hp is the Hall's personal space and Hs is the Hall's social space. This distance spaces are described by Hall [24] which defines regions, called by the author "proxemics", that a person tends to maintain to feel comfortable. Figure 3.2 shows the Hall's interpersonal distances, in feet and meters. The MC value stands for Maximum Cohesion and represents the maximum cohesion value a group can achieve (in this work it was used 3 for this value). For instance, if $\zeta_g = 0$ for a certain group g then $\mu_g = 3.6$, i.e. this group with low cohesion value can have the members more spread.

On the other hand, if $\zeta_g = 3$ then $\mu_g = 1.2$, meaning that members stay close to each other in order to be a group, since they have a strong connection and are more attracted.

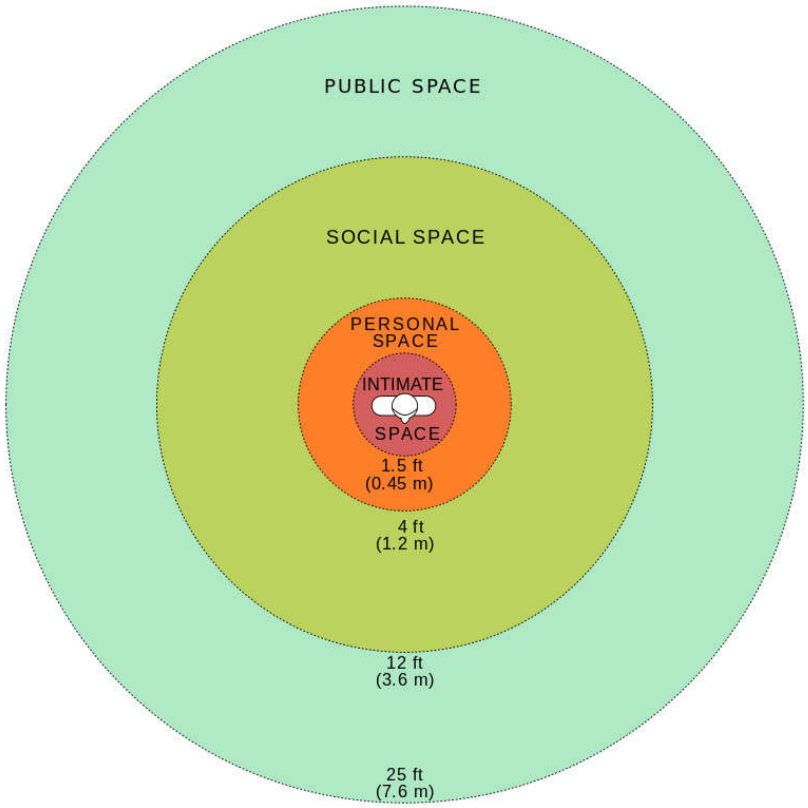


Figure 3.2 – Hall's interpersonal distances, in feet and meters. Available at https://en.wikipedia.org/wiki/Proxemics

If an agent gets farther from the rest of the group than the cohesion distance μ_g, it is removed from such group and creates a new group for itself. It is defined as follows: $D_{i,g} = d(\boldsymbol{p_i}, \boldsymbol{p_g})$, where d is the Euclidean distance between the agent i position and the center of its group g. If it surpasses the value μ_g, i.e. $D_{i,g} > \mu_g$, this agent leaves the group. In a similar way, if another agent j has its distance to the center of any group g lower or equal than μ_g, i.e $D_{j,g} <= \mu_g$, and the same immediate goal, it can enter to this group.

Groups of agents have a desired speed to be distributed among the members. Again, we propose to connect this concept with the group parameter. We defined the desired speed initial value of group g as $\psi_g = 1.2$ m/s. Besides desired speed, it is defined maximum and minimum standard deviation to imply in group members speeds, varying from $\sigma^{min} = 0$ to $\sigma^{max} = 0.2$. So, the individual speed of an agent A is determined as a function of group speed s_g and a standard variation σ_g which is computed as follows:

$$\sigma_g = \sigma^{max} - ((\sigma^{max} - \sigma^{min}).\frac{\zeta_g}{\zeta_{max}}), \tag{3.4}$$

where ζ_{max} stands for higher possible value for cohesion ($\zeta_{max} = 3$). Indeed, the speed deviation σ_g represents a percentage of the desired speed of the group to be randomized for the desired speed of agents of this group. For example, if a group has a desired speed $\psi_g = 1.0$ m/s and a speed deviation $\sigma_g = 0.1$, all agents of this group are going to have a desired speed randomized with 10% of variation (since $\sigma_g = 0.1$) from the group desired speed. Therefore, all agents would have a desired speed which lies in the interval [0.9, 1.1] m/s, randomly generated.

Besides desired speed and speed deviation, other two parameters are used in order to achieve group behavior: cohesion ζ and angular variation ϕ. Just as desired speed, these parameters values can be statically set and are defined, for default, as $\zeta = 3$ and $\phi = 0$. With all defined, our default groups show high cohesiveness ($\zeta = 3$), no angular variation ($\phi = 0$) and desire to move at a high speed ($s = 1.2$). Since cohesion is maximum, the speed deviation is minimum ($\sigma = 0$);

In this section, we proposed a way to find out the cohesion distance μ and the speed deviation σ based on group cohesion. Also, we defined the default values for our group parameters, such as desired speed ψ, cohesion ζ and angular variation ϕ. Therefore, we propose to define that parameters as a function of cultural aspects, presented in Section 3.3.2.

3.3.2 Hofstede's Mapping

Our approach aims to define the group parameters (i.e. cohesion, desired speed, angular variation and speed deviation) as a function of Hofstede's Cultural Dimensions (HCD) [33]. As presented in Section 2.3.1, it aims to describe how a society's culture can influence group behavior and, to this end, six dimensions are defined: PDI, MAS, LTO, ING, UNC and IND. In this work, only the MAS_g, LTO_g and ING_g dimensions (in percentages) are used for each group g. The three parameters lie between 0 and 100. In fact, in this Hofstede's mapping, our group parameters are calculated following the same mapping made by Favaretto et al. [18]. The main difference is that Favaretto used computer vision methods in video sequences to identify group of people on it and calculate its Hofstede's dimensions, while our work reverse engineer this method, using such dimensions to define group behavior.

As discussed by Favaretto et al. [18], we also propose that group cohesion ζ_g is a function of $(100 - MAS_g)$. Our assumption is that more feminine population can be referred to more cohesive population, as in Equation 3.5:

$$\zeta_g = (((100 - MAS_g) \times 3)/100), \tag{3.5}$$

where $(100 - MAS_g)$ gives the feminism percentage aspect of the dimension for group g. This value is normalized to lie between 0 and 3, which explains the multiplication by 3 and the posterior division by 100.

The group desired speed ψ_g, as described in [18], is calculated as a function of dimension ING. The idea behind is to refer an "indulgent" group as people who are in control of their lives, so it was related to the group speed. If a group is very indulgent (100% for instance), the group will try to achieve the desired speed. The value of ψ_g decreases as the value of ING_g, as stated in Equation 3.6:

$$\psi_g = (ING_g \times 1.2/100), \tag{3.6}$$

where the value 1.2 represents the maximum speed that an agent can achieve in the simulator. Therefore, this value must lie between 0 and 1.2. It is important to notice, although this calculated desired speed is set for the group, it may not be the actual simulation speed for each agent. If the speed deviation σ is higher than zero, the agent's desired speed may be higher than the group's desired value. For example, with a desired group speed equal to 1.2 and a speed deviation equal to 0.1, the agent's desired speed should lie between 1.1 and 1.3.

The angular variation present in the group motion ϕ_g was related to $(1 - LTO)$. Indeed, the assumption is that more angular variation is achieved in groups with lower value of LTO_g, which states for "Long Term orientation":

$$\phi_g = ((100 - LTO_g)/100), \tag{3.7}$$

where ϕ_g actually represents a percentage of a maximum angle, which was defined as 90 degrees.

Finally, the speed deviation σ_g is defined based on cohesion value, as stated in Equation 3.4.

Next, we propose to define our group parameters as a function of psychological aspects, presented in Section 3.3.3.

3.3.3 Durupinar's Mapping

As explained in Section 2.4.1, Durupinar et al. [13] built a simulation model based on OCEAN traits to represent emotions and its contagion among agents. In this work, the relationship between agent's low level characteristics and OCEAN traits is going to be used to determine our group attributes. The mapping made in Figure 2.12 is going to be replicated to this model and used to calculate cohesion, desired speed, angular variation and speed deviation. Just like what happened with Hofstede's cultural dimensions, just a few of them are going to be mapped, since not all of them are useful to this work. In order to define the desired speed, the Durupinar's Walking Speed behavior was chosen, since it is already a mapping for agent's speed. To define the angular variation, the Exploring Environment behavior was chosen, because it seems logical to think that the more an agent wants to explore a given environment, the more it is going to deviate from its original path. Plus, this behavior is a direct mapping from the Openness trait from OCEAN, which evaluate, among other things, the curiosity of an individual. For cohesion, the Impatience behavior was chosen, since it incorporates tolerant and orderly behaviors, necessary to maintain a group.

Finally, the Leadership behavior from Durupinar's method is used to determine if the group has a strong leader to be followed.

Firstly, Durupinar's leadership behavior is going to define how the groups parameters are going to be calculated. If the group has a strong leader, the group parameters (i.e. cohesion, desired speed, angular variation and speed deviation) are going to be calculated as a function of the Durupinar's behaviors found for this leader. Otherwise, these group parameters are going to be calculated as an average of all agents Durupinar's behaviors.

The cohesion value ζ of a certain group g is calculated as presented in Equation 3.8:

$$\zeta_g = (1 - Imp_{gD}) \times 3, \tag{3.8}$$

where Imp_{gD} represents the Durupinar's impatience (which can be the leader impatience or the average impatience of the group, as explained above). The value is multiplied by 3 in order to keep this parameter between 0 and 3, as used in this model.

The desired speed value ψ of g is calculated as in Equation 3.9:

$$\psi_g = 1.2 \times (\psi_{gD} - 1), \tag{3.9}$$

where ψ_{gD} represents the Durupinar's walking speed (which can be the leader walking speed or the average walking speed of the group, as explained above) and $(\psi_{gD} - 1)$ actually represents a percentage of a defined maximum speed (i.e. 1.2). Since Durupinar's speeds lie between 1 and 2, a simple normalization can give this value.

The angular variation value ϕ is described as in Equation 3.10:

$$\phi_g = 1 - (Ee_{gD}/10), \tag{3.10}$$

where Ee_{gD} represents the Durupinar's exploring environment (which can be the leader exploring environment or the average exploring environment of the group, as explained above) and ϕ_g actually represents a percentage of a maximum angle, which was defined as 90 degrees. The value is divided by ten in order to normalize Ee_{gD}.

Finally, the speed deviation σ_g is defined as Equation 3.4.

3.4 Agents

This Section presents attributes and behaviors which were used to enhance the agents. Section 3.4.1 presents the behaviors added in the simulation method in order to make it closer to reality. Section 3.4.2 explains how the comfort method works in our work, both for thermal and density comfort. Section 3.4.3 presents the path planning algorithm developed for this work, which is able to deal with dynamic path changing.

3.4.1 Behaviors

Usually, crowd simulation methods try to simulate different behaviors of agents. One of these behaviors, which is present in almost every method, is the Goal Seeking. Goal Seeking behavior refers to endow agents with the ability to move towards a given goal.

In the original BioCrowds model [9], as stated before, each agent has only one goal to follow, meaning a place in space it wants to achieve. Moreover, the agent knows previously the position of this goal in the geometric space. The first extension we propose in BioCrowds is to attribute to each agent a list of goals to achieve, like a schedule. For example, an agent may want to go to the restaurant, then to a soccer stadium and finally go home.

For that, it is proposed that each agent should have a list of desired goals (instead one goal only) and an intention value for each of them representing its willingness to go there. This goal list is going to be ordered from highest to lowest intention value, so the first index will always be the most "desired" agent's goal. It is important to notice that, in real life, people may not know where to go. A person may want to go to a specific restaurant, but does not know where it is. In the same way, assuming that someone wants to go to a restaurant and then to a store, if this person finds out where the store is (or already know from the beginning), it may change its intention and go to the store first, looking for the restaurant after that. So, besides the original Goal Seeking behavior already present in BioCrowds (and enhanced by our multiple goals model), we add an Exploratory Behavior, where agents do not know where their respective goals are located. For this, we use the state Looking_For (LF) explained in Section 3.2.1. It is used to make agents wander around the environment, trying to find their goals. The signs, explained also in Section 3.2.1, can help the agents to find their goals faster, giving information about the location of the objectives.

Another behavior added in our model is the cloth changing, which is directly related with the thermal comfort, presented in Section 3.4.2. If an agent becomes thermally uncomfortable, it is able to change its clothing during the simulation. So, if it feels hot, it may take some cloth off. In a similar way, if it feels cold, it may put more clothes on. More information about the cloth level is presented in Section 3.4.2.

3.4.2 Comfort

In a given environment, it is possible to exist several uncomfortable places. For example, a square may be too hot in a sunny day, or a party may have too many people near the stage. Usually, people tend to avoid uncomfortable places. In order to mimic this behavior, a comfort model is proposed, both for thermal and density situations.

A - Thermal Comfort

In order to achieve the desired thermal comfort behavior, Fanger's [17] PMV and PPD metrics were used. In this part of our simulation model, we intend to endow agents with the ability to respond to thermal comfort by changing their behaviors. At each simulation time step, the air temperature is updated according to heat propagation from existing hot or cold sources. Agents can act like hot/cold sources too, being able to influence the thermal comfort of the environment where they are inserted. Based on the calculated air temperature, as well the other five majors parameters discussed on Section 2.5.1, the thermal comfort can be calculated for each agent. In order to update the surrounding air temperature, a simple heat transfer model is used, as follows:

$$C\frac{a_T}{a_t} = \frac{a}{a_x}(k(x,y)\frac{a_T}{a_x}) + \frac{a}{a_y}(k(x,y)\frac{a_T}{a_y}) + Q(x,y,t), \tag{3.11}$$

where C is the volumetric heat capacity of the air, $k(x,y)$ represents the thermal conductivity and Q is the rate of heat generation caused by the hot or cold sources.

The PMV and PPD values are calculated for each agent, at each time step. To do so, the six major factors discussed in Section 2.5.1 are used, as follows:

- Metabolic Rate: it varies according agent's speed, with values between 1.6(idle) and 3.4(running at 5km/h) mets (1 met = 58Watts/m^2).

- Clothing Insulation: it is defined three levels of clothing: light ($I_{cl}^{min} = 0.4$), medium ($I_{cl}^{med} = 1.2$) and heavy ($I_{cl}^{max} = 2.0$).

- Air Temperature: calculated as explained in Equation 3.11

- Radiant Temperature: separated in two parts. First, t_{r0} is related with the environment itself, usually set equal to the air temperature. The second part is related with the surrounding agents. For this, it only considers agents located within the Hall's intimate zone, as shown in Figure 3.2. As one could expect, this influence should diminish as the distance gets higher and depend of their mutual orientation (i.e. it assumes the maximum value when this two agents are facing each other's face or back, and decay as their orientations approach 90 degrees). To do so, the following calculation can be applied:

  $$t_r = t_{r0} + \gamma\sum_i \frac{\epsilon + |\cos\theta_i|}{d_i^2}, \tag{3.12}$$

  where d_i is the distance between two agents and θ_i is the angle between one agent and the i_{th} agent, ϵ is an offset parameter that controls the minimum radiant temperature gain when the angle reaches its maximum value (i.e. 90 degrees). γ is defined as 0.1.

- Air speed and Humidity: parameters defined by the user.

Finally, PMV and PPD values are calculated for each agent at each timestep, as explained in Section 2.5.1. (Equations are placed here in this section again to facilitate the document reading).

$$PMV = (0.303e^{-0.036M} + 0.028)L(M, I_{cl}, t_a, t_r, v, r_h), \tag{3.13}$$

$$PPD = 100 - 95e^{-0.03353PMV^4\text{-}0.2179PMV^2}. \tag{3.14}$$

If the PMV value for a given agent stays out of the comfort zone for a defined interval of time (empirically defined as 180 seconds), a random percentage value is generated. If this value is lower than the agent's PPD value, it tries to change its clothing. If it is still uncomfortable, it tries to go to another place, following the decision method defined in Decision Method, later in this Section (C).

B - Density Comfort

As it was done in Chen et al [7], a density comfort was added in addition to the thermal comfort, making it possible to use one or another, or even both at the same time.

If agents guide themselves based only on the thermal comfort, it is possible that agents gather at high densities in cozy places. It is not an expected behavior, since people tend to keep their personal space [24] in order to stay comfortable. Therefore, besides thermal comfort, a density comfort model may be also used to determine agent's decisions and behavior. A simple model, similar to PPD, was used to estimate the local discomfort of any given agent, as follows:

$$PPD_d = 100\frac{n_i + \beta n_p}{M_i + \beta M_p}, \tag{3.15}$$

where n_i and n_p are the number of agents in the Hall's intimate and personal zones, respectively, M_i and M_p are the maximum number of agents in these two regions (defined as 6 and 12, respectively[1]), and β is the decay factor for the personal space due to its higher distance, defined as 0.2.

For clarity, the thermal PPD is from now on named PPD_t. Once two different PPD values are achieved, a simple weighted average function is used to determine the total PPD, as follows:

$$PPD = \alpha PPD_t + (1 - \alpha)PPD_d, \tag{3.16}$$

where $\alpha \in$ [0, 1] and controls the PPD compound. When it is set equal to 1, only the thermal comfort is considered. On the other hand, when it is set equal to 0, only the density comfort is considered. The way agents react to density information follows the same idea explained in Section 3.4.2, i.e. If it feels uncomfortable, it tries to go to another place, following the decision method defined in Decision Method, discussed next (C). It is important to clarify that the distances to determine the density comfort of the agents are not related with the agent radius of BioCrowds.

[1]Empirically defined

The last is used, in BioCrowds, to determine the maximum distance that an agent can reach out for markers in the environment, and is not related with the distances defined in this Section in order to find the density comfort of the agents.

C - Decision Method

As commented previously, if an agent feels uncomfortable in its current place, due to temperature or density, it may try to move toward another place (i.e. another room) to get more comfortable. This place is chosen based on the functionality of the place chosen by the agent (i.e. Restaurant, Shop, etc). When an agent wants to move, a random room with the same identifier tag as the agent's goal is chosen in the environment. For example, if an agent is inside a Theater, but feels uncomfortable in this place, another Theater room is chosen for this agent to move toward. A limitation of our method lies, right now, on this decision: if this very same agent feels uncomfortable in every room with its corresponding identifier tag, it will just keep changing rooms at every time window.

3.4.3 Path Planning

As it was already commented, we chose BioCrowds [9] as our crowd simulation method, since it is a state-of-the-art technique which guarantees collision-free movement between agents. This method aims to control the local navigation of the agents, it means, to define a new position for a given agent within a defined radius. It can arouse a problem when dealing with a scenario with obstacles: if an agent stands near an obstacle and this obstacle is larger than the defined radius, the agent will not be able to surpass this obstacle and will be locked. To solve this problem, a global navigation method can be used together with BioCrowds. We could use the global planning algorithm provided by Unity itself (i.e. NavMesh), but it would be hard to change the way it works in order to adapt to our necessities. Therefore, we built our own path planning algorithm.

We decided to implement the D* algorithm for our path planning method. This algorithm was developed by Stentz [51] as an acronym of "Dynamic A*", since the algorithm has an almost similar behavior of the well known method developed by Hart et al. [26]. The main difference between A* and D* is that the latter allows changing cost of the nodes in a dynamic way. So, each node can assume one of the following states:

- New: It is a new node, which was not yet checked;
- Open: It is in the Open List, it means, to be checked;
- Closed: It is already checked;
- Raise: Its cost is higher than the last time it was in the Open List; or

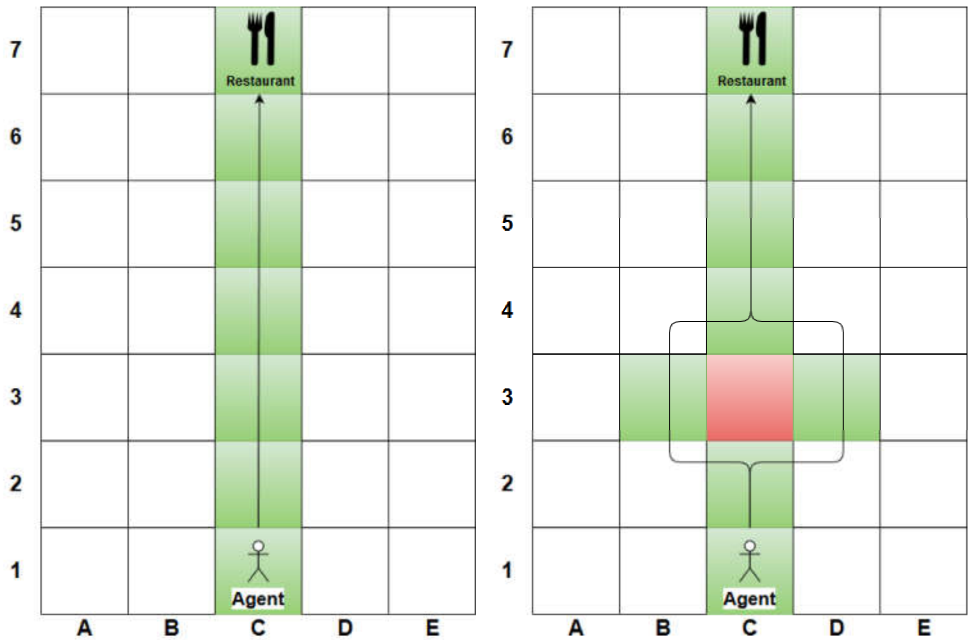
- Lower: Its cost is lower than the last time it was in the Open List.

The three first states (i.e. New, Open and Closed) were already defined in the A* method, while the Raise and Lower states were defined for the D*. The Open List stores the nodes which still need to be checked. Succinctly, the algorithm inserts the destination node in the Open List. Then, it evaluates this node and select its neighbors, adding them into the Open List also. The evaluated node is removed from the Open List and marked as Closed. Then, iteratively, the algorithm repeats this process for all nodes inside the Open List until it reaches the initial node. The evaluation process tries to find the neighbor node with the lowest cost to move to. Each chosen node keeps a pointer which refers to its predecessor, which allows the algorithm to build the path, backwards, from the destination node to the initial node.

When an obstruction is found along the path, this obstruction node is marked as Raised and added again in the Open List, as well all the affected nodes. Before increasing the cost of the node, the algorithm checks the obstruction neighbors nodes to evaluate if it can reduce the cost of the node. If not, it propagates the Raise state to all nodes in the chain path (i.e. all nodes after it which belongs to the actual path) and re-evaluate them. When an obstruction is removed, the node is marked as Lower as the algorithm follows the same pipeline as for the Raised state.

Note that the algorithm just need to work with the nodes which were affected by the obstruction, which reduces the computational cost. Even so, it can still be very costly, as following discussed. Our idea is to use the D* method to dynamically update the path of the agents according the comfort of this path. Using the thermal comfort as an example: let us say that an agent is moving towards its goal, following its defined path. Then, a thermal source is added in the middle of the path, which makes it uncomfortable so agents will avoid this place. D* algorithm is applied to solve this, raising the node cost and recalculating the path of the agent. But note that such event can occur at any time: therefore, we would need to keep updating the agent path (or, at least, checking it) at each frame or within a regular interval, which could not be so long that an agent would take too much time to change its path. Depending of the quantity of agents and the size of the scenario, the computational cost can become too high.

In order to solve this, we have made some changes on the algorithm. When a node has its cost increased (or lowered), we just recalculate the path in the betweens, it means, in the part of the path where the cost of the node changed. Figure 3.3 shows an example of how our path planning works. In Figure 3.3(a), an agent is positioned at the bottom of the environment (matrix position = (C,1)) and wants to reach a restaurant at the top (matrix position = (C,7)). Its path is defined by the black arrow and by the green nodes. The initial path of the agent is calculated just once, it means, when the goal is defined. In Figure 3.3(b), a discomfort (for example, a thermal source) is inserted in the path (matrix position = (C,3)), represented by the red node. Since, as commented before, we just calculate the path of the agent in the beginning and do not keep checking/recalculating it each frame/period, this path would not change and the agent would keep following the black arrow. However, when the event (i.e. the discomfort insertion) occurs, we check, for each agent in the simulation, if this node is present in its actual path. If it is, the path of the agent is recalculated just between the nodes not affected by the discomfort. In the example, it should be between nodes matrix position = (C,2) and matrix position = (C,4), so the agent new path can go from matrix position = (C,2) to both matrix position = (B,3) or matrix position = (D,3), and then to matrix position = (C,4). Note that, in this example, we just need to recalculate a short path between matrix position = (C,2) and matrix position = (C,4), instead to recalculate all the path again from matrix position = (C,1) to matrix position = (C,7), which improves the performance of our path planning algorithm and allows to run with more agents and bigger scenarios.



(a) Initial Path (b) New Path

Figure 3.3 – Example of our path planning calculation. In (a), an agent is positioned at the bottom of the environment (matrix position = (C,1)) and wants to reach a restaurant at the top (matrix position = (C,7)). Its path is defined by the black arrow and by the green nodes. In (b), an obstruction (for example, a thermal source) is inserted in the path (matrix position = (C,3)), represented by the red node. The path of the agent is then recalculated just between the nodes not affected by the obstruction (i.e. between nodes matrix position = (C,2) and matrix position = (C,4)), so the agent new path can go from matrix position = (C,2) to both matrix position = (B,3) or matrix position = (D,3), and then to matrix position = (C,4).

It is important to notice that the path planning will not deal with the navigation of the agents itself, which is the role of BioCrowds. Therefore, while BioCrowds deals with the local navigation of the agents, the path planning algorithm merely gives a rough path for agents to follow, avoiding obstacles in the way and situations like the one exemplified in the beginning of this subsection. In this matter, the path planning algorithm will deliver a set of subgoals for agents to follow (in a similar way as presented in Figure 3.3), and the agents are going to move towards their goals/subgoals according the definition of BioCrowds.

4. EXPERIMENTAL RESULTS

When the simulation ends, some information is available including data regarding the simulated agents. Indeed, our method generates the positions of each agent at each frame, as well as the time each of them arrived at its final goal. This data can be used in results analysis.

This section aims to present the experimental results achieved by this work and it is divided in five majors parts. Firstly, we show the achieved results in terms of the **navigation method** developed as an extension to BioCrowds [9], like the state Looking_For (LF) and the intention/signs (Section 4.1). Then, we discuss obtained results concerning the **cultural and psychological aspects** of the simulated crowds, not only for Hofstede, but also for Durupinar and Favaretto's approaches (Section 4.2). The results achieved with the **thermal comfort model** are presented in Section 4.3, followed by results achieved by our **new path planning algorithm** in Section 4.4. Indeed, we present the results with path planning here to show the integration with the thermal method. Lastly, we present our **interactive interface** to run such simulations, in Section 4.5.

4.1 Results obtained with the New Navigation Method

The first obtained result aims to provide a more realistic navigation model. As explained in Section 3.2.1, each agent may have a list of desired goals to achieve and a list of intentions defining its willingness to reach each destination. In addition, agents can start the simulation with no knowledge about where their goals are located, entering in state LF. If the agent perceives a sign in its field of view, its intention value to go to the goal informed in the sign can rise, according to Equations 3.1 and 3.2.

4.1.1 Setup

In order to proceed with the simulations, a 30x20 meters scenario is modeled, with two obstacles (gray shapes) and four goals (in red), illustrated in Figure 4.1. For each instantiated goal, a sign pointing to it is placed in its exact position. This is done to avoid a non-ending simulation. Otherwise, it would be possible to simulate a scenario with no signs, where agents would never find any goal, since they start the simulation with no knowledge about goal's location.

In all simulations, the agent's desire value δ_i to go to a specific goal is always set in 1 and the intention values γ are randomly generated in each simulation. Yet, agents have their field of view $\Upsilon = 5$, that is, they can perceive signs within a 5 meters radius. This value was empirically defined. Obstacles does not block agents view, so they are capable to perceive signs "behind" it, if within its field of view. All signs have an appeal value of 1. The red lines that can be seen in

Figure 4.1 represent the calculated paths between each agent and its next objective. The white lines surrounding each agent represent all vectors existing between that individual and its nearest markers.

4.1.2 Results

In this section, we are going to explore the results achieved by our method. First, we check if agents are truly following the pre-defined goals order assigned to them. Then, we proceed to verify if agents are being, in fact, affected by the signs placed in the environment. Finally, we present results achieved integrating our navigation method with an optimized virtual environment.

A- Goals Order

In order to test the scheduled goals, the initial setup is used. Two agents (in blue) are instantiated with desire to go to every goal in the environment, following a predetermined intention priority. Since we have chosen to have randomly generated intention values γ, it was defined that "Agent0" wants to go first to "Goal3", then "Goal1", "Goal2" and "Goal4", while "Agent1" wants to go firstly to "Goal2", then "Goal1", "Goal3" and "Goal4". Signs and states LF are removed to avoid influencing the pre-defined order. It is expected that both agents follow their defined schedule of goals and are capable of achieving them. Tests show that both agents follow their schedule just as expected i.e. obeying the determined order. Figure 4.1 shows two agents following their scheduled goals. Table 4.1 presents the arrival frame time for both agents, in each of their scheduled goals. It can be seen that the order each agent arrived at its goals is the same as defined in the initial list.

B- Signs Interaction

To check if interactions among agents and signs are properly occurring, the same initial scenario is used. Just one agent is instantiated with desire to go to every goal in the scene, following a predetermined intention order, which is: "Goal3", "Goal2", "Goal1" and "Goal4". Two signs are placed in the scene: "Sign1" pointing to "Goal1" and "Sign2" pointing to "Goal2". Figure 4.2 presents this configuration, where the yellow tags are the two new signs described (i.e. S1 and S2). Red tags represent the four goals (i.e. G1, G2, G3 and G4), while the gray geometric shapes are obstacles OB1 and OB2. The agent is represented by AG1. The green selection at the right shows

Table 4.1 – Arrival times for both agent0 and agent1

Agent0	Frame Time
Goal3	1239
Goal1	3764
Goal2	6020
Goal4	8182

Agent1	Frame Time
Goal2	243
Goal1	2751
Goal3	5241
Goal4	7633

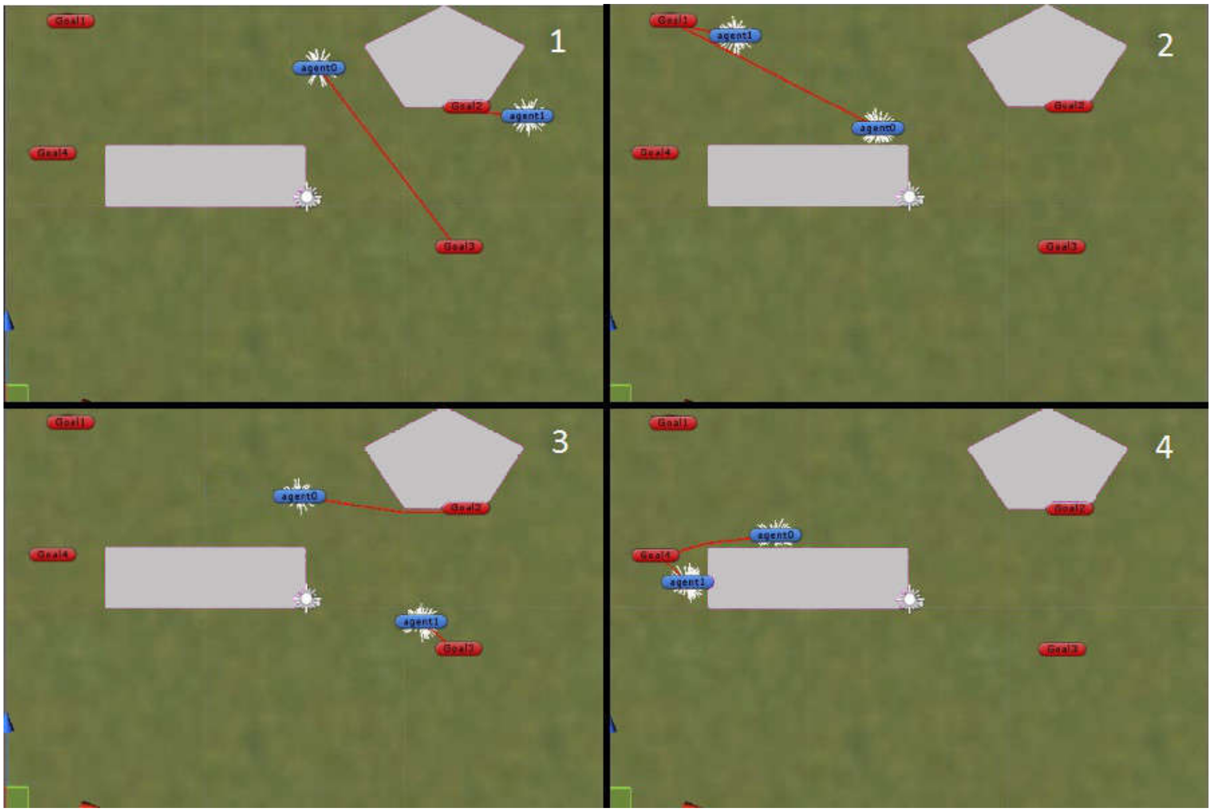


Figure 4.1 – Two agents following their scheduled goals.

the schedule of the agent, properly ordered by the intention values (blue selection). Therefore, the agent starts the simulation in the state LF and wants to achieve Goal3, Goal2, Goal1 and Goal4, respectively. It is important to remember that, as explained in Section 4.1.1, signs are placed in each goal's location to avoid a non-ending simulation. However, in this test, the signs in the "Goal1" location and "Goal2" location are removed. This is made in order to make it easier to check the interaction with the two new placed signs. If they remained in the simulation, they would be affecting the instantiated agent and it would be possible for it to achieve their goals even with the absence of the two new signs.

Tests show that interaction with signs works exactly as expected, it means, the two new instantiated signs truly impacts the agent's schedule. The agent starts the simulation looking for a sign, following the schedule which can be seen in Figure 4.2-1. However, when it perceives "Sign1" (S1), the intention to go to "Goal1" (G1) rises, overpassing the intention to go to "Goal3" (G3) (Figure 4.2-2). Consequently, the agents status changes from LF to TG , i.e. towards "Goal1" (G1). Figure 4.2 shows agent intentions changing, according signs on the way. Figure 4.2-1 shows the initial configuration of the simulation. The instantiated agent starts in the state LF. However, when moving toward its path, it passes near "Sign1" (S1) which points to "Goal1" (G1). As expected, when this sign comes into the field of view of the agent, its intention to go to "Goal1" (G1) rises, until the point that this value surpasses the state LF value (e.g. 0.8) and "Goal1" (G1) becomes the new agent's target to achieve, as it can be seen in Figure 4.2-2. Yet, Table 4.2 shows the arrival frame time of the agent, for all its goals, and the frames that an interaction with a sign started to occur. Table 4.3 presents the moment that the agent changed its path from an objective to another, due to the change of its intention value. It is possible to notice in Table 4.2 that the order which "agent0" reach its goals is different from the initial defined order, which seems to validate the expected behavior of changing the order of the agent's goals achievement, depending on the environment specification.

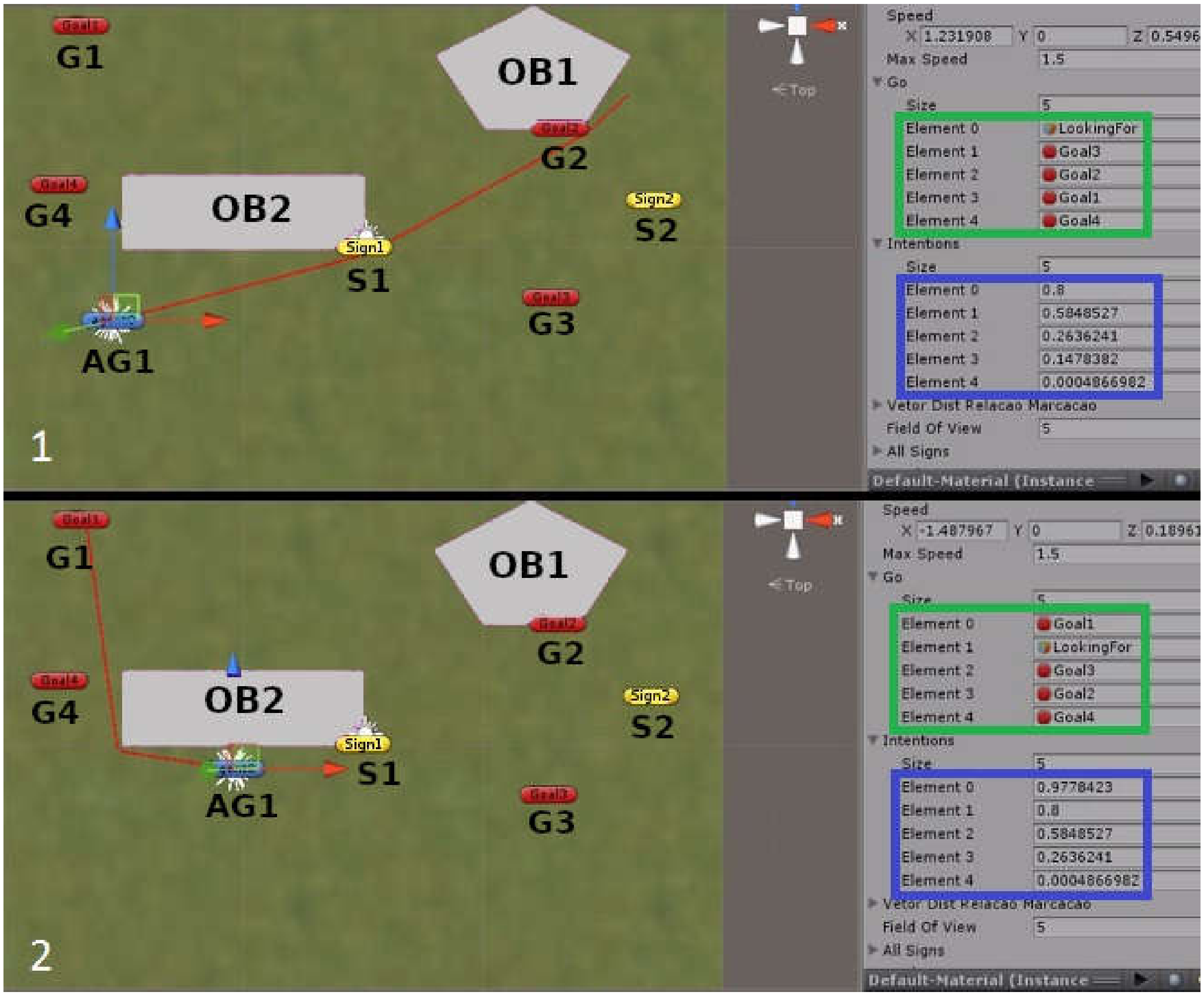


Figure 4.2 – Interaction with signs. Agent's intentions changes according signs in the environment. The red line leaving the agent represents its path. The green selection at the right shows the schedule of the agent, properly ordered by the intention values shown by the blue selection, also at the right. 1) Initial configuration of the simulation. 2) The moment when the agent's willingness for "Goal1" (G1) surpasses the one for state LF. At the right, it is possible to see the schedule order and the intention values for each objective.

In order to verify if the quantity of signs truly impact in the simulation time (i.e. the agents arrival time), ten simulations were run with exactly the same parameters, only changing the number of signs. Here, a 30x20 scenario is used with the same two obstacles already used before, where four goals are set and ten agents are instantiated with different random desires (but the same random desires are used for the ten simulations). The first simulation had no signs and each subsequent

Table 4.2 – Arrival times for agent0 and interaction with signs

	Frame Time
Goal4	1860
Goal1	2673
Goal2	5202
Goal3	8089

Table 4.3 – Changed paths as a function of time for agent0

From	To	Frame
LF	Goal1	827
Goal1	Goal4	1359
Goal4	Goal1	1860
Goal1	LF	2673
LF	Goal3	7565

simulation had a new sign added (i.e. 1, 2, ..., 9). A random goal was defined for each sign and its appeal value is set to 1. It was expected that the more signs, the shorter the simulation time would be. Table 4.4 shows simulation mean time and quantity of signs from all ten simulations. It happened in some of the analysis, however, as can be seen from simulations 6 to 10, it seems that this expected behavior is not true. Some hypothesis are raised here. It can be the result of the random process to place new signs and their positions, but it can also show that there is an optimal number of signs in such simulation. Plus, it can just be the effect of the random nature of the state LF. Further analysis are needed in order to properly conclude that.

C- Integration with Optimized Virtual Environment

This navigation model was used in a collaborative work [1] recently submitted [54]. The idea for that work was to change the environment in order to produce a desired crowd behavior, instead to alter the parameters of the crowd itself. In other words, adapt the environment to better

[1] Collaboration with Purdue University

Table 4.4 – Simulations mean times with quantity of signs

	Frame Time	Qnt Signs
Sim 1	8946.8	0
Sim 2	6471.5	1
Sim 3	4327.1	2
Sim 4	4015.7	3
Sim 5	4040.1	4
Sim 6	4777.9	5
Sim 7	2877.7	6
Sim 8	2458	7
Sim 9	3014.2	8
Sim 10	3294.7	9

suit the expected crowd behavior evolution. Figure 4.3 shows one of the urban layouts used to test the framework (Venice), along with agents walking on the streets.

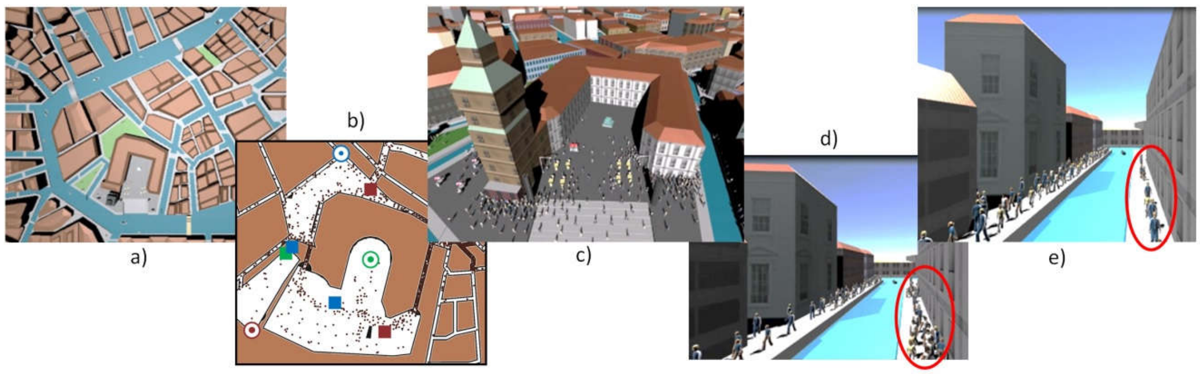


Figure 4.3 – Urban layout used as test in Purdue/PUCRS paper, along with agents walking on it. [54]

The approach has three main pillars. First, it is proposed a procedural representation of the environment, with walkways, signs and goals (in that work, the appeal of the signs was kept constant and equal to 1), where agents can evolve on it. Second, the proposed method in that work is extended for crowd simulation, allowing agents to walk and interact with signs along the way. Third, it is proposed an inverse modeling framework to find an optimal environment solution according defined expected indicators, like time taken to reach goals, walked distance and walkway cost. Tests showed that the approach was able to simulate large crowds in real world scenarios, like Venice, and make changes in the environment in order to find an optimal configuration, like changes with signs (add, remove, change position, etc.) and with the road structure (add/remove a walkway, change walkway width, etc.). It is important to notice that the mentioned third part of that work was developed by the researchers in Purdue University, so it is not included in our dissertation.

One of the results tested in that work was exactly the effect of changing signs quantity and location, taking into account the state LF. For this, a simple urban layout is constructed and 1000 agents are instantiated, as shown in Figure 4.4. Two spawning spots are defined (green circles) from where agents will enter in the environment, one at the lower left and another at the upper right corner. The spawned agents try to achieve a goal (g1 and g2) at the opposite side from where they start. In Figure 4.4a it is presented a basic scenario with no signs, where it was necessary more than 10000 time steps for 97% of the total crowd to reach their goal. In Figure 4.4b their inverse model calculated the best position for two signs (s -> g1 and s -> g2, where s -> g1 means a sign pointing to g1), which diminished the total time to 5626 time steps. Finally, Figure 4.4c shows the positions found by the inverse model to place four signs, which reduced the total time to 3505 time steps. In all images, signs are represented by the blue and brown squares.

Section 4.2 presents some results obtained with the model to describe Cultural aspects.

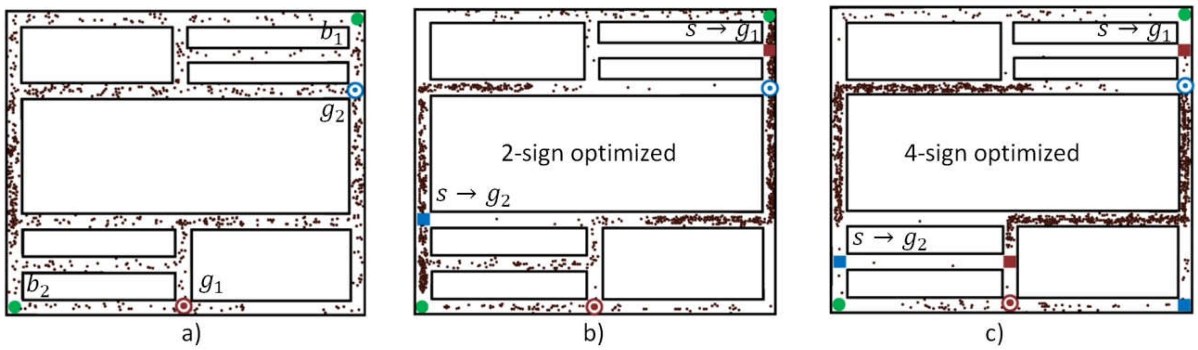


Figure 4.4 – a) Scenario with no signs, taking more than 10000 time steps. b) Scenario with two signs (s -> g1 and s -> g2), taking 5626 times steps. c) Scenario with four signs (s -> g1 and s -> g2), taking 3505 time steps. In all images, signs are represented by the blue and brown squares. [54]

4.2 Results Obtained with Cultural and Personality Method

As explained in Section 3.3.1, the cultural aspect proposed in this work is defined, so far, as a function of cohesion ζ, desired speed ψ, speed deviation Ω and angular variation ϕ. The main idea is to be able to use a known cultural/psychological method (for example, Hofstead and OCEAN described in Section 2) as a input, and map cultural parameters to group and individual motion definitions.

Hofstede was chosen because it is, as far as we known, the only work in literature which addressed the population cultural mapping. To not be restricted to only a cultural model, a psychological model is going to be considered too. In fact, Durupinar's model performs a mapping from OCEAN psychological traits to agent's behavior, more specifically to its emotions and how the emotions and contagion affect the decision and acting of simulated agents.

4.2.1 Setup

In order to proceed with the subject of cultural simulations, a 30x30 meters scenario is modeled with four goals (red markers), illustrated in Figure 4.5. The exploratory behavior is deactivated, since it is not necessary here.

For all cultural simulations, ten agents are instantiated inside the same group near the position that can be seen in Figure 4.5. They have a fixed list of goals to follow, which is Goal2, Goal1, Goal4 and Goal3.

Eight test simulations were made for each method (i.e. Hofstede's mapping and Durupinar's mapping), varying the input values (i.e. Hofstede's cultural dimensions and OCEAN). The idea is to check how different input values affect the crowd behavior, both from cultural aspects and psychological traits. As explained in Chapter 3.3, we translate cultural and psychological traits into groups features (i.e. cohesion, desired speed, angular variation). Therefore, it is expected that

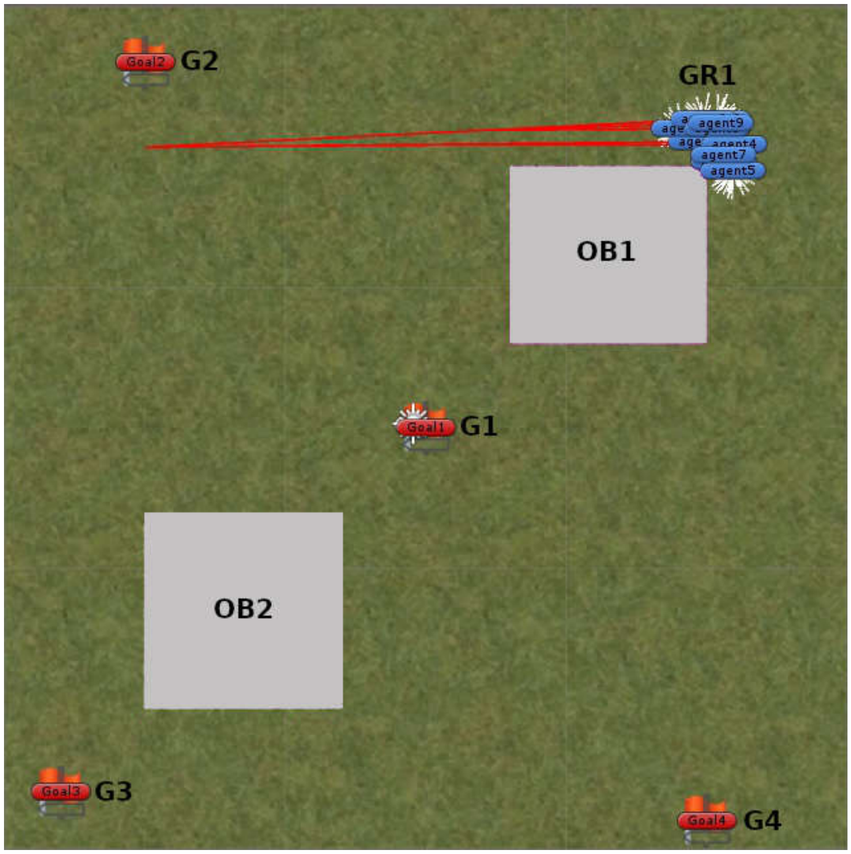


Figure 4.5 – Cultural testing environment. Ten agents (i.e. GR1) start near the upper right corner, inside the same group. Red markers are goals (i.e. G1, G2, G3 and G4), namely Goal1, Goal2, Goal3 and Goal4, while the gray squares are obstacles (i.e. OB1 and OB2).

with higher cohesion values, agents tend to stay inside its original group and within closer distances; as well spread more, in the environment, with lower cohesion values. Plus, it is expected some difference concerning groups formation when changing its desired speed and angular variation. High cohesion value is defined in the interval $[2.5; 3]$ and low cohesion value is defined between $[0; 1]$. As well, high desired speed value is defined between $[1; 1.2]$ and low desired speed value is defined between $[0.2; 0.4]$. Also, high angular variation value is defined between $[60; 90]$ degrees and low angular variation value is defined between $[0; 15]$.

In order to verify the differences in the obtained results with the simulations, some metrics were defined, as follows:

- Time: the time (in seconds) which the simulation needed to complete;
- Maximum quantity of groups: the maximum number of groups formed during the simulation;
- Average speed: the average speed (in m/s) achieved by the agents during the simulation;
- Average angular variation: the average angular variation (in degrees) agents assumed during the simulation; and

- Average distance: the average distance (in meters) which agents kept from its group's center.

For average speed, average angular variation and average distance, their respective standard deviation are going to be considered too. These metrics are used in Sections 4.2.2 and 4.2.3.

4.2.2 Results for Hofstede's Mapping

As stated before, eight simulations were executed where we varied the value of Hofstede's cultural dimensions, as presented in Table 4.5.

Table 4.5 – Input for Hofstede's Mapping

Sim	PDI	MAS	LTO	ING
1	50	17	90	90
2	50	9	90	20
3	50	80	90	90
4	50	72	90	20
5	50	17	30	90
6	50	9	30	20
7	50	80	30	90
8	50	72	30	20

Following the formulation presented in Section 3.3.2, we calculate each group parameter (i.e. cohesion, desired speed, speed deviation and angular variation), generating the following configuration for each simulation:

- 1: High cohesion, high desired speed and low angular variation
- 2: High cohesion, low desired speed and low angular variation
- 3: Low cohesion, high desired speed and low angular variation
- 4: Low cohesion, low desired speed and low angular variation
- 5: High cohesion, high desired speed and high angular variation
- 6: High cohesion, low desired speed and high angular variation
- 7: Low cohesion, high desired speed and high angular variation
- 8: Low cohesion, low desired speed and high angular variation

For all eight simulations, the metrics defined in section 4.2.1 are computed and shown in Table 4.6, as well the standard deviations in Table 4.7.

After all eight simulations, it is possible to notice that the result expected at the beginning was achieved, it means, groups with higher cohesion values (i.e. Simulations number 1, 2, 5 and 6)

Table 4.6 – Results for Hofstede's Mapping

Sim	Time	Max Groups	Avg Speed	Avg Ang	Avg Dist
1	126	1	0,61	16,7	1,25
2	595	1	0,13	17,77	1,19
3	89	1	0,81	11,9	2,19
4	628	3	0,14	12,05	2,19
5	178	2	0,55	34,8	1,32
6	780	2	0,13	38,08	1,22
7	122	1	0,67	27,13	2,3
8	732	1	0,12	31,13	2,2

Table 4.7 – Standard Deviations for Hofstede's Mapping

Sim	Avg Speed	Avg Ang	Avg Dist
1	0,08	5,64	0,08
2	0,02	6,33	0,10
3	0,06	4,84	0,20
4	0,02	4,70	0,24
5	0,11	11,16	0,14
6	0,02	10,19	0,09
7	0,09	6,87	0,18
8	0,01	7,11	0,18

presented closest agents and vice-versa. Yet, the desired speed and angular variation seemed to had influence in group behavior, specially to keep the group together, it means, no agents leaving the group.

The simulations with more cohesion were the ones with the lowest values for Average Distance, meaning that agents stayed closer between each other. Plus, in the simulations with low angular variation (the first four ones in Table 4.6), the only one that generated more groups was the number 4, which had a low cohesion value.

As expected, the angular variation influenced the crowd behavior, specially when the simulations had a high value for this trait. As can be seen in Table 4.6, even with high cohesion values, agents were able to split into other groups (simulations numbers 5 and 6). A possible explanation for this behavior can be due the way we generate the angular variation. To do so, we randomly pick one side of the agent (i.e. left or right) and nullify all markers that are inside the angular variation value. For example, if an agent has an angular variation defined as 20^o, a random side is chosen, let us say, right side. So, in the right side of the agent, we create a cone with an aperture of 20^o and nullify all markers inside this cone. Due the way as BioCrowds works, explained in Section 2.2.2, it should make the agent turn more to the left while walking. Since the markers of one side of the agent, inside the angular variation, are nullified, if this side is the group's center side, the agent will move away from the group and, therefore, split eventually. Yet, the Table 4.6 also shows that the last two simulations (i.e. 7 and 8), with low cohesion values, had less groups than the previous two (i.e. 5 and 6), with high cohesion values. It is possible that the angular variation may be having a great impact on the group formation, but more experiments are required in order to confirm or refute this.

4.2.3 Results for Durupinar's Mapping

For the simulations with Durupinar's mapping, the same eight simulations presented in Section 4.2.2 were run, just changing the input for the agents (i.e. from Hofstede to Durupinar). Therefore, each group parameter is going to be calculated according input Durupinar's behavior values, following the formulation described in Section 3.3.3. Table 4.8 shows the Durupinar's OCEAN input used for each simulation.

Table 4.8 – OCEAN Input for Durupinar's Mapping

Sim	O	C	E	A	N
1	0.9	0.9	0.9	0.9	0.1
2	0.9	0.9	0.2	0.9	0.1
3	0.9	0.2	0.9	0.2	0.1
4	0.9	0.2	0.2	0.2	0.1
5	0.3	0.9	0.9	0.9	0.1
6	0.3	0.9	0.2	0.9	0.1
7	0.3	0.2	0.9	0.2	0.1
8	0.3	0.2	0.2	0.2	0.1

For all simulations, the same metrics defined in section 4.2.1 are computed and shown in Table 4.9, as well the standard deviations in Table 4.10

Table 4.9 – Results for Durupinar's Mapping

Sim	Time	Max Groups	Avg Speed	Avg Ang	Avg Dist
1	128	1	0,6	16,89	1,26
2	579	1	0,13	16,95	1,21
3	87	1	0,84	11,73	2,23
4	648	4	0,16	11,11	1,97
5	170	2	0,56	34,41	1,32
6	789	2	0,13	38,5	1,22
7	124	1	0,65	26,5	2,31
8	608	2	0,14	31,16	2,22

After all eight simulations, it is possible to notice that the result expected at the beginning was also achieved, it means, groups with higher cohesion values (i.e. Simulations number 1, 2, 5 and 6) had closest agents and vice-versa. Plus, as observed with Hofstede's mapping simulations, the desired speed and angular variation seemed to had influence on group behavior, mainly in the group formation. Simulations with higher cohesion presented lowest Average Distances, so agents stayed nearer each other. Besides, when looking at the results for simulations with low angular variation, the only one which split the initial group was the number 4, which had low cohesion.

Table 4.10 – Standard Deviations for Durupinar's Mapping

Sim	Avg Speed	Avg Ang	Avg Dist
1	0,07	5,34	0,09
2	0,02	5,47	0,09
3	0,06	4,49	0,18
4	0,04	5,97	0,25
5	0,11	11,70	0,12
6	0,02	10,78	0,10
7	0,09	7,08	0,21
8	0,02	6,92	0,20

Just as observed in the simulations with Hofstede's mapping in Section 4.2.2, it is interesting to notice that high values of angular variation generated similar results. Even with high cohesion values, agents were able to split into other groups (simulations numbers 5 and 6). In fact, all simulations with Durupinar's values as input had very similar results with the simulations with Hofstede's values as input. As example, the simulation number 5 for Durupinar's mapping had the same Average Distance as the number 5 for Hofstede's mapping, and similar values for the other metrics. Figure 4.6 shows a comparison between the metrics found for both Hofstede's mapping and Durupinar's mapping. It is possible to see that the resulting bars for both methods, in every metric, present a similar behavior.

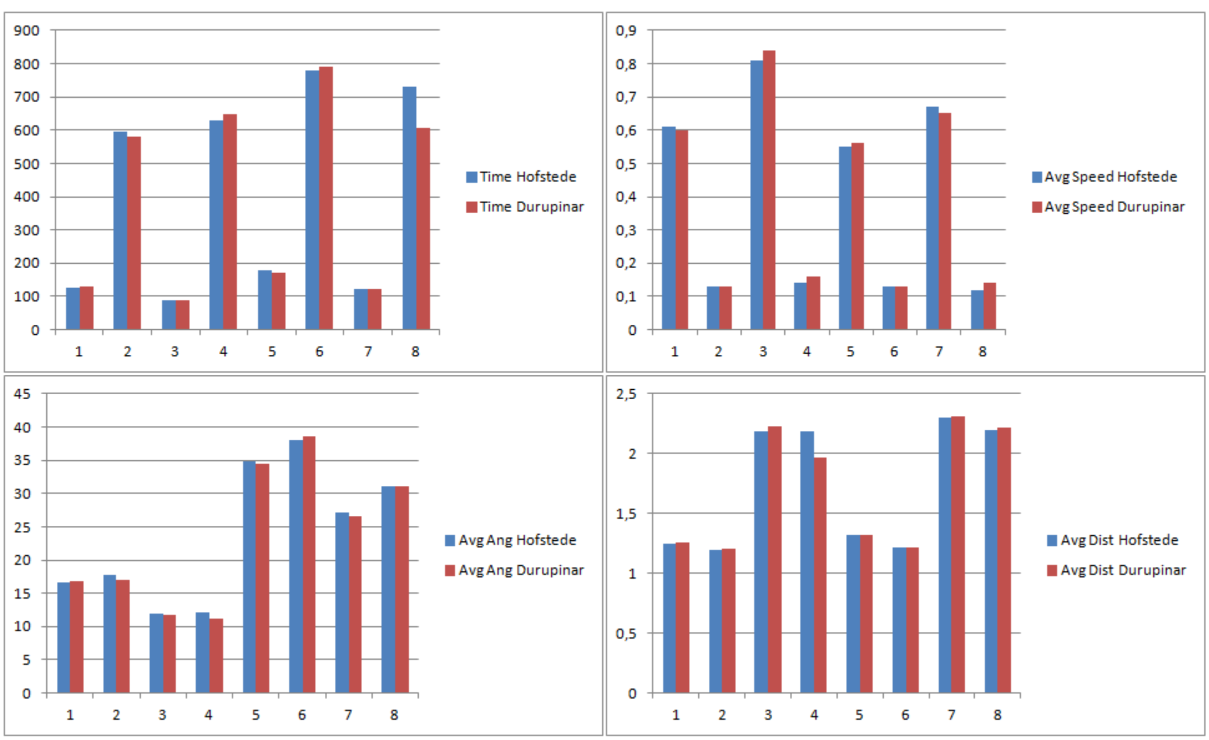


Figure 4.6 – Comparison between Hofstede's mapping and Durupinar's mapping

With the similar behavior observed in Figure 4.6 and the input values for both Hofstede's and Durupinar's mapping (Tables 4.5 and 4.8), it seems that a correspondence between both can be done, it means, the same behavior achieved with a Hofstede's mapping can be achieved with a Durupinar's mapping, and vice-versa. For example, the results achieved for Simulation number 1 with Hofstede's inputs set as: PDI = 50, MAS = 17, LTO = 90 and ING = 90; can similarly be achieved with Durupinar's inputs set as: O = 0.9, C = 0.9, E = 0.9, A = 0.9 and N = 0.1. As we inferred in Section 3.3.2, a low value for Hofstede MAS dimension is referred to a more cohesive behavior, while a high value for LTO and ING is related with a low angular variation and a high speed, respectively. The same relationship can be done for Durupinar's mapping, following the explanation presented in Section 3.3.3. It is possible to see, in Figure 4.6, that such expected behavior indeed occurred in Simulation number 1, for both methods.

4.2.4 Original BioCrowds

In order to compare our cultural/psychological results with the original BioCrowds, we also run simulations with the original model. We used the same scenario presented in Section 4.2.1, with agents following the same schedule (i.e. Goal2, Goal1, Goal4 and Goal3). In original BioCrowds, one can define a group formed by people who has the same goal, but no group structure is simulated, i.e. they do not try to be close to each other.

Figure 4.7 shows a comparison between the metrics achieved by all three sets of simulations (i.e. Hofstede's mapping, Durupinar's mapping and BioCrowds). It is possible to see the influence of cultural/psychological input when compared to the original behavior of BioCrowds algorithm. The simulations with BioCrowds took less time to complete in all 8 simulations. Also, the achieved speeds for Original BioCrowds are higher for all simulations, when compared with the subject cultural mappings. The angular variation was usually lower in BioCrowds when compared with the cultural simulations, just being little higher in Simulations number 3 and 4. We believe that, in the cultural simulations, the low cohesion values used in Simulations number 3 and 4 generated a lower angular variation, when compared with Simulations number 1 and 2, because agents had more space to walk. As discussed in Section 3.3.1, the cohesion value should control the willingness of agents of the group to stay close and together. When agents are too close, they have low space to move. It naturally generates a more noisy trajectory, it means, agents do not follow a straight line towards their respective goals. The same behavior can be seen when comparing Simulations number 5 and 6 with Simulations 7 and 8, which seems to validate our theory. The average distances between agents are always much higher in BioCrowds when compared with the other two scenarios. We believe that, once BioCrowds do not group agents, agents are more free to walk and achieve their desired speed (which also explains the higher speeds). However, since all ten agents begin the simulation around the same spawn position, some agents need to wait for a free space to move around, which makes them move slower. So, some agents are able to move faster in the beginning, distancing themselves from the agents that need to wait a little longer to move at a higher pace, which creates a distance gap. On the other hand, the simulations with cultural/psychological mapping are able to control the group formation, which could explain why the BioCrowds presented far higher distances between agents.

Therefore, the results achieved suggest that the cultural/psychological aspects introduced by our work helps to enhance the realism of the model, making agents behave in an heterogeneous way, depending on their personality and cultural behaviors.

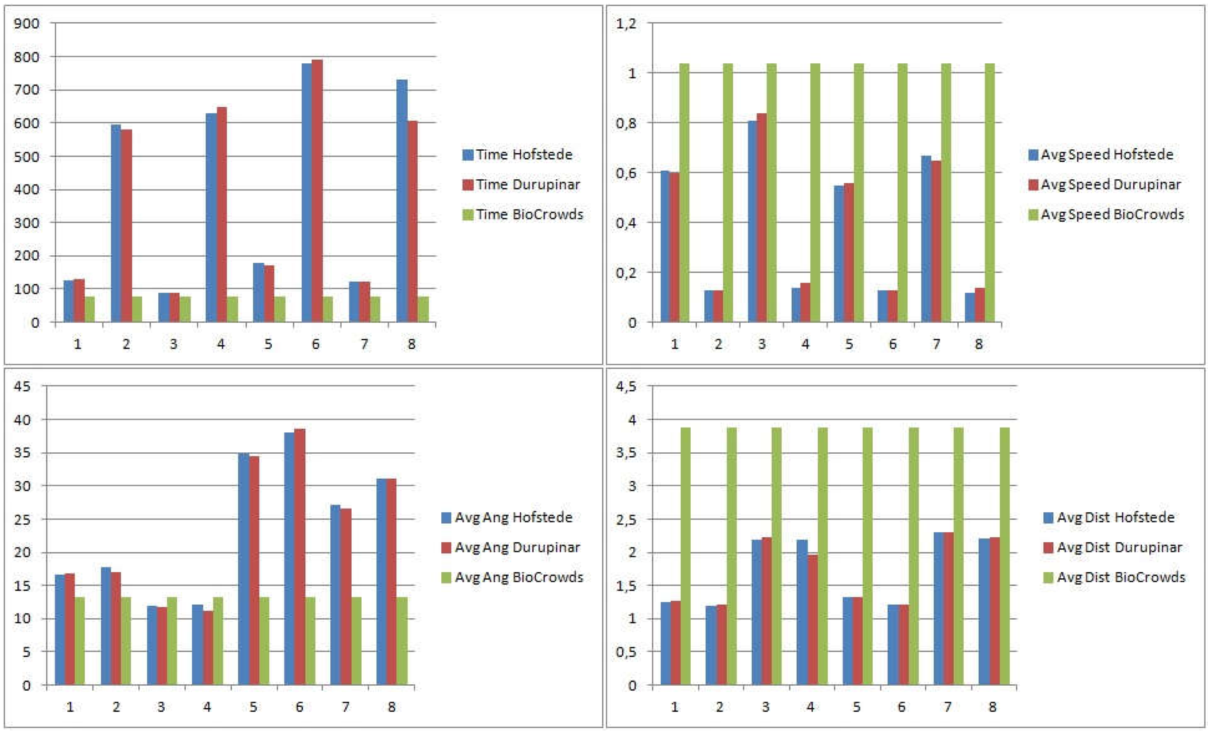


Figure 4.7 – Comparison between Hofstede's mapping, Durupinar's mapping and BioCrowds. It is possible to see that the achieved values in original BioCrowds are always the same for different simulations, when compared with varied values obtained with the cultural mappings.

Section 4.3 presents the results regarding the thermal comfort model.

4.3 Results Obtained with the Comfort Method

As explained in Section 3.4.2, a thermal comfort behavior was added for agents. In order to do so, Fanger's [17] PMV and PPD metrics were used. Plus, as it was done in Chen et al [7], a density comfort is also added alongside with the comfort, making it possible to use one or another, or even both at the same time.

4.3.1 Setup

In order to proceed with the subject comfort simulations, a 22x40 meters scenario is modeled, with eight rooms, illustrated in Figure 4.8. Each room has an identifier tag associated, which defines its space functionality. Agents are spawned at the center bottom of the environment (red dot), with a random general goal to achieve (i.e. restaurant, shop or theater) which matches one of the possible room's identifier tags. The room's colors denote its actual thermal comfort. Green means it is cozy ($t_a = 18$), blue means it is cold ($t_a = 12$) and yellow means it is hot ($t_a = 26$), where t_a represents the air temperature in Celsius.

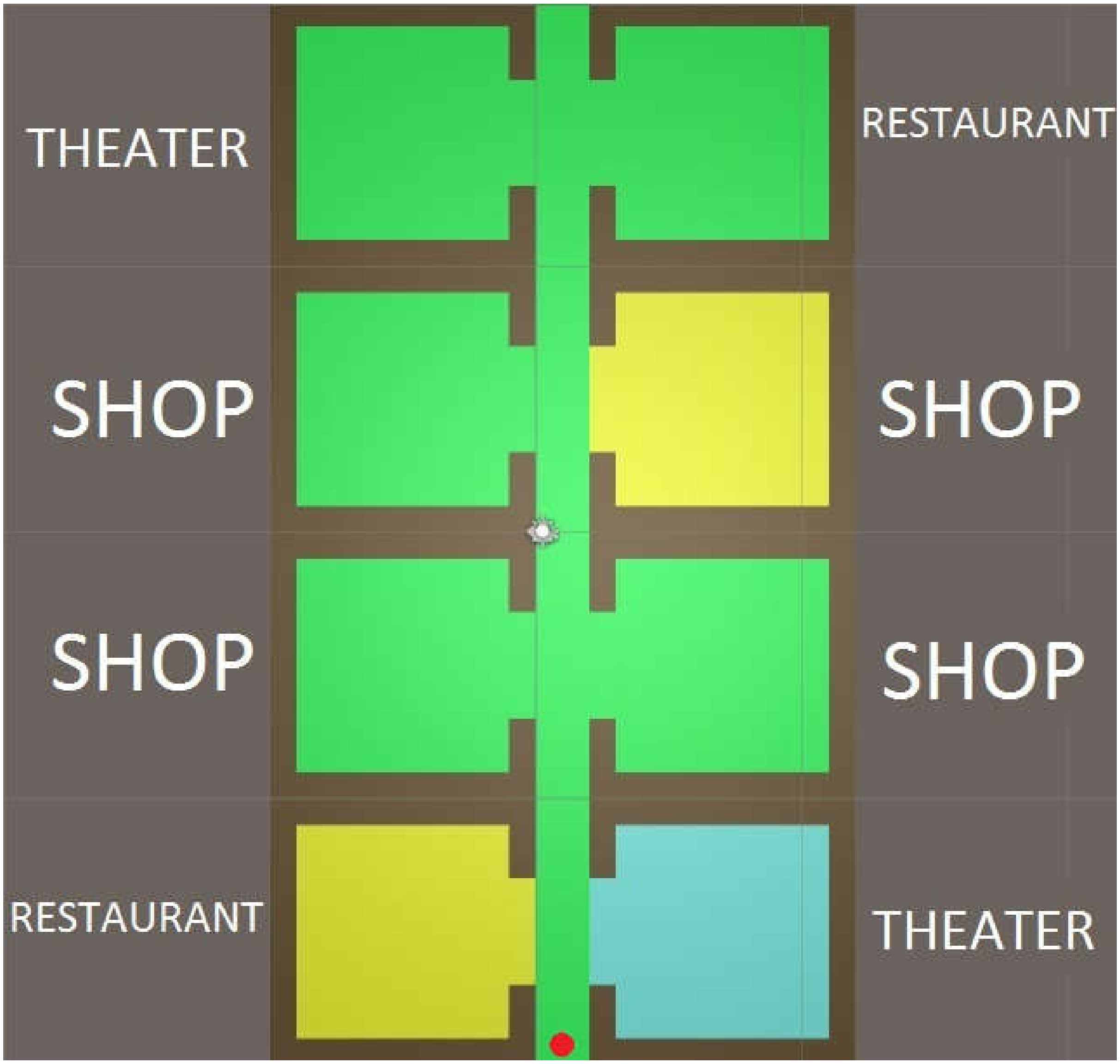


Figure 4.8 – Comfort testing environment. Agents are spawned at the center bottom of the environment (red dot), with a random general goal to achieve (i.e. restaurant, shop or theater). Green rooms means it is cozy, blue means it is cold and yellow means it is hot.

All comfort simulations keep spawning agents until a max defined value is achieved (i.e. 100). Agents start the simulation with a random general goal to achieve and medium clothing insulation (i.e. $I_{cl} = 1.2$).

It was ran three test simulations, one with solely thermal comfort, other only with density comfort and a last one with both. The idea is to check if agents avoid uncomfortable places, moving themselves to cozier locations. It is expected that uncomfortable agents search for another place with the same identifier tag, therefore, incommodious rooms should be almost or totally empty.

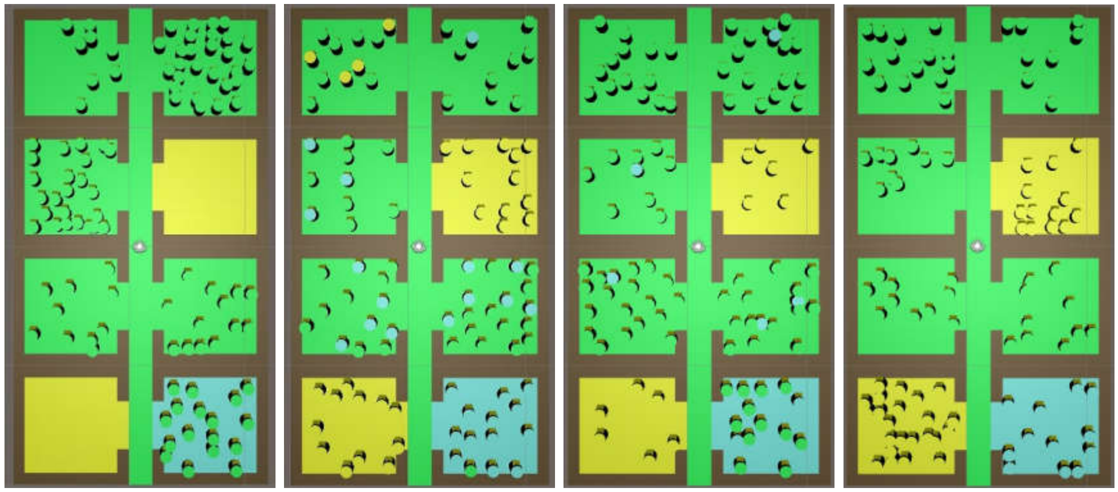
4.3.2 Results

The first simulation was ran using only the thermal comfort, therefore, only the thermal aspects of both environment and agents impacts on its comfort. Figure 4.9(a) shows the final positioning of agents. It is possible to notice that no agents were placed inside the bottom Restaurant, neither inside the first shop from the right (yellow rooms), which were the rooms with an elevated temperature. Some agents were comfortable inside the bottom Theater, which had a low temperature. This can be explained by the clothing insulation of these agents, which were self altered to 2 during the simulation, it means, they are wearing a heavy piece of clothing and feel comfortable inside this room.

The second simulation was ran using only the density comfort, therefore, only the density aspects of both environment and agents impacts on its comfort. Figure 4.9(b) shows the final positioning of agents. It is possible to notice that agents spread out across all rooms, independent of the temperature. Using the Restaurant general goal as an example, it is possible to see that agents in this simulation (Figure 4.9)(b) stayed more spread (i.e. 0.10 agents/m^2 in the top right room and 0.12 agents/m^2 in the bottom left room) than in the thermal simulation (Figure 4.9(a)) (i.e. 0.28 agents/m^2 in the top right room and no agents in the bottom left room).

The third simulation was ran using both thermal and density comfort, setting the bias $\alpha = 0.5$. Figure 4.9(c) shows the final positioning of agents. It is possible to notice that, as expected, it seems to be a halfway between only thermal and density methods. The two "hot rooms" (yellow ones) have agents inside, but in a lower number than it can be seen in the density method. As it was done in the previous simulation, it is possible to use the Restaurant general goal as an example. The densities found for both top right and bottom left rooms were 0.16 agents/m^2 and 0.06 agents/m^2, respectively, which had less agents for the hot Restaurant in Density simulation (0.12 agents/m^2), but more than in the Thermal simulation (0 agents/m^2).

At the end, a final simulation was run with the comfort behavior deactivated. The idea was to check if the proposed comfort method is being useful in order to deliver better results than a random choice. Figure 4.9(d) shows the final positioning of the agents. It can be noticed that agents are just randomly distributed across the eight rooms, according their respective goals. As it was done in the previous simulation, it is possible to use the Restaurant general goal as an example. The densities found for both top right and bottom left rooms were 0.08 agents/m^2 and 0.24 agents/m^2, respectively. The "hot room" has a lot more agents than the cozy room, which is completely different from the behavior found in the other three comfort simulations, where the densities for this room was always lower.



(a) Thermal Comfort (b) Density Comfort (c) Thermal + Density Comforts (d) No Comfort

Figure 4.9 – All three comfort simulation, plus the simulation with no comfort. In (a), no agents were observed inside the hot rooms (yellow rooms). Agents inside the cold room (blue one) are wearing a heavy piece of clothing. In (b), it is possible to notice that agents spread out across all rooms, independent of the temperature. Blue agents are feeling a bit cold and yellow agents are feeling a bit hot. In (c), Blue agents are feeling a bit cold and yellow agents are feeling a bit hot. In (d), simulation with comfort deactivated. Blue agents are feeling a bit cold and yellow agents are feeling a bit hot.

It is important to elucidate a drawback of this method. It is possible for agents to keep changing places due to discomfort in a non-stopping way. Taking the third simulation as example, an agent inside the "hot rooms" is feeling thermally uncomfortable. Even with this parameter just counting as half for the total PPD value, at some point it may want to go to another similar place. And again, if this new place is too hot and/or crowded, at some point this agent may want to go to another place again, and so on. In this work, a time window of three minutes is used in order to mitigate this problem, therefore, agents wait for three minutes before taking any action.

Section 4.4 presents the results regarding the path planning algorithm.

4.4 Results Obtained with Path Planning

As explained in Section 3.4.3, we used the path planning algorithm known as D* [51] to calculate the global path planning for our agents, including the thermal features.

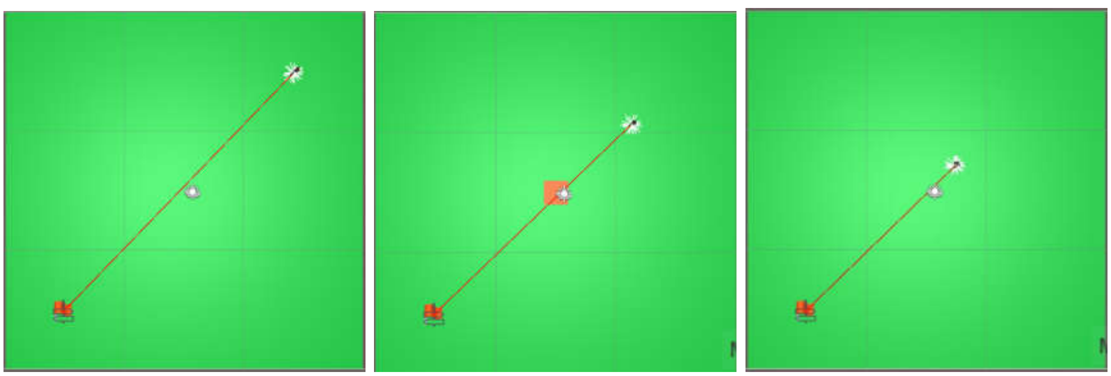
4.4.1 Setup

In order to proceed with the Path Planning simulation, a 30x30 meters scenario is modeled, with just one agent and one goal, illustrated in Figure 4.10a. The red line between the agent (at the top right of the environment) and its goal (at the left bottom of the environment) represents the path of the agent. The agent has its thermal comfort activated (i.e. $\alpha = 1$), so any temperature change in the environment should trigger a response on it. The idea is to check if our path planning is going to alter the agent path when a cell present in its path becomes uncomfortable. We expect that it occurs, making the agent avoid that cell. Also, we expect that the agent gets back to its original path once the uncomfortable cell becomes comfortable again.

(a) Initial Path with Dynamic Path Planning

(b) New Path with Dynamic Path Planning

(c) Back to Initial Path with Dynamic Path Planning



(d) Initial Path with Original BioCrowds

(e) Same Path with Original BioCrowds

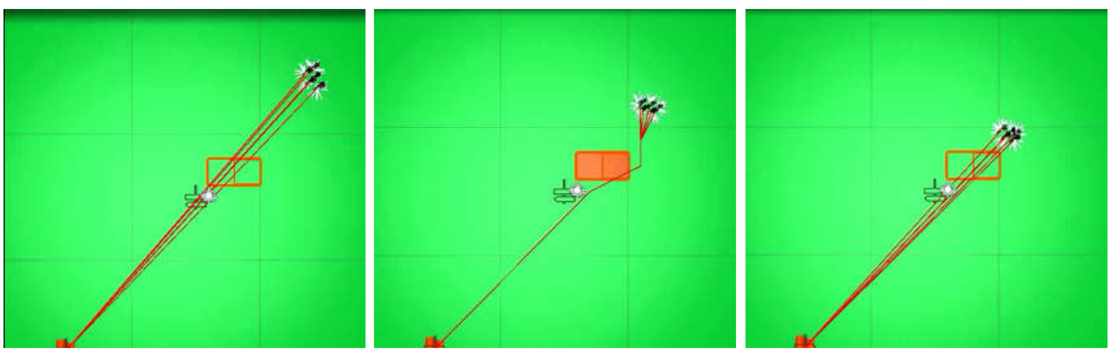
(f) Still Same Path with Original BioCrowds

Figure 4.10 – Example of our path planning calculation. The agent starts at the right top of the environment and wants to reach a goal in the left bottom. When the agent is created, its initial path is calculated (a). Then, in a given moment, a hot source is inserted in the scenario, which can be seen as a red cell (b). So, the agent has a new path to follow which avoids the uncomfortable cell. In the same way, if the hot source is removed (c), the agent returns to its original path. Also, in original BioCrowds, the agent starts with its initial path (d), but when we insert the hot source in the environment, the agent does not change its path (e). In the same way, the path remains unaltered when the same hot source is removed from the environment (f).

4.4.2 Results

As shown in Figure 4.10, we ran a simulation with one agent starting at the right top of the environment and willing to reach a goal at the left bottom. The red line shows the path that the agent follows towards its goal. When the agent is created, its initial path is calculated (Figure 4.10a). Then, in a given moment, a hot source is inserted in the scenario, which can be seen as a red cell (Figure 4.10b). When it happens, the algorithm finds out if that cell is part of the agent path. Since it is true, it recalculates the path only between the cells which are right before and right after this hot cell: so, the agent has a new path to follow which avoids the uncomfortable cell. In the same way, if the hot source is removed (Figure 4.10c), the agent returns to its original path.

We also ran a simulation with the same setup using only BioCrowds, without the dynamic path planning. The idea is to be able to compare our method with the original BioCrowds and see if it is generating different results. In Figure 4.10d, it is possible to see that the agent has the same original path as in Figure 4.10a. Although, when we insert the hot source in the environment, the agent does not change its path (Figure 4.10e). In the same way, the path remains unaltered when the same hot source is removed from the environment (Figure 4.10f). Therefore, it seems that the dynamic path planning is being able to generate a more realistic behavior for the agents.



(a) Initial Path with Dynamic Path Planning (b) New Path with Dynamic Path Planning (c) Back to Initial Path with Dynamic Path Planning

Figure 4.11 – Example of our path planning calculation with groups. Five agents start at the right top of the environment and wants to reach a goal in the left bottom. When the group is created, the initial path for each agent of inside it is calculated (a). Then, in a given moment, in a given moment, a hot source is inserted in the scenario, which can be seen as a red rectangle (b). So, each agent which path intersected the hot zone has a new path to follow which avoids the uncomfortable zone. In the same way, if the hot source is removed (c), the agents return to their original path.

We can also test our path planning algorithm with groups. We ran another simulation with the same setup presented in Section 4.4.1, using a group of 5 agents, instead one agent only. Figure 4.11 shows some key frames of it. When the group is created, the initial path for each agent of inside it is calculated (Figure 4.11a). Then, as it was done with the simulation with one agent only, in a given moment, a hot source is inserted in the scenario, which can be seen as a red rectangle (Figure 4.11b). When it happens, the algorithm finds out if that zone is part of the agents path, for each agent inside the group. Since it is true, it recalculates the path only between the cells which are right before and right after this hot zone: so, each agent which path intersected the hot zone has a new path to follow which avoids the uncomfortable zone. In the same way, if the hot source is removed (Figure 4.11c), the agents return to their original path.

4.5 Interactive Interface

As it was already mentioned, our crowd simulator has many and varied inputs, like desired speed for agents, cohesion for groups, cultural and psychological traits, etc. When a tool has that many inputs, it is expected that the person using it is going to change them in order to achieve a desired behavior, or just to see what happens if some parameters change.

In our simulator, all input parameters are controlled by different files. So, for example, we have an input file for the environment, with all cells and markers needed for BioCrowds to work; we have another file with the information regarding goals and signs present in the environment; yet another file with information about our agents, and so on. Therefore, if a person wants, for example, to change the quantity of agents in the simulation, he/she would need to open the project folder, find the respective file and alter the information there. An even worst case can be given: if this same person wants to change the simulation environment, it would be needed to find the respective file and change the configuration of all cells and markers in the scenario. The amount of work to do this would be huge.

To solve this problem, we have developed an interactive interface for users be able to tweak the parameters of their simulations in an easy way. Figure 4.12 presents our interactive interface. All inputs used by our simulator can be altered on it. In Agents Setup, the information about agents can be changed, like the quantity of agents to be simulated, their cultural or psychological traits, initial position of the agents, etc. In Scenario Setup, the user can define obstacles, goals and signs. Also, the PRE-COMPILE button redefine the environment using the defined Size X, Size Z and Markers Density, as well the obstacles defined by the user, using the respective file. So, the user does not need to define all cells and markers manually.

4.6 Discussion about Results

In this Chapter, we presented many results obtained with the various methods proposed in this work. We wanted to show how the crowds react more realistically when they do not know the goals position (exploratory behavior) and use signs in the environment to guide themselves. In fact, this extension was used in a collaborative work with the University of Purdue [54], which goal was to adapt an environment in function of the crowd behavior.

Regarding cultural and psychological aspects of the crowd, we wanted to show how varied agents behaviors can be having different inputs, such as Hofstede cultural dimensions and Durupinar. The results achieved show that agents presented, indeed, different behaviors when different inputs were used. For example, when using Durupinar Mapping, with an OCEAN input as follows: $O = 0.9$, $C = 0.2$, $E = 0.9$, $A = 0.2$, $N = 0.1$, agents were able to move faster then when using an OCEAN input as follows: $O = 0.9$, $C = 0.9$, $E = 0.2$, $A = 0.9$, $N = 0.1$. However, the second input generated groups with agents closer between each other than the simulation with the first input.

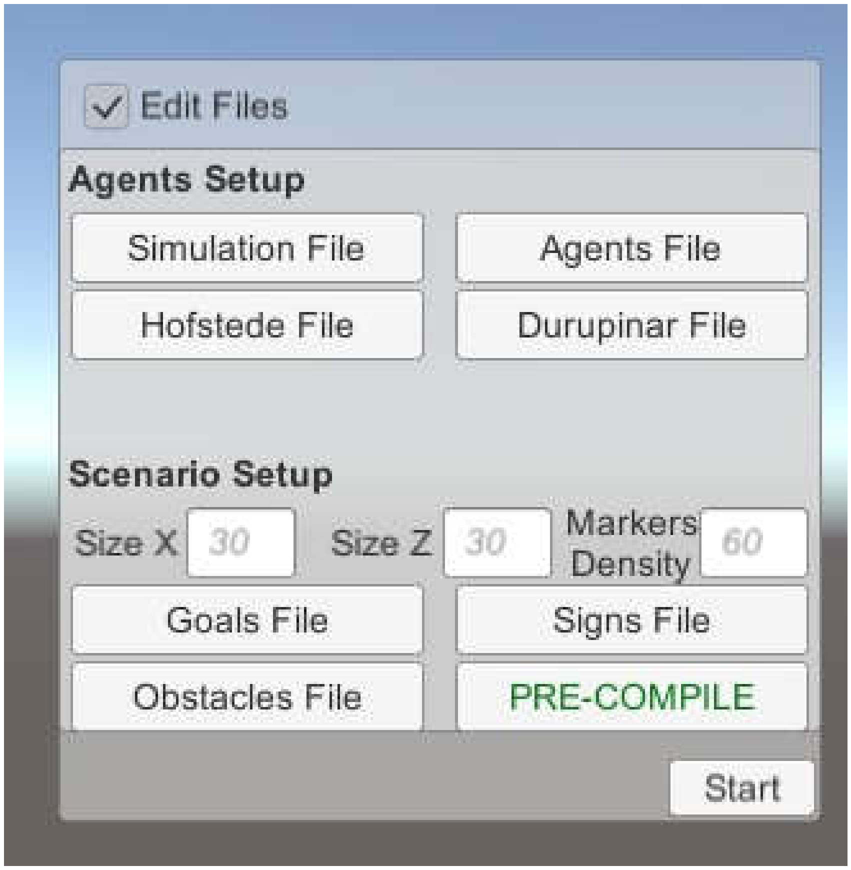


Figure 4.12 – Interactive Interface.

As we presented in Section 3.3.3, the desired speed of the agents is determined by the Walking Speed (from Durupinar) and the cohesion is determined by the Impatience (from Durupinar). In Durupinar's behaviors, the Walking Speed is related with E OCEAN trait, while the Impatience is related with E, C and A OCEAN traits. Therefore, since we changed E, C and A input traits between the two simulations, it was indeed expected differences in speeds and distances between agents.

With respect to the comfort, we wanted to show how agents would behave when they feel uncomfortable, being this discomfort due to thermal or density reasons. The achieved results show that agents presented different behaviors when comforts were used. For example, when the thermal comfort was activated, agents preferred to stay grouped in cozy rooms (i.e. rooms with a comfortable temperature). On the other hand, when the thermal comfort was replaced with the density comfort, agents preferred to distribute themselves along the rooms, trying to keep some distance between themselves. The path planning algorithm developed in this work allowed us to have a better control of the dynamic paths, where spatial comfort can dynamically change too. It is used to compute better trajectories when the environment becomes uncomfortable, and agents could better plan their paths. Finally, the interactive interface is specially useful for users, who do not need to search for input files and parameters to be able to tweak the simulations: they can only use the interface to inform the desired input values and run a respective simulation.

5. FINAL CONSIDERATIONS

This work proposes an extension for Biocrowds model [9] to make its agent's navigation more realistic, while providing agents endowed with characteristics that can generate diversity of behaviors. We propose to re-parametrize BioCrowds based on cultural and psychological dimensions, providing a framework able to simulate cultural crowds. The achieved results, using the navigation extension, seem to be consistent with what was expected, it means, agents were able to follow their defined schedule of goals and the objectives order of this schedule was truly updated due to interaction with signs along the way. In addition, agents were able to react to thermal and density comfort, improving their ability to react to environmental changes. Concerning the cultural approach, the results achieved with the Hofstede's [32] and Durupinar [13] inputs seem promising, since agents remained close when their group had a high cohesion value and tended to be spread when had a low cohesion value.

It is interesting to mention about this work validation. All results presented in Section 4 show that the framework works as intended, it means, it delivers the expected output, both for the cultural and comfort cases. However, this quantitative evaluation may not be enough to truly answer the question if the crowd is behaving according to its defined cultural parameters or the environment defined thermal/density values. Therefore, a qualitative evaluation could be done in order to complement the results already achieved. For example, an evaluation with subjects could be conducted with different cultural simulations, so it should be possible to see if real people could perceive the difference between these different simulations, or if they could identify a determined culture based on the visual simulation. For the comfort simulations, some real-life scenarios could be recreated as simulations in order to see if the achieved results are similar with what would be expected.

Another problem concerning the validation of our cultural method is due the ground truth data. Such information could be obtained by video sequences, from traffic cameras, drones, etc. However, when watching such videos, it is hard to determine the cultural or psychological aspects of the crowd and, therefore, observe such behavior. Favaretto et al. [18] developed a method able to extract Hofstede cultural dimensions from people in video sequence, and another work [19] which does the same for OCEAN traits. Although we can use Favaretto approaches to validate our simulations with video sequences, we can not know for certain that our agents are indeed behaving like, for example, a given personality. In another matter, even the comfort of people can vary according the society they are inserted, or their personality. People from hot countries are more used to stand under the sun than people from cold countries, which, in turn, would prefer to stay, for example, under a tree. As suggested by Chattaraj et al. in their work [6], people from India seems to be more used to walk in denser crowds than people from Germany, which could show a different measure for density comfort between these two countries.

As for future work, there are many things to be done. As already commented, an evaluation with subjects could be conducted with different cultural simulations in order to validate our method.

Other cultural and psychological models could be added in our framework (for example, Favaretto dimensions [18]). In fact, following the interactive interface idea implemented in this work, it could be extended to accept any cultural/psychological model. As our framework is, if a new cultural model need to be added, all the formulation need to be hard-coded for it to work properly. One idea is to be able to insert any model, along with its formulation, and the framework would be able to run such cultural/psychological simulations. Plus, the simulation model could be extended to vehicles, which would allow the framework to simulate the behavior of drivers according its nationality or personality. Also, in this work, we were just able to simulate a restricted amount of agents inside small environments. Another avenue of work would be to extended this model to be able to simulate large environments with a big quantity of agents. Such simulations would be highly costly and many methods would need to be applied in order to be able to build it. Also, the visualization of such simulations is not a trivial task, since it would need to be able to focus on different parts of the environment, or on a given event/group/agent. Finally, the cultural/psychological aspects of the crowd could also be applied to define the willingness of agents to achieve a given goal, as well its susceptibility to react due interaction with signs along the way.

BIBLIOGRAPHY

[1] Berlonghi, A. E. "Understanding and planning for different spectator crowds", *Safety Science*, vol. 18–4, Feb 1995, pp. 239–247.

[2] Bosse, T.; Duell, R.; Memon, Z. A.; Treur, J.; van der Wal, C. N. "Agent-based modeling of emotion contagion in groups.", *Cognitive Computation*, vol. 7–1, Feb 2015, pp. 111–136.

[3] Carretero, M. R.; Peters, C.; Qureshi, A. "Modelling emotional behaviour in virtual crowds through expressive body movements and emotion contagion". In: Proceedings of the Sociedad Iberoamericana de Grafica Digital, 2014, pp. 95–98.

[4] Cassol, V.; Oliveira, J.; Musse, S. R.; Badler, N. "Analyzing egress accuracy through the study of virtual and real crowds". In: Proceedings of the Virtual Humans and Crowds for Immersive Environments, 2016, pp. 1–6.

[5] Cassol, V. J.; Testa, E. S.; Jung, C. R.; Usman, M.; Faloutsos, P.; Berseth, G.; Kapadia, M.; Badler, N. I.; Musse, S. R. "Evaluating and optimizing evacuation plans for crowd egress", *IEEE Computer Graphics and Applications*, vol. 37–4, Jul-Aug 2017, pp. 60–71.

[6] Chattaraj, U.; Seyfried, A.; Chakroborty, P. "Comparison of pedestrian fundamental diagram across cultures", *Advances in Complex Systems*, vol. 12–03, May 2009, pp. 393–405.

[7] Chen, L.; Jung, C. R.; Musse, S. R.; Moneimne, M.; Wang, C.; Fruchter, R.; Bazjanac, V.; Chen, G.; Badler, N. I. "Crowd simulation incorporating thermal environments and responsive behaviors", *Presence Journal*, in press.

[8] Cook, R. L. "Stochastic sampling in computer graphics", *ACM Transactions on Graphics*, vol. 5–1, Jan 1986, pp. 51–72.

[9] de Lima Bicho, A. "Da modelagem de plantas à dinâmica de multidões: um modelo de animação comportamental bio-inspirado", Ph.D. Thesis, Universidade de Campinas, 2009, 114p.

[10] de Lima Bicho, A.; Rodrigues, R. A.; Musse, S. R.; Jung, C. R.; Paravisi, M.; Magalhães, L. P. "Simulating crowds based on a space colonization algorithm", *Computers & Graphics*, vol. 36–2, Apr 2012, pp. 70–79.

[11] Degens, N.; Jan, G.; Mascarenhas, S.; Silva, A.; Paiva, A.; Kistler, F.; André, E.; Swiderska, A.; Krumhuber, E.; Kappas, A.; Hume, C.; Hall, L.; Aylett, R. "Traveller - intercultural training with intelligent agents for young adults". In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, 2013, pp. 1387–1388.

[12] Dodds, P. S.; Watts, D. J. "A generalized model of social and biological contagion", *Journal of Theoretical Biology*, vol. 232–4, Feb 2005, pp. 587–604.

[13] Durupınar, F.; Güdükbay, U.; Aman, A.; Badler, N. I. "Psychological parameters for crowd simulation: From audiences to mobs", *IEEE Transactions on Visualization and Computer Graphics*, vol. 22–9, Sep 2016, pp. 2145–2159.

[14] Durupinar, F.; Pelechano, N.; Allbeck, J.; Gudukbay, U.; Badler, N. I. "How the ocean personality model affects the perception of crowds", *IEEE Computer Graphics and Applications*, vol. 31–3, May-Jun 2011, pp. 22–31.

[15] Dyaram, L.; Kamalanabhan, T. "Unearthed: the other side of group cohesiveness", *Journal of Social Science*, vol. 10–3, Oct 2005, pp. 185–90.

[16] Eysenck, H.; Eysenck, M. "Personality and individual differences: A natural science perspective". USA: Plenum, 1985, 452p.

[17] Fanger, P. O.; et al.. "Thermal comfort. Analysis and applications in environmental engineering." DK: Danish Technical Press, 1970, 244p.

[18] Favaretto, R. M.; Dihl, L.; Barreto, R.; Musse, S. R. "Using group behaviors to detect hofstede cultural dimensions". In: Proceedings of the 2016 IEEE International Conference on Image Processing, 2016, pp. 2936–2940.

[19] Favaretto, R. M.; Dihl, L.; Musse, S. R.; Vilanova, F.; Costa, A. B. "Using big five personality model to detect cultural aspects in crowds". In: Proceedings of the 30th Conference on Graphics, Patterns and Images, 2017, pp. 223–229.

[20] Fridman, N.; Zilka, A.; Kaminka, G. A. "The impact of cultural differences on crowd dynamics in pedestrian and evacuation domains", Technical Report, The MAVERICK Group Computer Science Department Bar Ilan University, 2011, 60p.

[21] Fruin, J. J. "Pedestrian planning and design", Technical Report, Metro Asso Urban Dsgners & Envir Plners, 1971, 206p.

[22] Goldberg, L. R. "An alternative description of personality: the big-five factor structure.", *Journal of Personality and Social Psychology*, vol. 59–6, Jun 1990, pp. 1216–1229.

[23] Guy, S. J.; Kim, S.; Lin, M. C.; Manocha, D. "Simulating heterogeneous crowd behaviors using personality trait theory". In: Proceedings of the 2011 ACM Special Interest Group on Computer Graphics and Interactive Techniques/Eurographics Symposium on Computer Animation, 2011, pp. 43–52.

[24] Hall, E. "The hidden dimension, vol. 1990", *USA: Anchor Books*, 1969, 240p.

[25] Hansen, N. "A cma-es for mixed-integer nonlinear optimization", Ph.D. Thesis, Institut National de Recherche en Informatique et en Automatique, 2011, 13p.

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

[27] He, L.; Pan, J.; Narang, S.; Wang, W.; Manocha, D. "Dynamic group behaviors for interactive crowd simulation", *Captured in: https://arxiv.org/abs/1602.03623*, Sep 2018.

[28] Helbing, D.; Keltsch, J.; Molnar, P. "Modelling the evolution of human trail systems", *Nature*, vol. 388–6637, Jul 1997, pp. 47–50.

[29] Helbing, D.; Molnar, P. "Social force model for pedestrian dynamics", *Physical Review E*, vol. 51–5, May 1995, pp. 4282–4286.

[30] Henderson, L. "The statistics of crowd fluids", *Nature*, vol. 229, Feb 1971, pp. 381–383.

[31] Henderson, L. F. "On the fluid mechanics of human crowd motion", *Transportation Research*, vol. 8–6, Dec 1974, pp. 509–515.

[32] Hofstede, G. "Dimensionalizing cultures: The Hofstede model in context", *Online Readings in Psychology and Culture*, vol. 2–1, Dec 2011, 8p.

[33] Hofstede, G. H. "Culture's consequences: Comparing values, behaviors, institutions and organizations across nations". USA: Sage, 2001, 616p.

[34] Jelić, A.; Appert-Rolland, C.; Lemercier, S.; Pettré, J. "Properties of pedestrians walking in line: Fundamental diagrams", *Physical Review E*, vol. 85, Mar 2012, 9p.

[35] Knowles, E. S.; Bassett, R. L. "Groups and crowds as social entities: Effects of activity, size, and member similarity on nonmembers.", *Journal of Personality and Social Psychology*, vol. 34–5, Nov 1976, pp. 837–845.

[36] Krontiris, A.; Bekris, K. E.; Kapadia, M. "Acumen: Activity-centric crowd authoring using influence maps". In: Proceedings of the 29th International Conference on Computer Animation and Social Agents, 2016, pp. 61–69.

[37] Lala, D.; Thovuttikul, S.; Nishida, T. "Towards a virtual environment for capturing behavior in cultural crowds". In: Proceedings of the 6th International Conference on Digital Information Management, 2011, pp. 310–315.

[38] Le Bon, G. "Psychology of Crowds (annotated)". UK: Sparkling Books, 2009, 208p.

[39] Mascarenhas, S.; Degens, N.; Paiva, A.; Prada, R.; Hofstede, G. J.; Beulens, A.; Aylett, R. "Modeling culture in intelligent virtual agents - from theory to implementation (extended abstract)". In: Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, 2016, pp. 909–910.

[40] McDAVID, J. W.; Harari, H. "Psicologia e comportamento social". BR: Interciencia, 1980, 446p.

[41] Mehrabian, A. "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament", *Current Psychology*, vol. 14–4, Dec 1996, pp. 261–292.

[42] Mitchell, D. P. "Generating antialiased images at low sampling densities". In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques, 1987, pp. 65–72.

[43] Momboisse, R. M. "Riots, revolts, and insurrections". USA: Charles C. Thomas Publisher, 1967, 523p.

[44] Musse, S. R.; Cassol, V. J.; Jung, C. R. "Towards a quantitative approach for comparing crowds", *Computer Animation and Virtual Worlds*, vol. 23–1, Feb 2012, pp. 49–57.

[45] Neto, A. B. F.; Pelachaud, C.; Musse, S. R. "Giving emotional contagion ability to virtual agents in crowds". In: Proceedings of the International Conference on Intelligent Virtual Agents, 2017, pp. 63–72.

[46] Ortony, A.; Clore, G. L.; Collins, A. "The cognitive structure of emotions". UK: Cambridge University Press, 1990, 207p.

[47] Paravisi, M.; Werhli, A.; Junior, J.; Rodrigues, R.; Jung, C.; Musse, S. "Continuum crowds with local control". In: Proceedings of the Computer Graphics International, 2008, pp. 108–115.

[48] Reynolds, C. W. "Flocks, herds and schools: A distributed behavioral model". In: Proceedings of the ACM Special Interest Group on Computer Graphics and Interactive Techniques, 1987, pp. 25–34.

[49] Runions, A.; Fuhrer, M.; Lane, B.; Federl, P.; Rolland-Lagan, A.-G.; Prusinkiewicz, P. "Modeling and visualization of leaf venation patterns", *ACM Transactions on Graphics*, vol. 24–3, Jul 2005, pp. 702–711.

[50] Sachs, T. "The control of the patterned differentiation of vascular tissues", *Advances in Botanical Research*, vol. 9, Feb 1981, pp. 151–262.

[51] Stentz, A. "Optimal and efficient path planning for partially-known environments". In: Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 3310–3317.

[52] Still, G. K. "Crowd dynamics", Ph.D. Thesis, The University of Warwick, 2000, 281p.

[53] Thalmann, D.; Musse, S. R. "Crowd Simulation, Second Edition." UK: Springer, 2013, 296p.

[54] Tharindu, M.; Knob, P.; Aliaga, D. G.; Musse, S. R. "Urban walkability design using virtual population simulation", *Computer Graphics Forum*, in press.

[55] Treuille, A.; Cooper, S.; Popović, Z. "Continuum crowds". In: Proceedings of the ACM Transactions on Graphics, 2006, pp. 1160–1168.

[56] Van Den Berg, J.; Guy, S. J.; Lin, M.; Manocha, D. "Reciprocal n-body collision avoidance". In: *Proceedings of the Robotics Research*, 2011, pp. 3–19.

[57] Yersin, B.; Maïm, J.; Pettré, J.; Thalmann, D. "Crowd patches: populating large-scale virtual environments for real-time applications". In: Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, 2009, pp. 207–214.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br