PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
DEPARTAMENT OF INFORMATICS
GRADUATE PROGRAMME IN COMPUTER SCIENCE

SIMULATION OF INTER AND INTRA GROUP BEHAVIORS USING SEMANTIC VIRTUAL ENVIRONMENTS

RAFAEL HOCEVAR DE MOURA

Dissertation presented as partial requirement for obtaining the master's degree in Computer Science from Pontifical Catholic University of Rio Grande do Sul.

Adviser: Prof. Dr. Soraia Raupp Musse

Porto Alegre, Brazil
2013

This Page Intentionally Left Blank.

"Because your own strength is unequal to the task, do not assume that it is beyond the powers of man; but if anything is within the powers and province of man, believe that it is within your own compass also."

Marcus Aurelius

ACKNOWLEDGMENTS

I express here my sincere gratitude to all the amazing people that I had the opportunity to work with, during the master's experience.

I would like to thank my adviser Soraia Musse, an excellent professor whom I am proud of work with, good years of my life. Thanks for guiding me to so many achievements and accomplishments that surely will be decisive for my future. Professor Nicolas Courty, thanks for giving me the opportunity to work beside you during these last few months, certainly was an incredible experience that I could happily enjoy the most. Professor Márcio Pinho, thanks for the aid, "pitacos" and suggestions that were indeed very welcome. Thanks to Professor Milene Silveira, who showed me that it is possible to overcome challenges by simply having a little of dedication.

My VHLab teammates, thank you for all companionship: lots of chimarrões and stories to tell. Vinícius Cassol and Henry Braun, thanks for all the support and the sincere friendship. Colleagues Fernando Marson, Rossana Queiroz, Adriana Braun, Leandro Dihl, Laura Flach, Julio Jacques, Cristina Scheibler, Luiz Cunha, Anderson Silva and Federico Delgado, I'm sure you all will have much success and a bright future.

I also thank my colleagues at UBS for being so nice to me. Ali Makke, Petra Bosilj, Pamela Medrano and Lei Chen, thanks for all the help when I needed. Abdulkader Benchi and Armel Esnault, thanks for the french lessons and coffee talks. Thibaut Le Naour and Djamel Benferhat, I really appreciated our soccer matches: "le temps importe peu"! Jiří Havel, thank you for showing me Vannes.

To friends who were always around for sharing good moments: Roger Granada, Alexandre Seki, Vanderson Dill and Diogo Lima. Marcos Eich and Lucas Bagatini, thanks for all the chatting and beers.

Narjara Costa, thank you from all my heart for being part of my life. I learned so much from the wonderful person you are. I love you!

Thanks to my father Valdir Moura, my mother Sandra Hocevar and family for always being by my side and providing me everything I needed in life.

This work was developed with support of the Coordination of Improvement of Higher Education Personnel (CAPES), Brazil.

SIMULATION OF INTER AND INTRA GROUP BEHAVIORS USING SEMANTIC VIRTUAL ENVIRONMENTS

ABSTRACT

Simulation of everyday situations from real life can be a very useful tool in entertainment applications and training systems. Such applications, as games or computer animated movies usually need to provide virtual environments populated with virtual autonomous agents. Commonly, the agents need to be able to evolve in their environment, avoiding collision with each other and obstacles, besides interacting with other characters in order to provide realistic simulations. This work presents a model to simulate coherent group behaviors based on procedural modeling and semantic environments. The main focus is to provide agents connected to the virtual environment they are evolving, mainly applied in the background of games or movies generated with few/without user intervention.

Keywords: Crowd Simulation; Virtual Humans; Group Behavior.

SIMULAÇÃO DO COMPORTAMENTO INTRA E INTER GRUPOS USANDO AMBIENTES VIRTUAIS SEMÂNTICOS

RESUMO

Simulação de situações cotidianas da vida real pode ser uma ferramenta muito útil em aplicações de entretenimento e sistemas de treinamento. Essas aplicações, como jogos ou filmes de Computação Gráfica, normalmente precisam ter ambientes virtuais povoados com agentes virtuais autônomos. Geralmente, os agentes precisam ser aptos a evoluir no seu ambiente, evitando colisões com obstáculos e outros agentes, além de interagir com outros personagens a fim de reproduzir simulações realistas. Este trabalho apresenta um modelo para simular o comportamento de grupos de forma coerente, baseado em modelagem procedural e ambientes semânticos. O foco principal é simular agentes conectados ao ambiente virtual que eles estão evoluindo, que é principalmente aplicado em segundo plano nos jogos ou filmes, gerados com pouca ou nenhuma intervenção do usuário.

Palavras-Chave: Simulação de Multidões; Humanos Virtuais; Comportamento de Grupos.

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
BVP	Boundary Value Problems
CG	Computer Graphics
FPS	Frames Per Second
GPU	Graphics Processing Unit
HDR	High Dynamic Range
PC	Population Class
ROI	Region Of Interest
RRT	Rapidly-exploring RandomTree
SVE	Semantic Virtual Environment
VE	Virtual Environment
VH	Virtual Human
VHLab	Virtual Humans Laboratory

SUMMARY

1. INTRODUCTION

The modeling and simulation of crowds have been studied in many different areas of science, due to its several applications. The models to simulate crowds can be used in the industry of entertainment, in order to realisticaly simulate the movement of a great number of virtual humans. These models can be used for populating immersive virtual environments in order to raise the feeling of presence, by simulating the movement of crowds, in order to evaluate complex environments of difficult locomotion for large populational concentrations, among other applications [10].

Human beings usually group with others in ordinary situations. When grouped, individuals interact with others according to their type of relationship, as well as the environment characteristics. Studies about human behaviors are produced since the $20th$ century [7, 44]. The goal of such studies is to identify, for example, the distribution of individuals participation in small groups and also to analyze their interactions. Simulations of virtual agents interacting with others in a virtual environment can be applied in different areas, such as entertainment (animation, movies, computer games), engineering and security (training systems).

A group can be formed by two people who are in the same space in a given situation. As the number of group members increase, a crowd can be formed. To Sighele [43], a crowd is a heterogeneous and inorganic aggregate of people. Heterogeneous because it is composed by individuals of all ages, both genders, different classes and social conditions, degrees of morality and culture; and inorganic, because it is formed suddenly, without prior arrangement and improvisation.

Indeed, the great majority of existing studies investigated a crowd as a collection of isolated individuals, each having its own desired speed and direction of motion, as in [34, 18, 22, 1]. In practice, however, it turns out that the majority of pedestrians actually do not walk alone, but in groups [8, 2, 21]. We note that the term ’group’ is used here in its sociological sense [15], that is, not only referring to several proximate pedestrians that happen to walk close to each other, but to individuals who have social ties and *intentionally* walk together, such as friends or family members. In particular, the duration of the interaction and the communicative setting distinguish from an occasional agglomerate [15].

The group behavior can be seen clearly in environments which the density of people is relatively low, because the individuals have space to move freely, without any competition for space between the agents. However, as the density of the environment raises, the groups characteristics will gradually being lost, and in case the density is very high, these characteristics may become invisible in the crowd. Nowadays we can find in literature different approaches that aim to model different group behaviors, e.g. structured steering behaviors, as briefly described in Chapter 2.

1.1 Goals

In this work, we propose a model for simulating human groups behavior, which uses the recent method proposed by Bicho *et al.* [11], that was inspired in a biological algorithm, based on competition for space in a coherent growth of veins and branches [42]. Since this original model presents free-of-collision motion, we used such method to provide collision avoidance in our method as well. Our method also includes the detection of possible interactions between agents in order to create new groups during the simulation.

Therefore, our main goals are:

i. to enable group behaviors with fewer user interventions (agents characteristics are created as a function of environment and time); and

ii. to provide a strong connection between groups of virtual agents and the environment, i.e. world can be used to change the group behaviors.

Simulations automatically generated using our technique allow the animator to be focused in the big picture and in the first plan characters [46]. Thus, our model can be applied in games in order to coherently populate the environments, such as buildings, parks and sidewalks.

1.2 Organization

This document is organized as follows: related work are described in the next chapter. Chapter 3 contains a detailed description of the proposed model. Results are shown in Chapter 4, where we explain our prototype and its features in Section 4.1, while simulation results are presented in Section 4.2, Section 4.3 and Section 4.4. Chapter 5 explains important topics for improving our model as well final considerations are discussed.

2. RELATED WORK

Although crowd behavior is a research subject studied for a long time [3], proposals of computational models to simulate them are relatively recents (particularly about groups behavior), partially due to the technological restrictions of the past. The present work have focus in medium and low density situations, where the group behaviors are still visible, because, as the density raises, the characteristics of the groups are diluted in the crowd. In this chapter we present some related work to facilitate the understanding of our model. Next sections include work related to crowd simulation and group behaviors.

2.1 The *BioCrowds* Model

Bicho [10] developed a model named *BioCrowds* that aims to simulate crowds considering characteristics that are present in crowd dynamics. The *BioCrowds* model is based on a biological approach of space colonization proposed by Runions [41], in order to simulate the growth of veins in leaves of plants. Bicho represents the free spaces in a virtual environment through a set of dots called markers, analogously to the plants auxins. These markers are treated like resources which the agents compete for. Also, Bicho's model [10] is based on the *proxemics* concept created by Edward Hall [14], which is the study of measurable distances between people as they interact. The specification of such distance can be based on different parameters: the agents relationship, the environment, the density of characters, among others.

The proxemics identification is based on the set of markers that are closer to an agent than any other. Based on this, the model defines an perception area that circumscribes the agent, allowing it to view the distributed markers in the virtual environment and identify which ones are contained in its proxemics. Figure 2.1 denote the proxemics concept in *BioCrowds*: the agents are represented by small colored squares inside a space perception area. The agentes and its allocated markers are represented by the same color.

Once identified the markers inside the personal space, the next step of the agent will be calculated in order to achieve its goal. The movement of each agent is calculated iteratively, and for each iteraction the position and vector, that indicates the direction of the agent (toward the goal), are updated. Figure 2.2 present simulations generated using the *BioCrowds* model, where in image (A) we can see the agents and their allocated markers, connected to the agent through a line segment. Image (B) shows the agents with a sort of circle - calculated using the *Convex Hull* algorithm [37], which represents the personal space [10].

Cassol [5] proposes the integration of the *terrain reasoning* concept to *BioCrowds*

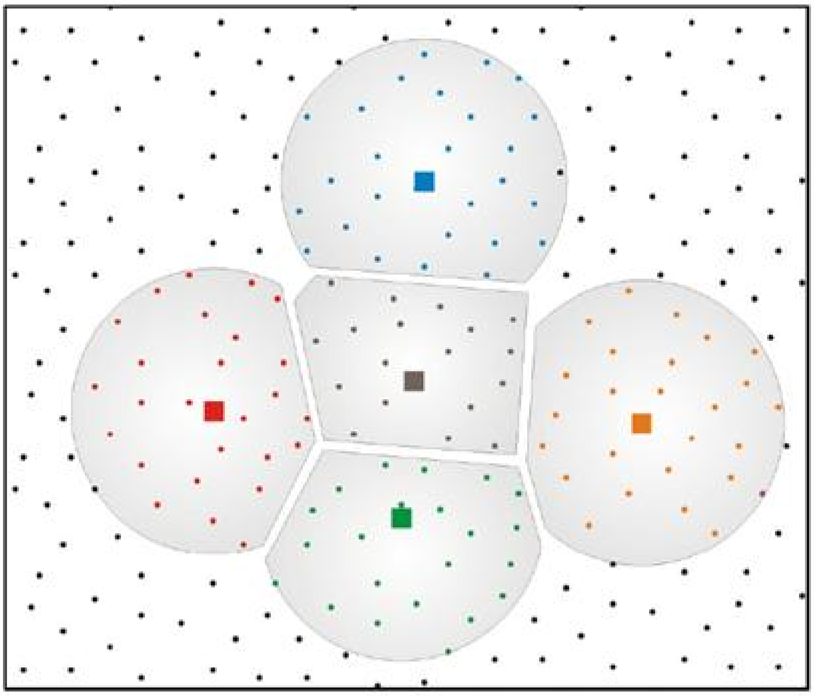


Figure 2.1 – Representation of 5 agents and their respectives personal spaces. The markers inside the personal space of each agent are represented by different colors according to the agent [10].

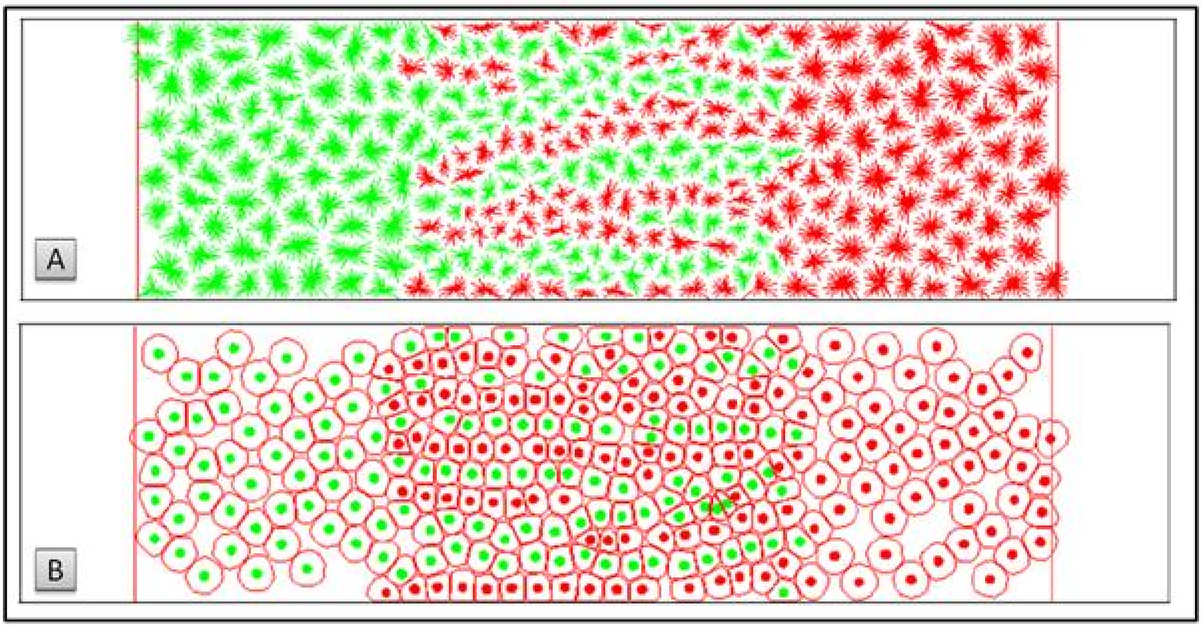


Figure 2.2 – Simulation results generated in *BioCrowds*: image (A) represents the agents and their allocated markers, connected to the agent through a line segment. Image (B) shows the agents with a sort of circle - calculated using the *Convex Hull* algorithm [37], representing the personal space [10].

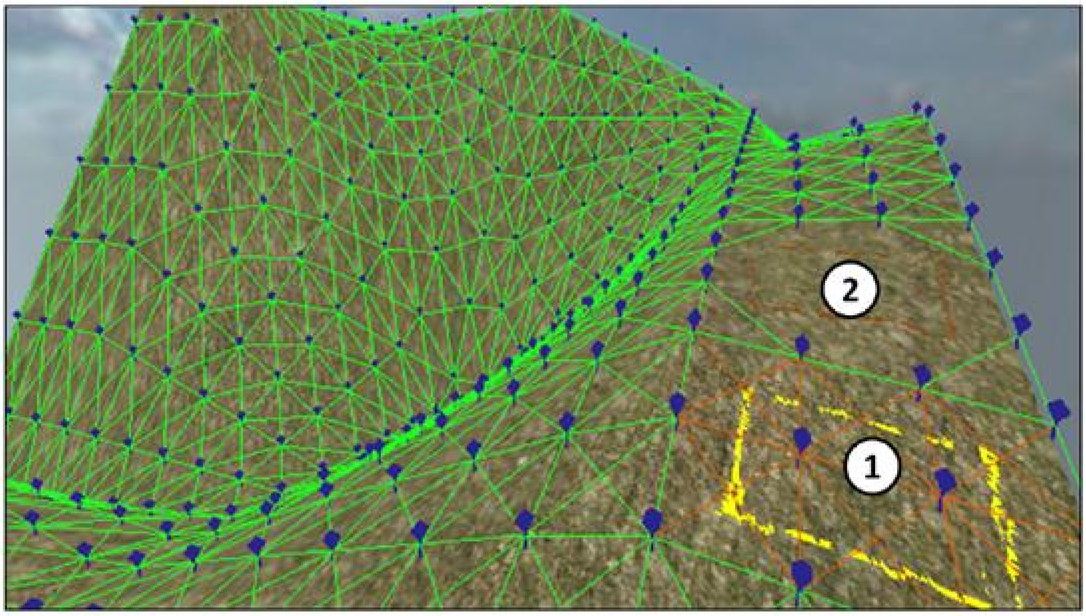


Figure 2.3 – Graph overlayed on the terrain. The graph was generated according to the presence of semi-walkable (1) and non-walkable (2) regions [5].

model, considering semantic information contained in the terrain to generate the pedestrians movement. For this, the author proposes the distribution of different types of markers in the environment, which represents walkable, semi-walkable and non-walkable regions. A non-walkable region mean a region where the agents' movement is not allowed, as well as a semi-walkable region could allow locomotion in particular situations.

Besides, Cassol [5] proposes the use of *path planning* algorithms, like A* [16]. These algorithms support the agents' movement, because they allow the generation of a route to be pursued during their move. The algorithm is executed using as input a graph originated of the terrain, which considers the presence of the different types of regions, i.e. these regions influences the edges creation and weighting. For example, the edges located in a region that contains semi-walkable markers will be heavier/expensive than the ones located in a walkable region. Similarly, in non-walkable regions no edges will be created, once the agents should not walk through these regions. Figure 2.3 ilustrates an example of graph used as input to the execution of A* algorithm. Given a initial and a final node, the algorithm returns a list of nodes that compose the path to be pursued by each agent. Each node contained in the path represents a partial objective to be reached by the agent, until it achieve the final goal.

2.2 Group Behaviors

Several aspects of group behaviors have been analyzed in the last years. Results of groups behavior analysis provide an useful reference for simulation/animation of groups and

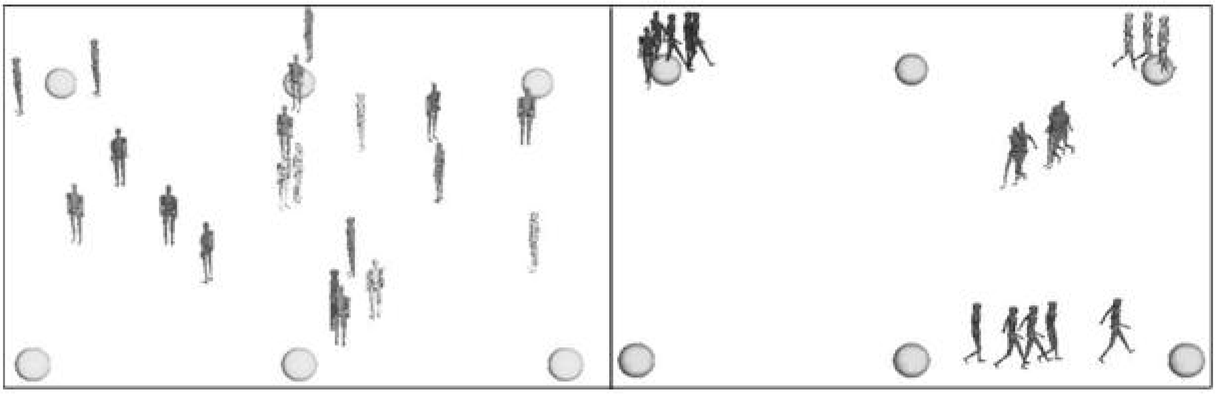


Figure 2.4 – Initial population visiting a museum (left). The virtual humans are in their initial position, which were automatically generated. After a hundred interactions, the agents gather in groups accordingly to their reactions and walk around the museum (right) [36].

crowds [19, 20, 12, 17, 45]. Two important aspects that guide the motion of real people are: goal seeking, which reflect the target destination of each individual; and the least-effort strategy, reflecting the tendency of people to reach the goal along a path requiring the least effort [45].

Musse and Thalmann presents in [36], a model to simulate the crowd behavior considering the relationship between groups of individuals and the emergent behavior originated by this relationship (i.e. the global effect generated by local rules). Based on sociological aspects, the authors treats the individuals as autonomous virtual humans that reacts in the presence of other individuals and changes their own parameters according to the reaction. Two scenarios where simulated by the authors: the first is a sociogram, that is, a sociological graph representing a population, their relationships, and the different levels of influence; the second scenario simulates people visiting a museum (showed in Figure 2.4).

More specifically, concerning the motion of groups, Kamphuis and Overmars [24] introduce a two-phase approach, where a path for a single agent is generated by any motion planner. Then, a corridor is defined around the path, where all agents stay inside. Lavalle [28] introduced the concept of a Rapidly-exploring Random Tree (RRT) as a randomized data structure for path planning problems. An RRT is iteratively expanded by applying control inputs that drive the system slightly toward randomly-selected points.

Champagne and Tang [6] presents an approach for crowd simulation using bidimensional Voronoi diagrams for the localization of groups of agents in a virtual environment. Using a GPU, the polygons of the Voronoi diagrams are calculated from algorithms as the polygon scan and the z-buffer depth. In the model proposed by the authors, the region of a polygon is associated to the centroid of a group. Each agent have a circumscribed circle in order to verify possible collisions with other agents and with static obstacles. Figure 2.5 shows the result of a simulation using Champagne e Tang's model.

Figure 2.5 – Grouped agents, avoiding collision with other agents and groups [6].

Lien and collaborators [30] proposed ways for using roadmaps to simulate a type of flocking behavior called shepherding behavior in which outside agents guide or control members of a flock. Data-driven models are quite recent in comparison with other methods, and aim to record motion in a pre-production stage or to use information from real life to calibrate the simulation algorithms. Another method using data-driven technique were proposed by Lee *et al.* [29]. The authors recorded the movement of a crowd of humans from an aerial view using a camera, extracted the movement trajectories in 2D for each individual, and then a model of agent is learned from the observed trajectories. The model of agent decides each agent's action based on the environment characteristics and the nearest agents. Once the model of agent is learned, the virtual crowd can be simulated, as shown in Figure 2.6. This figure illustrates the patterns learned form real videos (on top) and the respective simulations (on bottom): figures in left shows a pattern with group formations and individuals, while figures in right shows a line formation pattern.

Metoyer and Hodgins [32] proposed a method for generating reactive path following based on the user's examples of the desired behavior. Dapper *et al.* [9] proposed a path planning model based on a numerical solution for Boundary Value Problems (BVP) and field potential formalism to produce steering behaviors for virtual humans. Rodriguez *et al.* [39] proposed a heuristic approach to plan an environment with moving obstacles using dynamic global roadmap and kinodynamic local planning. Kallmann and Mataric [23] proposed dynamic roadmaps for online motion planning in changing environments. When changes are detected in the workspace, the validity state of affected edges and nodes of a precomputed roadmap are updated accordingly.

Recent works aim to produce coherently and realistically group behaviors considering

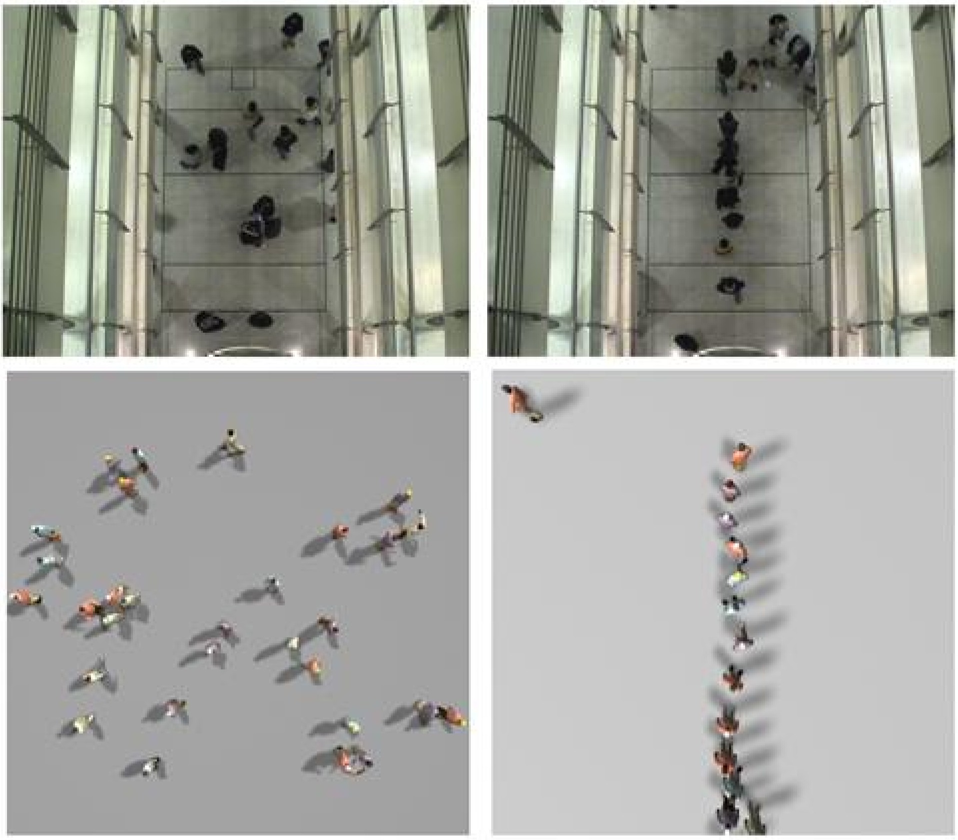


Figure 2.6 – On top, different patterns of interaction learned from real videos. Figures on bottom shows the respective simulations, based on the model of agent learned [29].

steering and groups formation. The study of Moussaid *et al.* [35] revealed through empirical analisys that more than 70% of the people actually walk in groups, for being friends, relatives, co-workers, acquaintances...i.e. people that have some social link and intentionally walk together. In this work, the authors analyses the movement of 1500 groups of pedestrians in regular conditions, showing that the social interactions between the group members generates typical patterns of movement that influences the crowd dynamics. Besides, the work presents a model based on social forces - originally proposed by Helbing [18], which are responsible for maintain the groups formations.

In low density situations, the group members tend to walk side by side, forming a perpendicular line with the movement direction. Although, as the density raises, this linear formation starts to tilt, transforming it to a pattern called *V-Like*, in case of a group of 3 pedestrians. When the pedestrians density is too high, the formation structure named *River-Like* occurs (See Figure 2.7).

Karamouzas and Overmars [25] work was inspired in the social forces model proposed in [35] to simulate the behavior of small groups of pedestrians, where the velocity space for planning the avoidance maneuvers of each group is used to maintain a configuration that facilitates the social interactions among the group members. The focus of this work is the

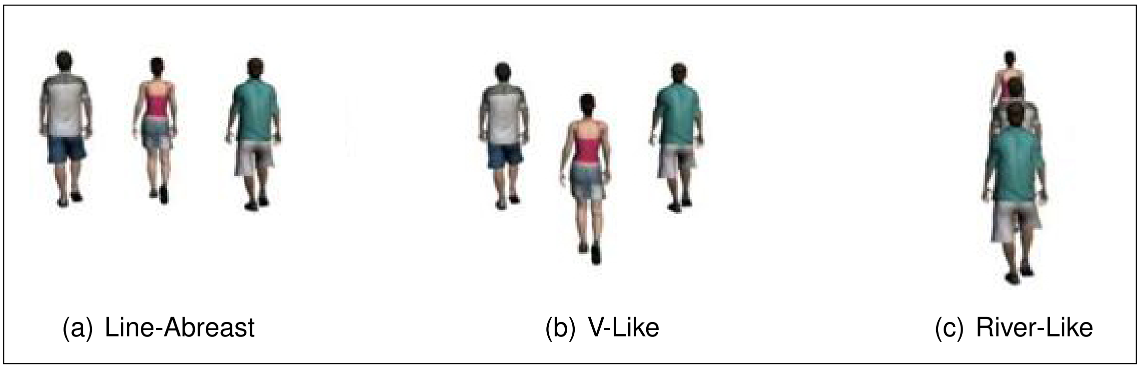

(a) Line-Abreast (b) V-Like (c) River-Like

Figure 2.7 – Group formations, according to Moussaid *et al.* [35]. The *Line-Abreast* (or *Side by Side*) formation (a) is adopted for low density situations, while the *V-like* (b) and *River-Like* (c) patterns may appear if the density raises [25].

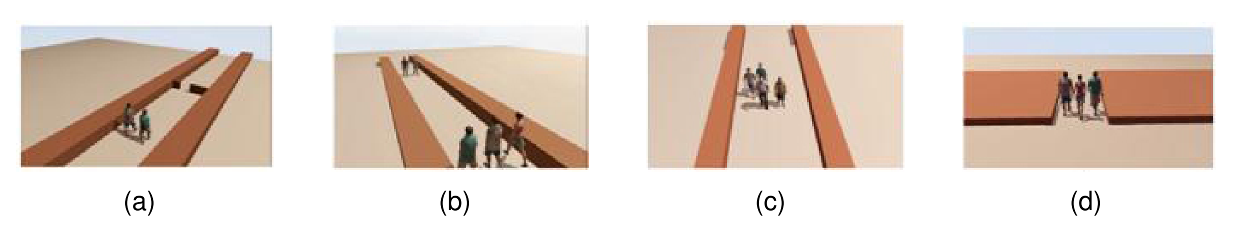

(a) (b) (c) (d)

Figure 2.8 – Results of simulation scenarios presented in the work of Karamouzas and Overmars [25]. For details see text.

local behavior of these groups, i.e. how the group members interact among themselves and other groups. They argue that this approach works for a large variety of scenarios, as shown in Figure 2.8, which demonstrates the results of four simulation scenarios. In Figure 2.8(a) a group must adapt its formation to pass through a door; Case of Figure 2.8(b) simulates interactions between groups in a confined environment, while in Figure 2.8(c) shows a group overtaking another. A group of 3 agents walking through a narrow corridor appears in Figure 2.8(d).

Kendon [26] presents a study that shows how participants may jointly establish and maintain a spatial-orientational system, named in his research as *F-formation*, used to preserve the integrity of their occasion of interaction. Through an extensive observational study, Kendon provide several elementary observations on how people employ space, bodily orientation and positioning as a means of organizing the attentional structure of social encounters. In Figure 2.9 we have two formations evinced in Kendon's research: on left, an L-shaped dyadic F-formation in which the participants are positioned creating a right angle. The image on right shows two people oriented face to face, featuring a Vis-a-vis F-formation. Besides, the study shows that for more than two participants, the groups' arrangement tend to be circular with the members oriented approximately to its center.

Groups simulated by our proposed model also keep formation, however, based on environment restrictions (e.g. obstacles, density of agents), group formation is adapted

Figure 2.9 – An L-shaped dyadic F-formation (left) and a Vis-a-vis F-formation (right) [26].

in order to provide the best efficiency as possible for social interactions [35]. An important contribution from our model is the connection between the population and the semantic environment, which can constraint the motion behavior. Next chapter explains our model in details.

3. THE MODEL

Our model is mainly focused on the groups behaviors when evolving in a virtual environment regarding other groups location, density of agents in the space and environment characteristics (obstacles, interest locations, etc). It is important to stress that our model is suitable for background actors and actions, requiring minimum intervention of designers or users.

The model is composed by four modules as illustrated in Figure 3.1. The Environment module is responsible to generate the environment structure to be used in the Simulation module, i.e. floorplan, spaces and objects, and it is also responsible to provide certain definitions for the Virtual Population module (e.g. objects of interest contained in each space). The Virtual Population module creates the population definitions based on the characteristics of the environment. Finally, the Simulation module is responsible for providing agents motion and their interactions into the environment. This module generates a file containing all agents position in each frame/time, which is used as input to visualize the simulation in the Visualization module. We emphasize that the present work is focused mainly on Virtual Population and Simulation modules - showed in blue by Figure 3.1, so we use tools from previously developed work for the Environment and Visualization modules. Details about each module will be provided respectively in the next sections.

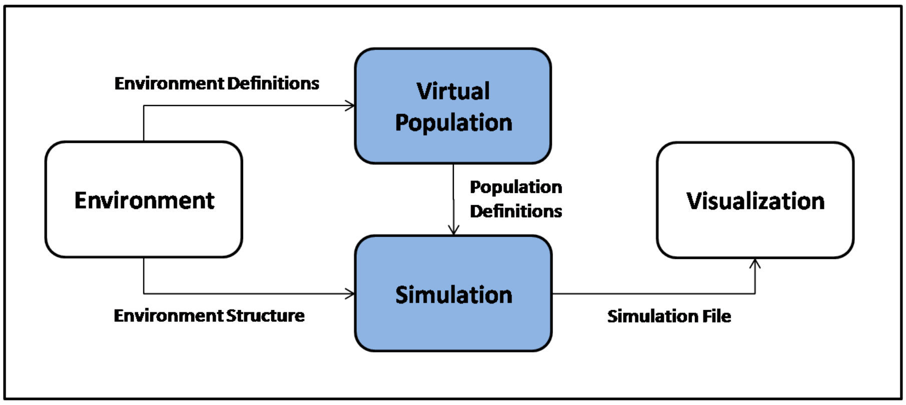


Figure 3.1 – Overview of our model.

3.1 Semantic Environment

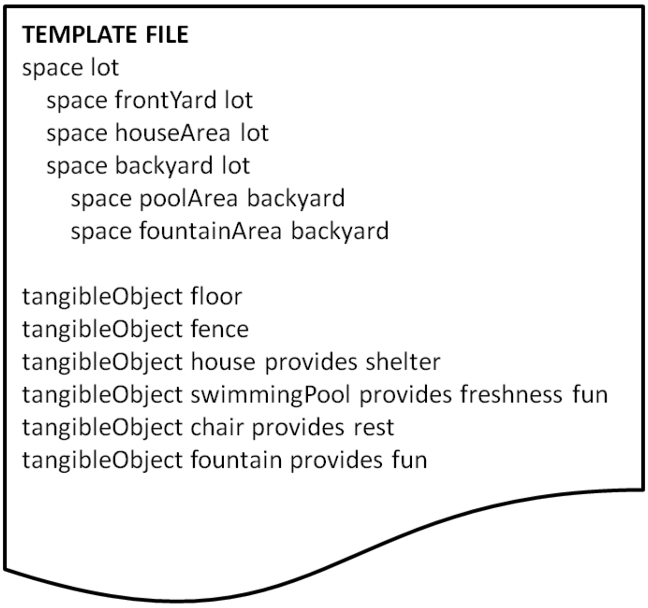
A Semantic Virtual Environment (SVE) [38, 13] is a virtual environment that is populated with entities enriched with semantics. A simulation environment is a complex space that is composed by a hierarchical set of simpler spaces, such as a city. Commonly, several neighborhoods composes a city, which are composed by many lots. These lots might have several types of buildings, with different types of rooms: kitchen, bedroom, bathroom, living room, among others. To specify goals in the environment we can assign a special attribute to any object that indicates some interesting thing or a resource that a certain object can provide. For instance, a TV provides *fun* as a chair provides *rest*. If an object that provides some resource is placed inside of a space, this space will provide that resource as well.

In this work, we use a Semantic Engine based on [27] and previously developed by [31] in order to specify, create and store all spaces and objects. Figure 3.2 shows an example of files used to represent the semantic environment. The environment specifications are made by a *template file* that contains the space hierarchy and all objects that will be instantiated in our environment. In the case of Figure 3.2(a), *lot* is the main space and it contains other three spaces: *frontYard*, *houseArea* and *backyard*. We have two more spaces, *poolArea* and *fountainArea*, which are inside of *backyard* space. At the end of this file we declare all tangible objects and their resources. After have defined these conceptual spaces and objects, we create instances that will have geometric representations in the environment. The space instance file (Figure 3.2(b)) contains information about the space type (open space or closed space), its geometry, the position of each space, and, if this space is a closed space, its entrances and windows. In the *space instances* definition (Figure 3.2(b)), we use *m* to specify the main space named *myLot* in this case. We use *o* to specify an open space, which means that there are no walls surrounding such space. If we desire walls, we use *c* to declare a closed space instead of an open space. After the specification of spaces and objects, it is possible to create one or more instances for each of them. The object instance file (Figure 3.2(c)) describes the type, name, geometry file, space it belongs, position and rotation respectively for each object.

Once defined the environment, the next step is to create information about how to populate these spaces and what agents can do during the simulation. For that, we create a *Population Class* (PC), which can automatically create a random population for a specific environment or still deal with a population script defined by the user as will be described in Section 3.2. As output of the SVE Generator we have a 2D layout that contains goals, walkable and non-walkable regions and a graph that will be used by the virtual agents to compute their paths - similar to the work of Cassol [5] presented in Section 2.1. Such graph includes the nodes into the walkable regions and the goals definition (automatically defined close to objects). As consequence, agents will be able to walk in the space. Another output is a 3D scene, that contains all 3D geometric representation of objects and spaces.

3.2 Virtual Population

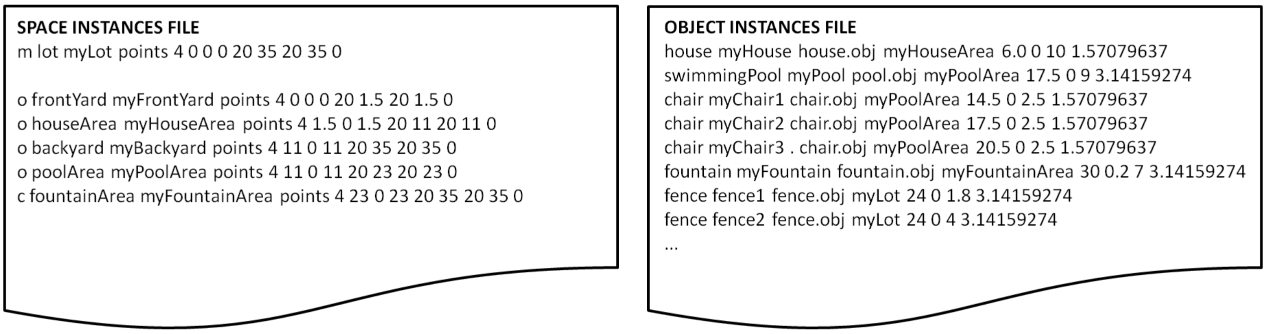
This section describes how agents are generated in the VE, considering the PC that is composed by the following information:



```
TEMPLATE FILE
space lot
  space frontYard lot
  space houseArea lot
  space backyard lot
    space poolArea backyard
    space fountainArea backyard

tangibleObject floor
tangibleObject fence
tangibleObject house provides shelter
tangibleObject swimmingPool provides freshness fun
tangibleObject chair provides rest
tangibleObject fountain provides fun
```

(a) Template file.



```
SPACE INSTANCES FILE
m lot myLot points 4 0 0 0 20 35 20 35 0

o frontYard myFrontYard points 4 0 0 0 20 1.5 20 1.5 0
o houseArea myHouseArea points 4 1.5 0 1.5 20 11 20 11 0
o backyard myBackyard points 4 11 0 11 20 35 20 35 0
o poolArea myPoolArea points 4 11 0 11 20 23 20 23 0
c fountainArea myFountainArea points 4 23 0 23 20 35 20 35 0
```

```
OBJECT INSTANCES FILE
house myHouse house.obj myHouseArea 6.0 0 10 1.57079637
swimmingPool myPool pool.obj myPoolArea 17.5 0 9 3.14159274
chair myChair1 chair.obj myPoolArea 14.5 0 2.5 1.57079637
chair myChair2 chair.obj myPoolArea 17.5 0 2.5 1.57079637
chair myChair3 . chair.obj myPoolArea 20.5 0 2.5 1.57079637
fountain myFountain fountain.obj myFountainArea 30 0.2 7 3.14159274
fence fence1 fence.obj myLot 24 0 1.8 3.14159274
fence fence2 fence.obj myLot 24 0 4 3.14159274
...
```

(b) Space instances file.

(c) Object instances file.

Figure 3.2 – The set of script files provided by the user to be used as input for our model [31]. For further information see text.

i. the global seed and the simulation total time;

ii. the higher density of agents to be attained during the simulation and the time it should occur. In this case, the simulation process is responsible for creating and destroying the agents (e.g. at the beggining and ending of a party) in order to attain the expected density of agents at a specific period of the simulation. The density of agents is specified as agents per m^2, and we defined $LOW\_DENSITY < 1\ agent/m^2$, $MEDIUM\_DENSITY < 2\ agents/m^2$ and $HIGH\_DENSITY >= 2\ agents/m^2$;

iii. groups distribution presented in a certain population, i.e. how many agents are not grouped (individuals) or grouped in groups of 2 or 3 agents;

iv. the distribution of interests (where agents should go), spawn locations (where agents should be created) and kill locations (where agents should go to be removed from the simulation).

It is also possible to define specific populations P informing data in a template file including PC definition, as showed in Listing 3.1. Next sections explain how agents and groups are created during the simulation.

```
POPULATION CLASS
seed=500                             //Global seed
simulation_total_time=2000           //Simulation total time, in frames
higher_density=HIGH                  //The higher density the simulation should have
peak_time=400                        //The time that the peak of density should occur, in frames

Groups_distribution:
one_agent=0.36                       //Percentage of individuals
two_agents=0.36                      //Percentage of groups of two members
three_agents=0.28                    //Percentage of groups of three members

Interest_resources:
fun=0.4                              //Percentage of agents that seek for the "fun" resource
freshness=0.3                        //Percentage of agents that seek for the "freshness" resource
shelter=0.1                          //Percentage of agents that seek for the "shelter" resource
rest=0.2                             //Percentage of agents that seek for the "rest" resource

spawn_points=3,51                    //Node Ids of spawning locations
kill_points=3,51                     //Node Ids of killing locations
```

Listing 3.1 – An example of script provided by the user containing the population definitions.

3.2.1 Creation of Agents

Number of agents to be created agreed with the density specified in the script. From the first frame of the simulation until the expected peak of density, agents are linearly created from predefined spawn points in the environment. Once the expected density is achieved, the agents starts to be linearly removed from the simulation by steering to predefined kill points. Also, groups are pre-defined, i.e. we define which agents are part of specific groups. The spaces defined in <space instances file> as well as the objects in the <objects intance files> as described in Section 3.1 can be the goals in the simulation, and they are randomly distributed through the agents. Once one agent reaches a goal, it stays there for a random time and then a new goal is randomly chosen. Moreover, agents can group with others as discussed in next section.

3.2.2 Creation of Groups

Into a specific population P, one or more groups G can be created, i.e. agents from P that should physically interact are pre-defined, being maximum three agents in each group. We justify the decision to pre-define which agents are going to interact since emergent groups is not the focus of our work. Moreover, we also specified that groups should have maximum three agents since the major part of groups in real life are formed by maximum three people [33]. Therefore, P and G can have the same size, for example, a small family can have only one group. On the other hand, a party can be modeled as a P formed by

15 groups G, formed by maximum three people, for instance. When distance d among two members of G_i is into a range defined by Hall [14] as a social distance [1], one group G is formed. According to Bicho [11], the *personal space* for each agent A is modeled as a circular region (with radius R), that represents a "perception field" (see Figure 3.3(a)) which can be used by each agent to avoid collision with other agents. Moreover, each agent A has a radius R_i which can be equal to everyone else or randomly chosen, where $i \in [1, N_{ag}]$, being N_{ag} the total of agents in the simulation. In Bicho's model [11], this circular region is important in the sense that it is a region where any other agent cannot penetrate, providing the collision avoidance of the method. In our case, we adopt another circular region we called group space (see Figure 3.3(b)) which includes the N members of G_i, and it is computed based on the Equations 3.1 and 3.2:

$$Af = max(dist(A_i, \vec{C})), \tag{3.1}$$

$$Rg = dist(A_{Af}, \vec{C}) + \overline{R_{Af}}, \tag{3.2}$$

where $\vec{C}$ is the centroid of all A_i positions and it is also the center of G_i with radius Rg, Af is the index of the agent which position is farther from the centroid $\vec{C}$ and $\overline{R_{Af}}$ is the radius R of agent A_{Af}. The group space is important to define the group region used to avoid collision among groups, detailed in Section 3.3.3. At this point, for each environment generated as presented in Section 3.1, we have a population P formed by one or more groups G which are formed by N members, being maximum three. Moreover, G has a circular region of radius Rg which determines the group space. When agents are grouped they have equivalent goals and speeds in order to stick together. The grouping process is showed in Figure 3.4, illustrating the moment when two individual agents join, creating a single group entity.

It is important to highlight that the groups features are maintained just while group members are still together, i.e. all members should be inside a specified distance treshold between each other to be considered as linked. We consider that a group is linked when its members keep certain distance treshold T_{Lk} from each other. For example, in a group containing two members, if the distance between them is bigger than T_{Lk}, the group structure is lost. Otherwise, the group is considered as linked. In next section, we describe how all this data is used in our simulations providing the behavior of individuals and groups.

---

[1] Close phase is attained when distance is from 4 to 7 feet (1.2 to 2.1 m) and Far phase is 7 to 12 feet (2.1 to 3.7 m).

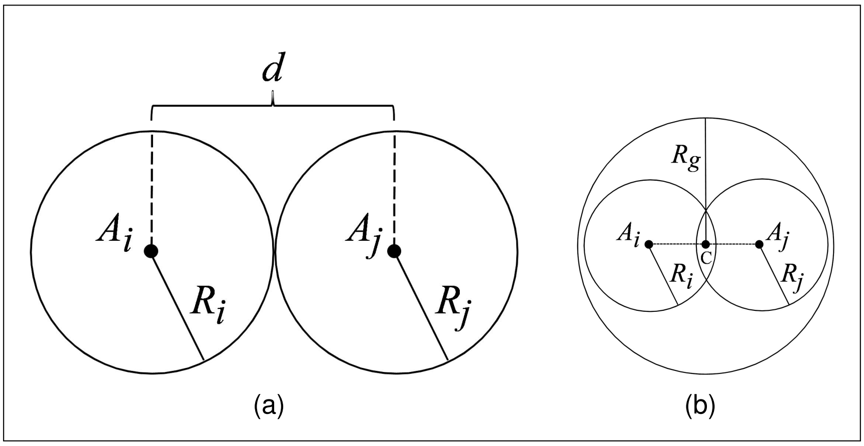


Figure 3.3 – a) Hall's [14] social distance (d) between two agents. b) If the agents are from same group G, a group space is defined representing the group as an unique entity.

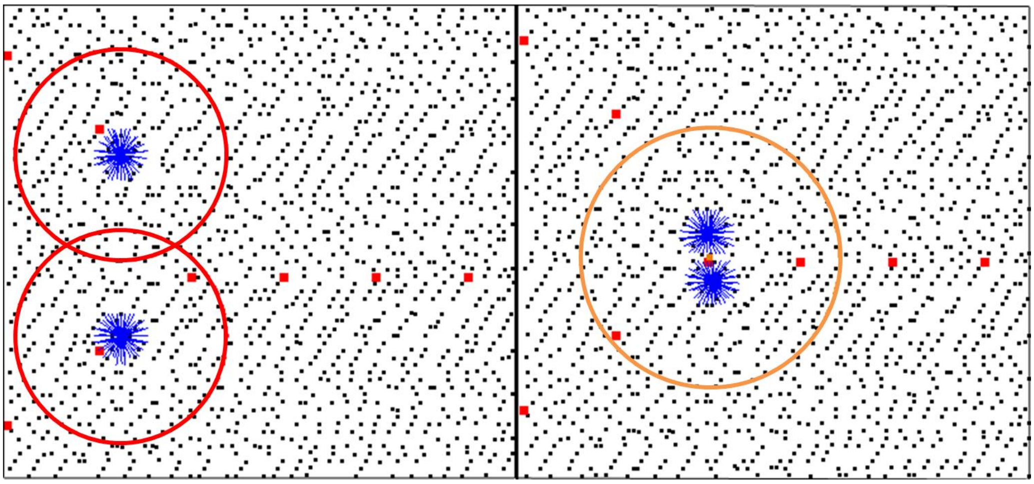


Figure 3.4 – Illustration of the meeting of two agents, predefined as friends/acquaintances. On the left, the intersection of individual perception fields, in red. The agents then group, creating a single perception radius for the group in orange, on the right.

3.3 Simulation

This phase is responsible for providing groups and agents motion and their interactions into the environment. Each group G_i can evolve in a virtual environment, which can also be populated by other groups G_j. The following aspects are considered in order to define the individual behavior function of an agent A which is member of G_i:

i. agent's goal which is defined based on environment information;

ii. density of agents close to G_i;

iii. obstacles and other constraints around G_i;

iv. location of close groups G_j.

We define a distance threshold for computing the obstacles and close groups for each G as $T_o = 2.Rg$ [2]. It is important to note that only agents and obstacles into such threshold distance and into the environment as well are considered. It avoids to consider agents in another room or space, for instance. In the same way, spawn points are used to initialize agents positions. To reach the target, we use A* algorithm [16] to provide the best path for each agent. Moreover, the agents movement is performed inspired in the algorithm presented in [40]. As in the original model, the environment is represented by a set of markers (dots in the space) which discretizes the space. Overlayed to the markers, we create a grid of nodes in the space where motion is allowed (as explained in Section 2.1) and used as reference to the A* path planning algorithm, considering environment features [16, 5].

While the agents move from their initial location to their target by using the nodes in the environment, at each time step we verify the presence of other agents (to be aggregated in a group), obstacles and other groups (to avoid collision). Basically, two group behaviors emerge from this connection between agents and environments. Firstly, grouped agents can present groups formation while evolving in the virtual environment (intra group behaviors). Secondly, they can vary their behavior (formation and trajectories) based on environment constraints and people density (inter group behavior).

The model we used to provide agents motion [40] can create an undesired artifact in group interactions. Since the original model does not consider the group structure, the agents avoid collisions with other always considering that the agents are moving alone. For instance, in real life non-dense crowds, groups tend to keep together avoiding collision with other groups by avoiding the group areas [33]. In order to avoid that groups occupy the same space (i.e. a group space penetrates another group space) we keep a small region inside the group space which markers cannot be used for agents from a different group. As a consequence, groups avoid collision with each other and no interpenetration happens in groups spaces. This behavior becomes a problem if the crowd is dense. In this case, depending on the density around the group, this behavior is turned off in order to preserve crowd dynamics, and also because groups behavior are not visible in dense populations.

After have defined the members in a group, we are able to compute the physical agents position into the group space to determine the *group formation*, detailed in Section 3.3.1. Section 3.3.2 shows details about group formation adaptation considering obstacles in the environment, while Section 3.3.3 provides information about the method used for collision avoidance between groups. Finally, Section 3.3.4 explains how conversational groups are created in our model.

---

[2] This distance is sufficient for the group to perform the avoidance maneuvers, according to our tests, and it can be easily changed in a configuration script.

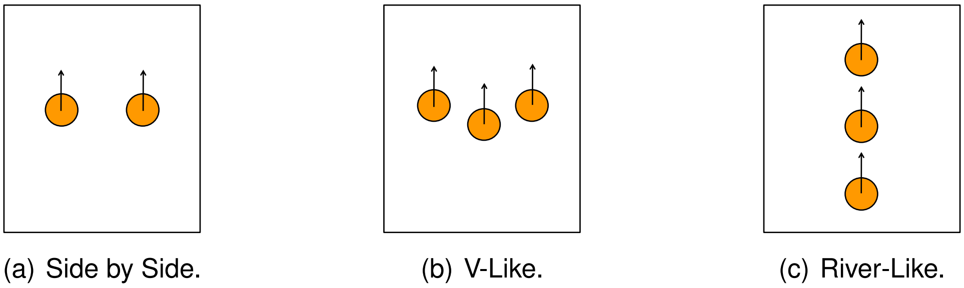

(a) Side by Side.
(b) V-Like.
(c) River-Like.

Figure 3.5 – Group formations to be performed by our model, based on the work of Moussaid [35].

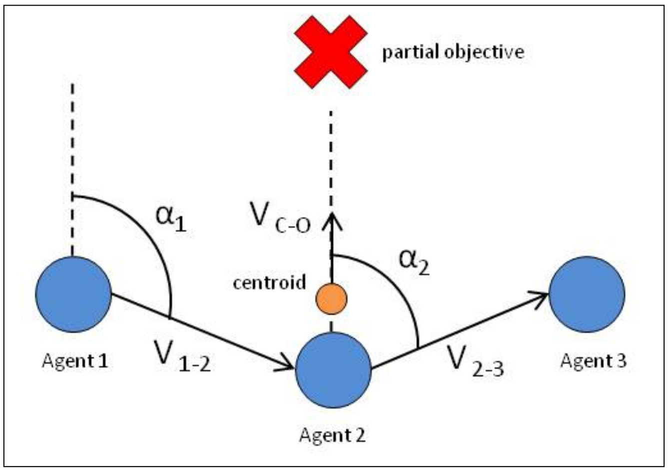


Figure 3.6 – Sketch of vectors and angles between agents in a group, based on the work of Moussaid *et al.* [35].

3.3.1 Group Formation

As previously presented, each group G has a circular region which represents the group space with radius Rg. In such area, we can provide three formations, inspired in the work proposed by [35] (explained in Section 2.2) and illustrated in Figure 3.5.

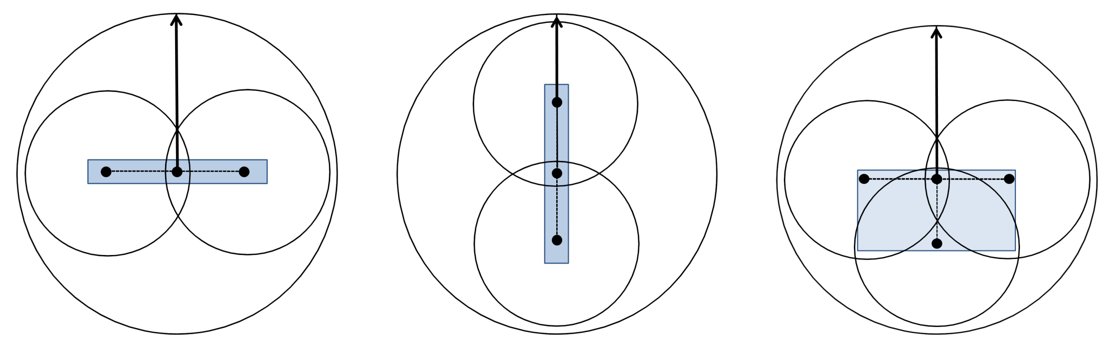
The *River-Like* formation can be considered an emergent behavior in our model (see Figure 3.5(c)). Considering the agents with the same goal, being part of the same group, they are able to move in the same direction at the same speed, as in [40], emerging such formation. To provide the *Side by Side* formation (Figure 3.5(a)), we perform a simple test of angles in order to keep the agents aligned and perpendicularly placed given their goal. As skecthed in Figure 3.6, the angles for groups with 3 members are based on the vector formed by the centroid and the next partial objective ($\vec{V}_{C-O}$), and the vector between an agent and its closest neighbor on the right-hand side ($\vec{V}_{1-2}$ in case of $Agent_1$). According to Moussaid *et al.* [35], the angle α_1 should be around $97.8°$, and angle α_2 around $87.1°$ for *Side by Side* formation. In the other hand, for *V-Like* (Figure 3.5(b)), these angles float around $107.9°$ for α_1 and $70.6°$ for α_2. When α_1 is greater than the specified value, we slightly decrease $Agent_1$'s speed, and if it is smaller, we slightly increase its speed. The same rules are applied for $Agent_3$, but in reverse order. With such rules, the agents keep walking close to each other in a specific formation during the motion process for groups of 2 and 3 agents. A high-level description of the method used to maintain the group formations - explained above, is showed in Algorithm 3.1.

Algorithm 3.1 Maintain Group Formation

```
for each Group $G$ do
    for each Agent $A$ member of $G$ do
        if $A$ is on the LEFT then
            $angle \leftarrow calcAngle(vec\_centroid\_objective, vec\_A\_A_{Right-side})$
            if $angle < ALPHA_1$ then
                $increaseSpeed(A)$
            else
                if $angle > ALPHA_1$ then
                    $decreaseSpeed(A)$
                end if
            end if
        end if
        if $A$ is on the RIGHT then
            $angle \leftarrow calcAngle(vec\_centroid\_objective, vec\_A\_A_{Right-side})$
            if $angle < ALPHA_2$ then
                $decreaseSpeed(A)$
            else
                if $angle > ALPHA_2$ then
                    $increaseSpeed(A)$
                end if
            end if
        end if
    end for
end for
```

3.3.2 Formation Adaptation

In order to achieve better simulation results, we developed a method for adapting group formation according to the free space ahead the group. Our method considers the free space in the goal direction trying to find out if there is available space for the group to pass through. Indeed, we compute for each group a 2D Region of Interest (ROI) that is included in the group space and represents the minimum area that should exist in the environment for the group performance. These regions can be simply computed based on agents positions and their sizes A_S. For this, we could mathematically estimate the ROI



(a) Side by Side. (b) River-Like. (c) V-Like.

Figure 3.7 – Group formations and their respectives $ROIs$.

for the three group formations and test them against the environment space[3], computing ROI_{side}, ROI_{river} and ROI_V that are respectively illustrated by Figure 3.7. The most suitable formation is chosen based on the following tests: First test aims to keep the *Side by Side* formation, since it represents the best way, in social terms, for a group to go everywhere in low dense situations [35]. So, ROI_{side} is checked if it can be included in the region forward to the group space. For instance, if a group G containing three members is passing through a door with size smaller than ROI_{side}, then next group formation (*V-Like*) is tested against the available free space in the goal direction. If ROI_V is still larger than the free space in the environment, then *River-Like* formation is adopted. For groups of two members, if there is no space for *Side by Side* formation, then *River-Like* formation is performed. Besides that, we make an interpolation of the ROI in order to fill the gap between the current group position to the projected ROI, assuring that all the way through the passage ahead is tested. Figure 3.8 illustrates an operational example of the method, where a group of three members performing *Side by Side* formation is close to environment obstacles (Figure 3.8(a)). The algorithm detect collision between the projected ROIs and the environment bounding boxes, passing to the next test, in which the *V-Like* estimated ROI is chosen (Figure 3.8(b)). As no collision was detected for the *V-Like* ROI, the group will then assume the *V-Like* formation. Algorithm 3.2 presents the method used for adapt the formations described in pseudocode. The simulation result for this case is presented in Section 4.2.3.

3.3.3 Group Collision Avoidance

In the original model, we have an undesirable situation called local minima. This occurs when two groups have colinear trajectories in opposite directions, creating an impasse once both groups have associated all markers around them, and could not move forward, as

---

[3] Each iteration, we test collision between the ROI and the environment bounding boxes (objects in the environment).

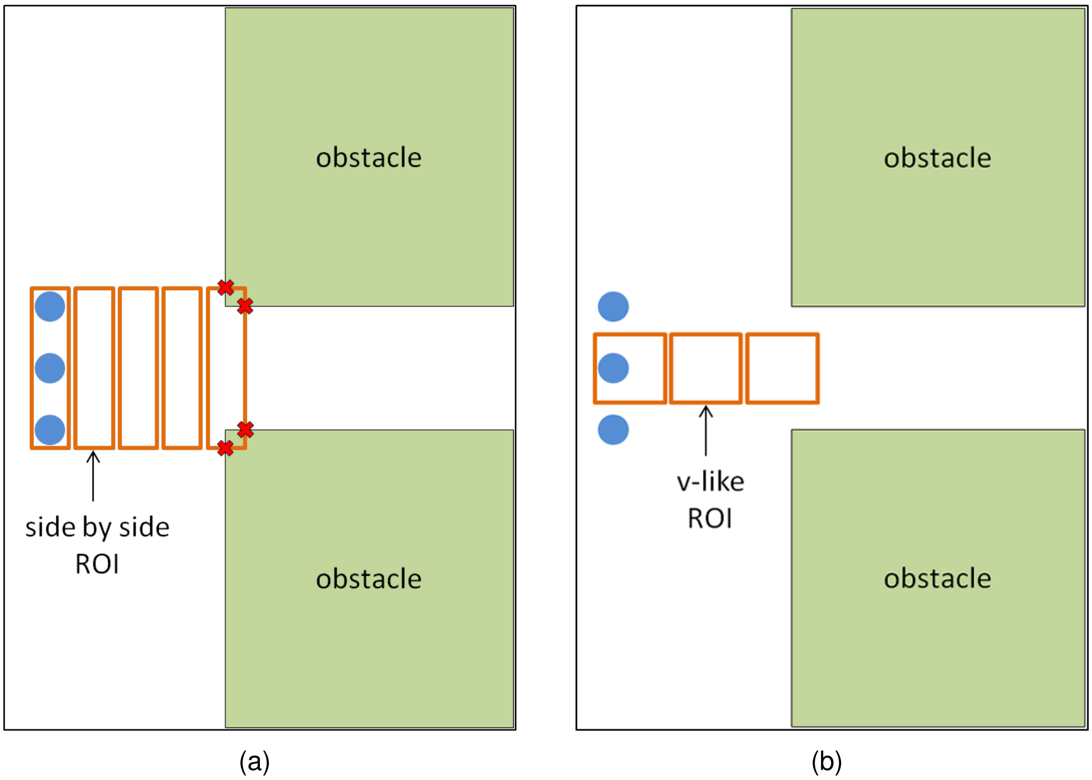


Figure 3.8 – Developed method for group formation adaptation according to free space ahead. For explanation see text.

Algorithm 3.2 Adapt Group Formation

```
for each Group $G$ do
    for each Obstacle $O$ do
        if $distance(G, O) \leq T_o$ then                                    ▷ $T_o \leftarrow 2 * G.radius$
            if $testCollision(ROI_{side})$ then                              ▷ ROI Side-by-Side
                if $G$ contains $2$ members then
                    $G.currentFormation \leftarrow RIVER\_LIKE$;
                else
                    if $G$ contains $3$ members then
                        $G.currentFormation \leftarrow V\_LIKE$;
                        if $testCollision(ROI_v, O)$ then                    ▷ ROI V-Like
                            $G.currentFormation \leftarrow RIVER\_LIKE$;
                        end if
                    end if
                end if
            end if
        end if
    end for
end for
```

shown in Figure 3.9. In order to avoid this case, we developed a method to detect possible situations of local minima and slighly change group trajectories, in simulation time, so the groups bypass each other and the impasse does not occur. If an impasse possibility was detected[4], we temporarily distribute increased weights for the edges of the next node-objective[5], and recalculate the path of one of them, randomly chosen, from the current objective. As a result, the path planning algorithm would not choose the heavier edges, selecting a path that contour the oncoming group. A high-level description of this method is presented in Algorithm 3.3. Figure 3.10(a) demonstrates the overlap of the graph in the simulation environment, illustrating the edges that would have their weights increased and the new trajectorie calculated by A* algorithm for the group in blue. Figure 3.10(b) shows the maneuver performed by the groups before and after the crossing, respectively.

Algorithm 3.3 Perform Group Collision Avoidance

```
for each Group $G_A$ do
    for each Group $G_B$ do
        if $distance(G_A, G_B) \leq T_o$ then                    $\triangleright T_o \leftarrow 2 * G.radius$
            if $detectLocalMinima(G_A, G_B)$ then
                $Group\ G \leftarrow random(G_A, G_B);$
                $Node\ N \leftarrow G.getNextGoal;$
                for each Edge $E$ connected to $N$ do
                    $increaseWeight(E);$
                end for
                $pathPlanning(G);$
                $restoreGraph();$
            end if
        end if
    end for
end for
```

3.3.4 Conversational Groups

Usually, when a group of people reach its destination, they arrange themselves spatially in order to be able to socially interact with each other, specially for establish conversations. In order to reproduce more realistic simulations, our groups also assume formations when they reach a goal, as mentioned. Based on the work of Kendon [26] (presented in Section 2.2), we adapted our model to provide *F-formations* for groups up to 3 agents, i.e. the spatial position for each agent member of a given group. The developed method is simple and it is based on the algorithm for maintaining groups formations, explained in Section 3.3.1. For

---

[4] The algorithm search for collision every time a group reaches a partial objective, considering groups inside a distance of $2.Rg$.

[5] The edges weights are altered just for recalculate the new trajectory, being restored to the original values before ending the path planning algorithm.

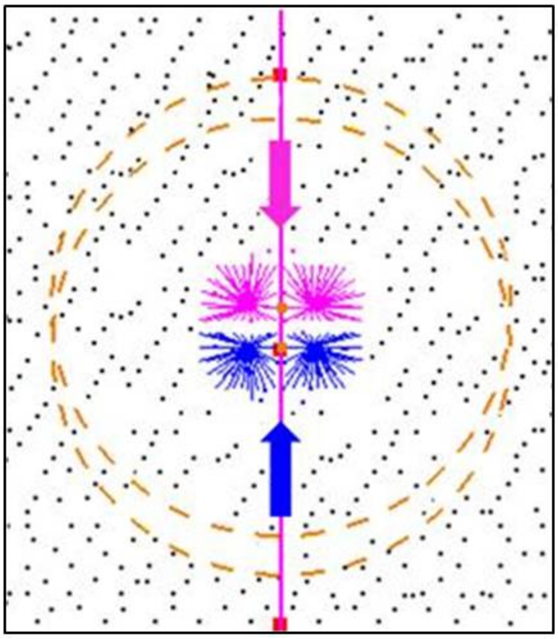
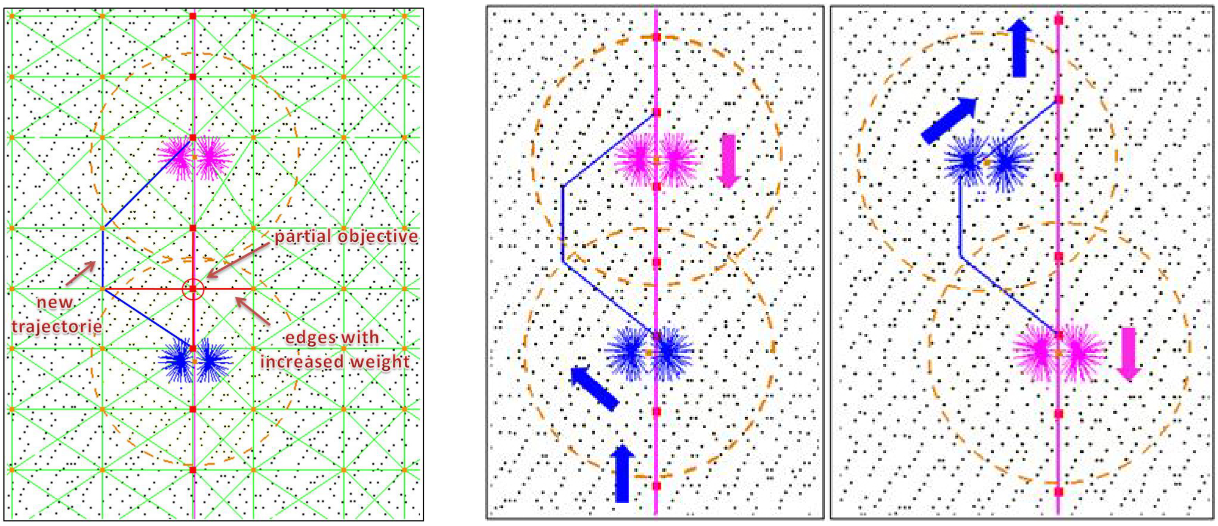


Figure 3.9 – Local minima between two groups with colinear trajectories in opposite directions.



(a) (b)

Figure 3.10 – (a) Graph overlayed to the environment, indicating the node and the edges which had weight increased, in red, and the new trajectorie for the group in blue illustrated also in blue. (b) Group in blue steer its new trajectorie that contour the group in pink, avoiding local minima.

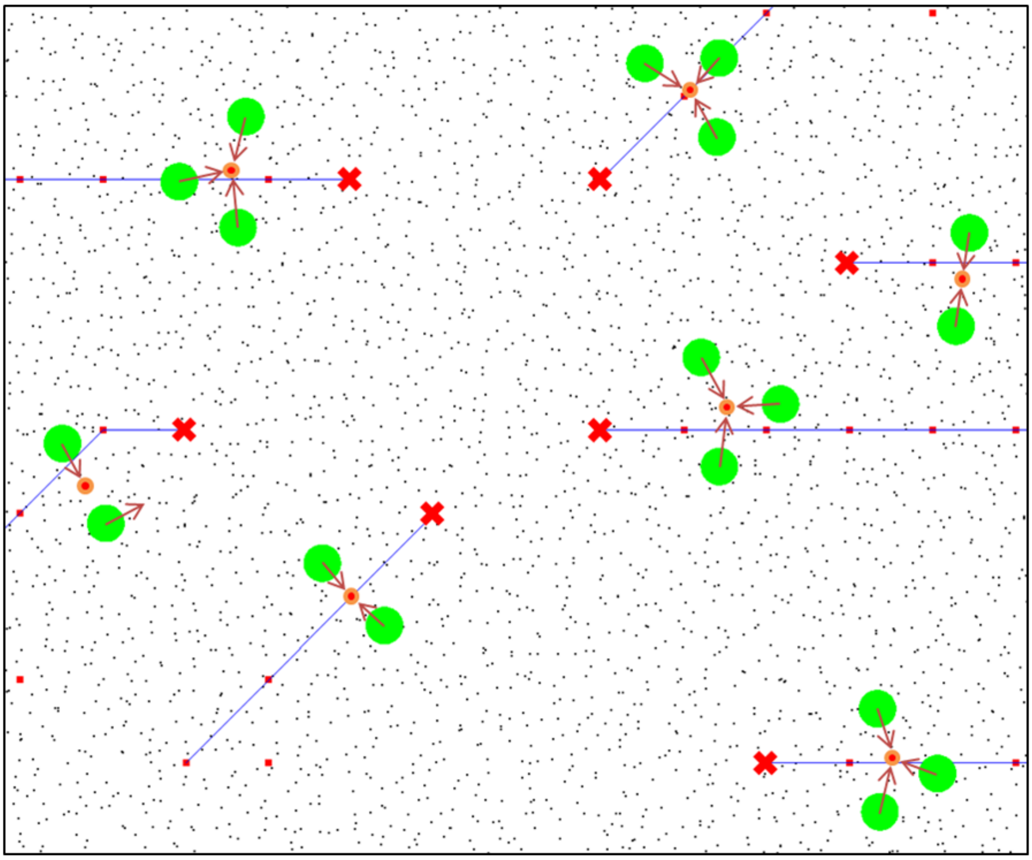


Figure 3.11 – Groups represented by green circles performing F-formations [26] after arrive in their goals.

groups of 3 members, we arrange the formation accordingly to the angles formed between the agents and the vector pointing to the next objective (see Section 3.3.1). For instance, if the group is close to its final goal, the members adapts their positions in order to keep certain angle in relation to the others, by slightly changing their speeds. This way, the group already arrive in the goal arranged in a *F-formation*. According to Figure 3.6, the angle α_1 should be around $150°$ and the angle α_2 around $30°$ (in order to create the "circle"). In case of groups containing 2 members, the *Side By Side* formation already provide the correct spacial position for a conversational group and it can assumes a *L-shaped dyadic F-formation* or a *Vis-a-Vis F-formation* (see Figure 2.9), according to their orientations.

After a group reach his final goal in a *F-formation*, we need to adjust the agents' orientation in order to visualize believable simulations. According to Kendon's study [26], humans tend to maintain their bodies oriented toward the centroid of the group when arranged in F-formations, except for the *L-shaped dyadic F-formation*, in which the members arrange themselves perpendicularly to each other. In our model, we define the agents' orientation based on the features noticed by Kendon, so the group members are oriented toward the centroid of the group. To do this, we calculate, for each member A of G_i, the angle between the vector agent-centroid and the environment horizontal axis (X axis), according to equation 3.3:

$$O_{A_i} = ang(vec(A_i, \vec{C}), \vec{H}), \tag{3.3}$$

where $\vec{C}$ is the centroid of all A_i positions and it is also the center of G_i, and $\vec{H}$ is the horizontal axis of the global coordinates system. To create the *L-shaped dyadic F-formation* (for groups of 2 members), we randomly add or subtract 90° from the O_{A_i} of one group member in order to form a perpendicular angle between them. Figure 3.11 shows an example of simulation, where several groups arrive in their goals arranged in *F-formations* and oriented toward the group centroid, indicated by the arrows in red. The orientation O_{A_i} is then stored in the simulation file to be reproduced by the visualization module, that will be presented in the next section.

3.4 Visualization

In order to better visualize the simulation results, we use a framework previously developed at VHLab[6] research group, presented in [4]. After loading a set of virtual humans (including their animations), we load the simulation file (generated by the Simulation module), which contains the position of each agent at each frame/time, i.e. the motion of the agents is described by a set of 3D points over a time sequence. Listing 3.2 demonstrates an example of the simulation file format, used as input for the Visualization module.

The framework is also responsible for playing the animations accordingly to the situation. Based on the agent's moving speed, the framework is able to determine which animation should be played: walking, jogging or running, for example. For better rendering results, the framework applies a post process High Dynamic Range (HDR) rendering technique. This technique is implemented in the GPU enabling to be executed in real time. For the fixed pipeline rendering and animation, our prototype uses the Irrlicht Engine [7] and Cal3D [8], respectively. Next section contains the results obtained through simulations generated from our model.

---

[6] Virtual Humans Simulation Laboratory: http://www.inf.pucrs.br/vhlab

[7] `http://irrlicht.sourceforge.net`

[8] `http://gna.org/projects/cal3d`

```
<SIMULATION>
    <AGENTS>
        <AGENT id="0"></AGENT>
        <AGENT id="1"></AGENT>
    </AGENTS>
    <FRAMES>
        <FRAME>                                                        //Frame [0]
            <AGENTS>
                <AGENT id="0">
                    <POSITION>3.05 12.21 0.00</POSITION>               //Position of Agent [0] at Frame [0]
                </AGENT>
                <AGENT id="1">
                    <POSITION>37.41 12.21 0.00</POSITION>              //Position of Agent [1] at Frame [0]
                </AGENT>
            </AGENTS>
        </FRAME>
        <FRAME>                                                        //Frame [1]
            <AGENTS>
                <AGENT id="0">
                    <POSITION>3.05 12.21 0.00</POSITION>               //Position of Agent [0] at Frame [1]
                </AGENT>
                <AGENT id="1">
                    <POSITION>37.41 12.21 0.00</POSITION>              //Position of Agent [1] at Frame [1]
                </AGENT>
            </AGENTS>
        </FRAME>
    <FRAMES>
</SIMULATION>
```

Listing 3.2 – Simulation XML input data example for two distinct agents at two frames.

4. RESULTS

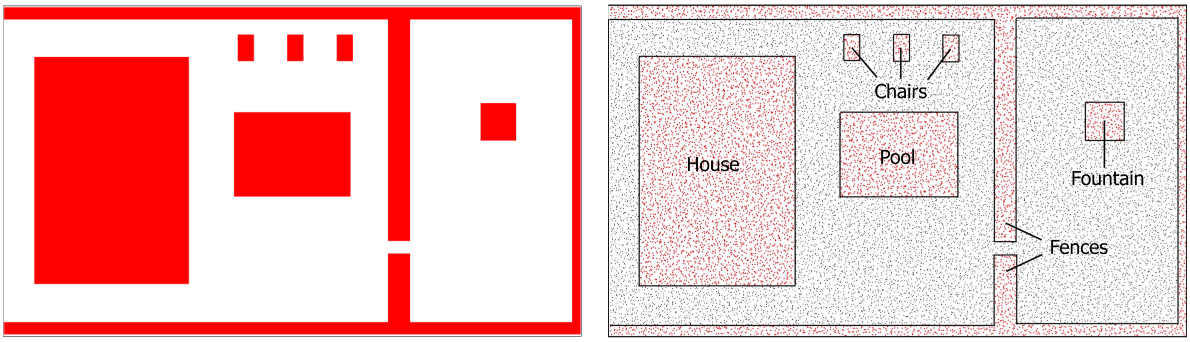
In this chapter will be presented information about the developed prototype for simulating the groups behaviors, using the proposed model. The experiments were performed using an Intel Xeon W369 equipped with NVidia Quadro 4000. We divided the results into sections for better organization. Section 4.1 explains details about the prototype developed to validate our model. In Section 4.2 we show results of particular simulation cases in order to analyse how our model deal with specific situations, while in Section 4.3 we reproduce a more complex environment. Finally, Section 4.4 contains results of simulations in more dense environments and evaluations of groups structures maintenance.

4.1 The Prototype

For analysis and validation of the proposed model, we developed a prototype in order to visually verify the produced simulations. The prototype was developed in C/C++ programming language and uses the graphic API OpenGL [47]. The environment is generated using the SVE Generator [31], as explained in Section 3.1, which creates as output, the 2D layout of the environment that contains goals, walkable and non-walkable regions and a graph, that will be used by the virtual agents to compute their paths. Next, we create the Population Class script file, similar to the one showed in Section 3.2, which can be generated automatically or manually. Since we have these data as input, we can start our simulations by computing the movement of the agents over time in the environment.

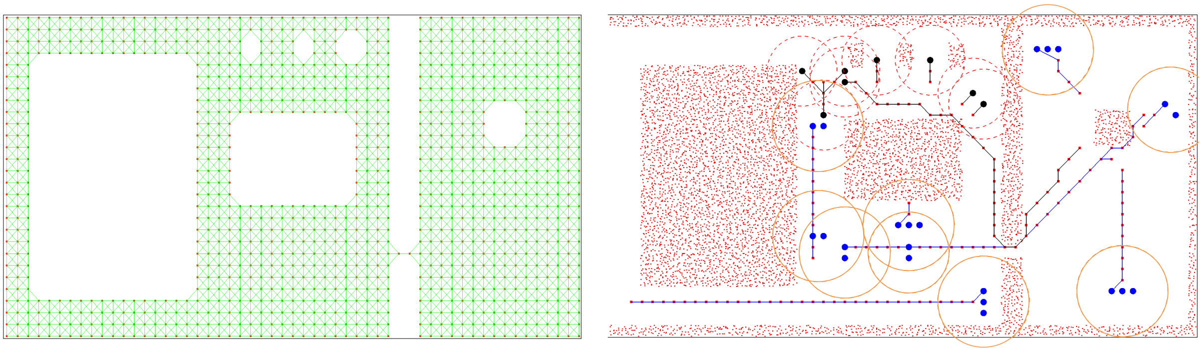
Figure 4.1 presents an example of creation of a certain environment, showing all steps of the pipeline. This figure represents a pool area in a backyard, where it is possible to observe modules of our model: in (a) is illustrated the bounding boxes that represents the semantic environment while in (b) and (c) are illustrated the markers and the graph, respectively, which are used as input to the simulation. Following, in (d) a view of our simulation environment presents the agents and their respective paths provided by the path planning algorithm. We can observe a population of 30 agents, 9 groups and 8 individuals manually generated as well as their goals, initial locations and groups behaviors. The agents in blue means agents that are part of some group and will move across the environment following the same path. On the other hand, agents in black represents individuals. Each group can be identified in Figure 4.1(d) by a circular area that represents the group space (explained in Section 3.2).

Using data from the population (*goals, paths* and *agents*), our algorithm is able to provide the agents motion across the environment. In such process, coherently groups formations are performed by our agents (*Side by Side, River-Like, V-Like*). Moreover, the groups are able to identify the presence of other groups and compute a new path or new formation when



(a) Bounding boxes of all instances of tangible objects that are present in the environment.

(b) Markers, in black, representing the walkable space. The dots in red inside the bounding boxes means regions where the motion in not allowed.



(c) Graph used for path planning algorithm considering the regions available for agents motion.

(d) Agents and their computed paths in the simulation module.

Figure 4.1 – Results of the main phases of our model.

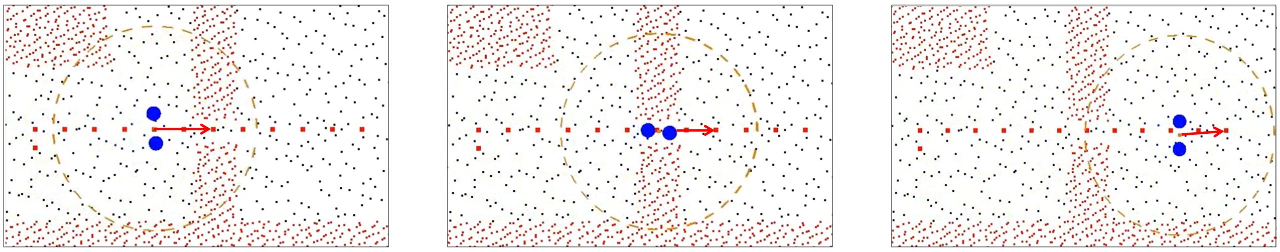
needed. Figure 4.2 illustrates the visualization of simulations in a 3D virtual environment, which represents a house containing several objects distributed through the spaces. It is possible to observe the virtual agents performing coherent group behaviors provided by our model (4.2(a)). Figure 4.2(b) demonstrates the same environment but raining, where the virtual humans are running toward an awning, which, in this case, is an interest point.

4.2 Simulation Cases

We created three simulation cases based on the recent work of Karamouzas and Overmars [25] (detailed in Section 2.2) , which are described in the following sections.

(a) (b)

Figure 4.2 – Snapshots of the 3D framework [4] visualizing an example of simulation generated by our model.



(a) (b) (c)

Figure 4.3 – Two agents applying a *Side by Side* formation before (a) and after (c) the door, while the *River-Like* is adopted to pass through the door (b).

4.2.1 Doorway

We simulated a group of two agents passing through a narrow door. Figure 4.3, shows a group of two members in formation *Side by Side*, that walk through a corridor until they face a door with small diameter (Figure 4.3(a)). Therefore, the algorithm - explained in Section 3.3.2, detects that there are no space for the group to pass with the current formation and changes to the *River-Like* formation (Figure 4.3(b)). After passing through the door, the *Side by Side* formation is then restored (Figure 4.3(c)). The door and the corridor, in Figure 4.3, are represented by non-walkable markers, in red.

4.2.2 Overtake

Figure 4.4 demonstrates the simulation of two groups moving to the same direction (Figure 4.4(a)), but one group is moving faster (in pink) than the other (in blue), resulting in an overtake (Figure 4.4(b)). This situation could represent a group of teenagers walking on the sidewalk, when a couple of older people is walking slowly in front of them. So, the teenagers tend to overtake the couple using the free space on the sides, returning to walk

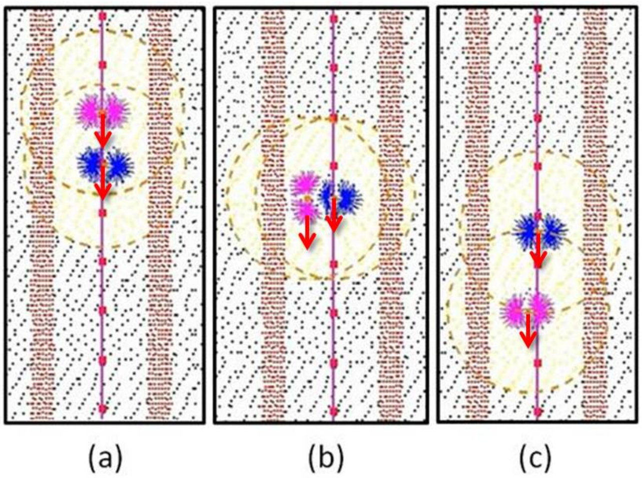


Figure 4.4 – A group overtaking a slower group. The group in pink moves faster than the group in blue (a), so an overtake occurs (b). After, the group restore the formation *Side by Side* (c).

side by side then (Figure 4.4(c)).

4.2.3 Narrow Corridor

Figure 4.5 shows the result of a simulation which a group walk through a narrow corridor. In this scenario, the group detect that there will be no space for the current formation - using the method explained in Section 3.3.2, and adapts its shape to the *V-Like* formation in order to fit in the corridor. After crossing the corridor, the *Side by Side* formation is then restored.

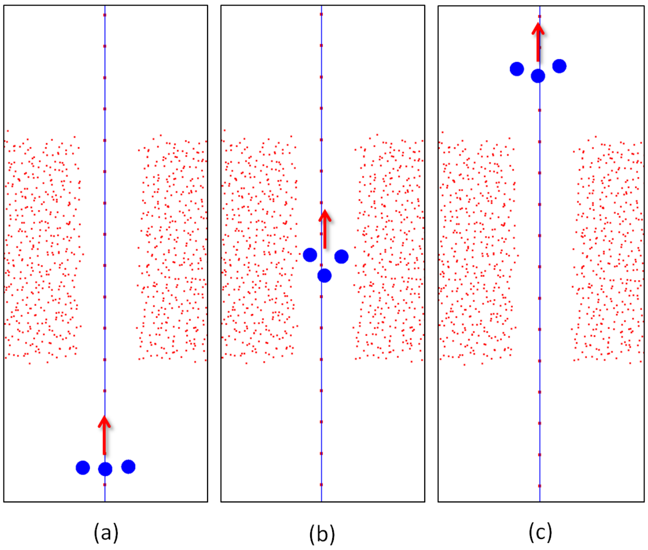


Figure 4.5 – Group crossing a narrow corridor. In (a), the group is about to enter the corridor. The group then deform its shape in order to walk through the corridor (b) and after, return to a more convenient formation (c).

4.3 Complex Environment

In order to test our model in more complex situations, we created a scenario containing several obstacles distributed throughout the environment. This scenario represents a backyard of a house and the main idea here is to simulate a party. In this party, people begins to arrive looking for fun, and after certain time, everybody starts gradually to leave the party. For this, we specified an arbitrary global seed and that the density of people should be LOW, having its peak in the half of the simulation total time. Besides, the population should contains 30% of individuals, 40% of groups of two members and 30% of three members (since more than 70% of the people walk in groups [35]). We distributed the interest resource "fun" through the whole space, so the agents could go wherever they desire in order to get the resource. The definition file for the PC used in this simulation is exhibited in Listing 4.1.

```
POPULATION CLASS
seed=123                              //Global seed
simulation_total_time=10000           //Simulation total time, in frames
higher_density=LOW                    //The higher density the simulation should have
peak_time=5000                        //The time that the peak of density should occur, in frames

Groups_distribution:
one_agent=0.30                        //Percentage of individuals
two_agents=0.40                       //Percentage of groups of two members
three_agents=0.30                     //Percentage of groups of three members

Interest_resources:
fun=1.0                               //Percentage of agents that seek for the "fun" resource

spawn_points=360,395                  //Node Ids of spawning locations
kill_points=3,136,525                 //Node Ids of killing locations
```

Listing 4.1 – Population Class definitions used for the simulation in the elaborated environment.

The simulation begins with the environment empty, when gradually the agents starts to appear in the defined spawn points. The groups are added to the environment at a rate $RtS = 1\ group/s$, until the density reaches its peak time or the higher density limit, which is defined as LOW (less than $1\ ag/m^2$)[1]. During the process, groups perform formation structures while seeking its goal and create conversational groups when arrive. The group (or individual) stays in the goal for a certain time[2] until another point of interest is randomly chosen for the group, that starts to pursue its new trajectorie. After reached the peak of density, RtS is set to 0 and the agents are gradually sended to the defined kill points, at a rate $RtK = 1\ group/s$, where they will be removed from the simulation. Finally, when there is no agent left or the simulation total time is achieved, the simulation finishes.

Figure 4.6 shows a screenshot of the simulation generated, where the environment structure and agents can be visualized. Obstacles are represented by non-walkable markers

---

[1] To achieve higher densities, RtS can be increased.

[2] The minimum and maximum amount of time in which groups stays in the goals are predefined and it is randomly attributed to each group. For this simulation, the minimum and maximum values are 30s and 100s, respectively.

in red, while free spaces are in blank and contains walkable markers. The figure also exhibits a grid for better notion of space and density, and each square of the grid represents $1\ m^2$. Spawn points are indicated by yellow stars and kill points are indicated by red X's. All agents are represented by circles, in order to simplify the visualization [3], and each color denotes the agents' status, as described below:

- Black: Individual moving toward its goal;
- Green: Group or individual arrived on its goal, waiting for a new goal attribution;
- Red: Group or individual moving toward a kill point;
- Other colors: Group moving toward its goal (members of the same group have the same color and are probably close to each other).

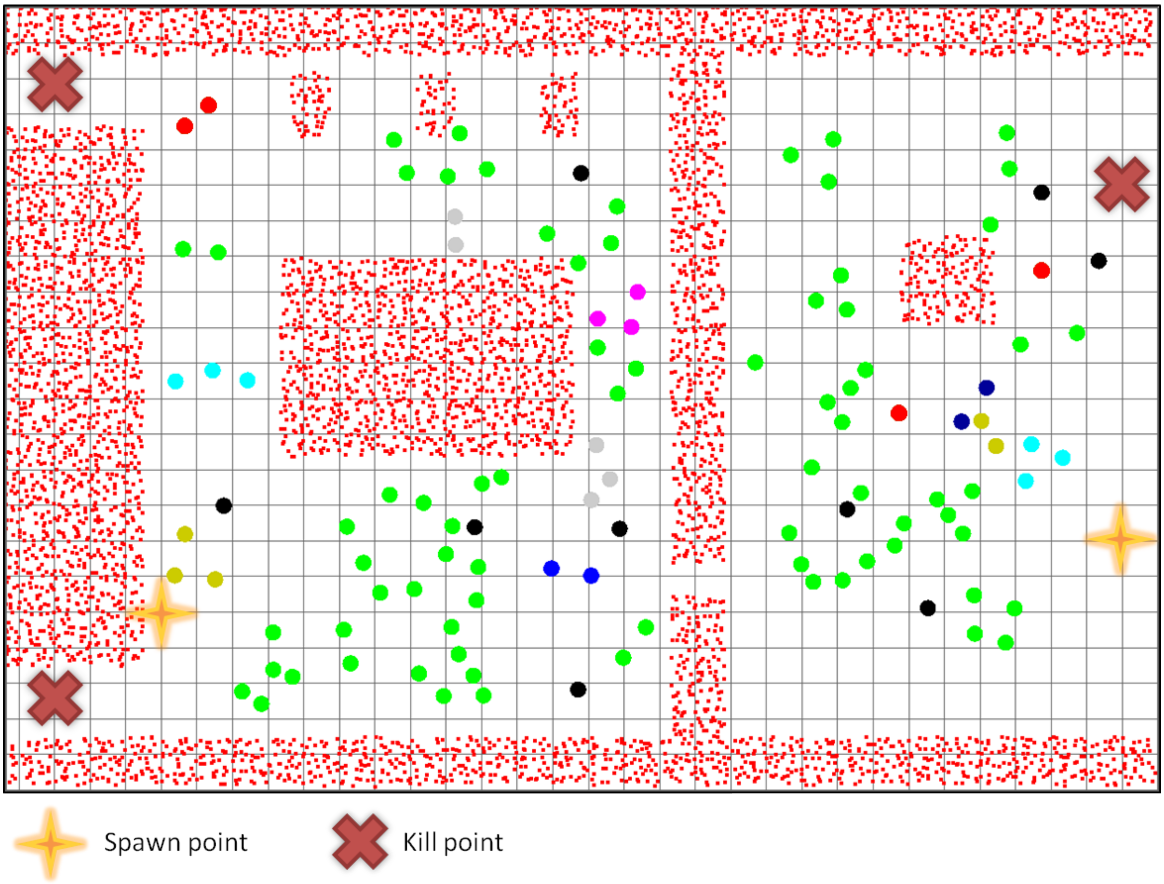


Figure 4.6 – Simulation screenshot of the elaborated environment. Obstacles are represented by red dots, while spawn and kill points are indicated by yellow stars and red X's, respectively. Agents are represented by circles and each color indicates the agents' status.

---

[3] Agents still having their *personal space*, calculated by the *Convex Hull* algorithm.

Figure 4.7 presents the density of people measured from the beginning until the end of the simulation. As the time passes, the average density starts to rise gradually as the agents are being added to the simulation. The density reach its peak at the half of the simulation total time (about 5000 frames) and its value is close to $0.5\ ag/m^2$. Next, the density begins to decrease as the agents are being removed from the simulation, until no agents left or the simulation total time is achieved.

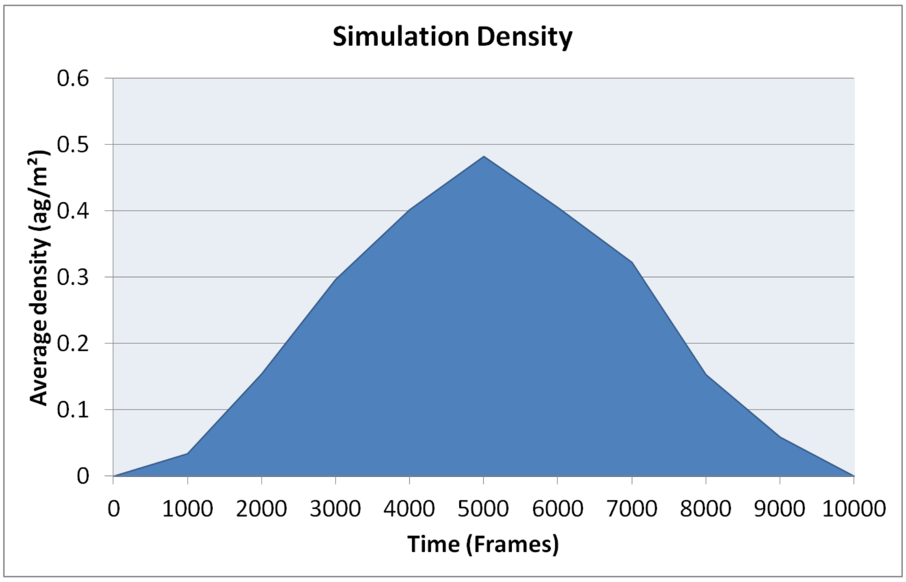


Figure 4.7 – Average density obtained from the beggining until the of the simulation.

4.4 Group Structure Evaluation

We created three different simulation scenarios in order to evaluate the groups structures, according to the density of people in the environment. The goal of these simulations is to obtain information about the maintenance of the group structure if the density of people begins to raise, such as the formations that are most performed in each density, the groups linkage and if the speed could influence the structure maintenance. We generated five simulations for each scenario, using PCs containing 30% of individuals, 40% of groups of two members and 30% of three members. In the simulation visualization, pedestrians that make part of some group are represented by the same group color and will probably be close to other group members. In the other hand, individuals are represented by black circles. Next sections will explain details about each simulation scenario.

4.4.1 Same Flow

In this scenario, we simulated pedestrians walking in the same flow, while the density starts to increase through time. Figure 4.8 illustrates the simulation scenario, where all the pedestrians are moving toward the direction indicated by the arrows in red until they reach the goal for being removed from the simulation. Also, it is possible to identify some formations during the trajectory. Some of them are marked by squares and arrows in orange - that indicates the moving direction - in Figure 4.8.

During the simulation, we measured the percentage of groups that were performing each possible formation - or no formation, while the density of people raise through time. The density of people are related to the rate that agents were added to the simulation, i.e. the higher the agents addition rate, the more dense the environment becomes. The simulation finishes when the density of people reach 4 agents per square meter.

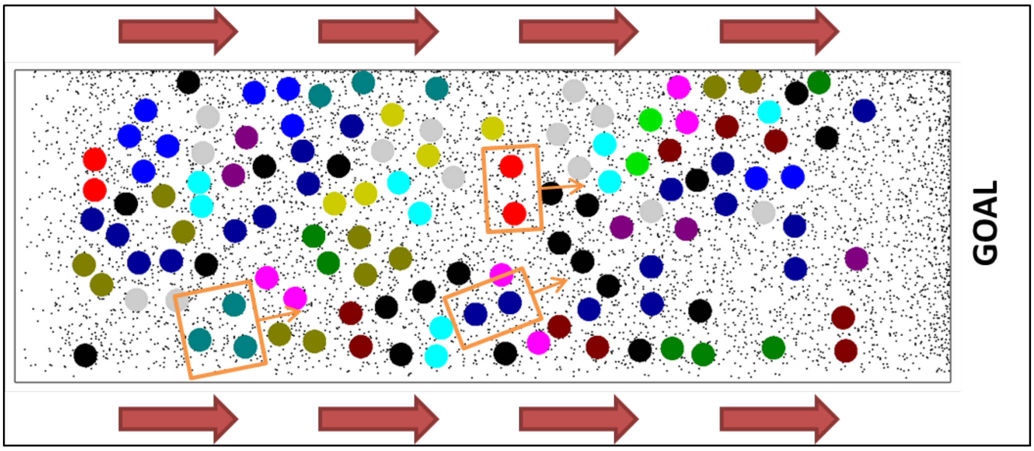


Figure 4.8 – Pedestrians moving in the same flow, indicated by arrows in red, toward the goal in the right. Some formations can be seen during the trajectory, showed by squares and arrows in orange, that indicates the group moving direction.

Figure 4.9 presents the measured group structure percentage for each formation in relation to the density. Each point of the chart represents the average percentage of groups performing that particular formation with the standard error, based on the 5 simulations. We noticed that in the begining of the simulation (low density), most groups performs the *Side by Side* formation, since this is the most convenient formation [35] and they have enough space to perform it. As the density starts to raise, the *Side by Side* formations begins to diminish and the *V-Like* and *River-Like* formations appears more often. After $1\ ag/m^2$ of density, the *River-Like* formation is more likely to appear than the others, as well as groups with no formation at all.

Furthermore, we measured the group linkage in relation to the density of people, i.e. if the group members stay close to each other (linked) as the density raises. This information is interesting to evaluate the distribution of groups inside the crowd. For instance, if the global measure of group linkage is low, it means that most of the group structures are disrupted and the individuals are very diluted in the crowd. However, if the group linkage is high, it indicates that most of the groups structures are maintained and the groups could be identified easier

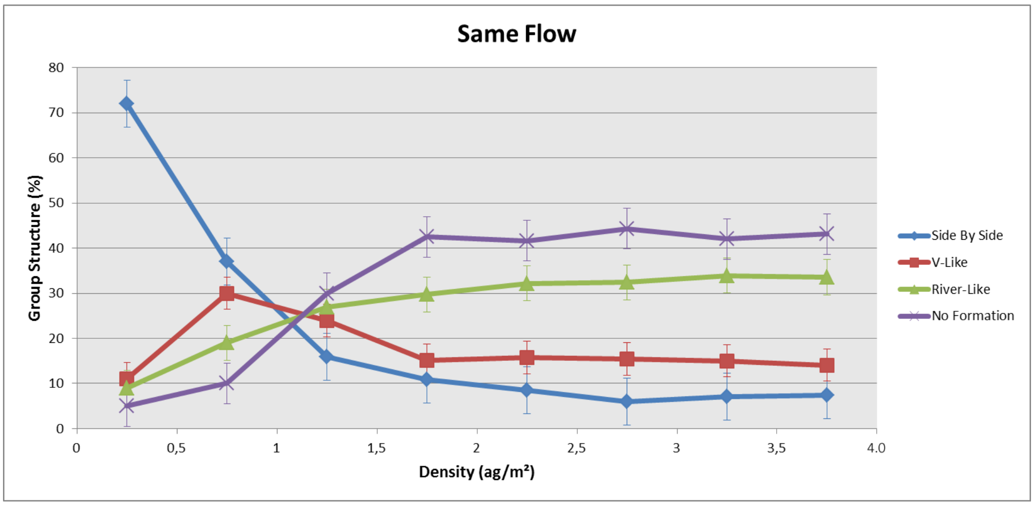


Figure 4.9 – Group structure percentage for each formation in relation to the density of people for the Same Flow scenario.

when visualizing the crowd. We consider that a group is linked when its members keep certain distance treshold T_{Lk} from each other, as explained in Section 3.2.2.

Figure 4.10 shows the average percentage of groups linkage in relation to the density of people. The measure were made considering the same simulations described before, using $T_{Lk} = 1.8\ meter$[4]. The chart indicates that groups linkage is high when the density is low. After reaching $1\ agent/m^2$ of density, groups linkage starts to slightly decrease, but still above 90% for higher densities.

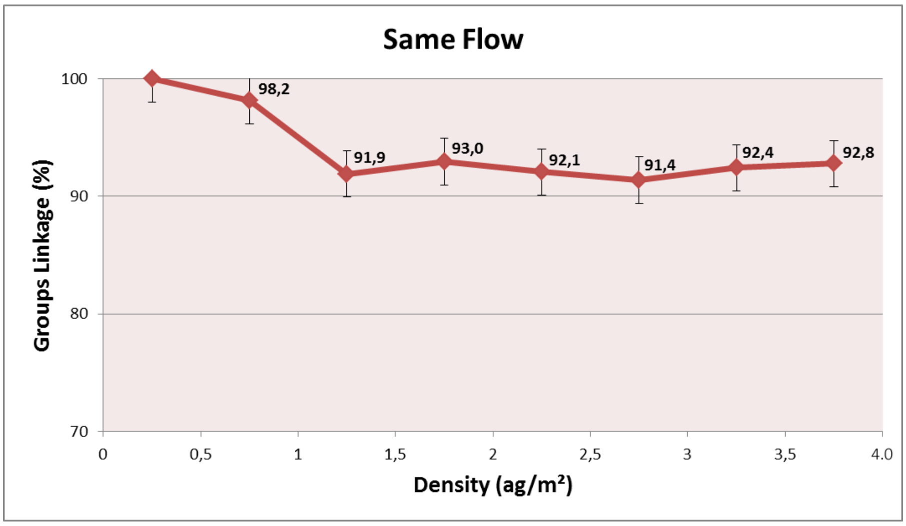


Figure 4.10 – Groups linkage percentage in relation to the density of people for the Same Flow scenario.

---

[4] Usually, the group members keep a distance of about 1.2 meter between each other in our tests, so the chosen treshold value represents 50% above the regular distance.

4.4.2 Cross Flow

In this scenario, we considered the same parameters used for the previous simulations (Section 4.4.1), however, in this case, the pedestrians will be added in both sides of the corridor, and will have goals disposed in the opposite side in which the agent has started. This way, the pedestrians will have to cross each other during the simulation in order to achieve its objective. Figure 4.11 presents the groups structure average percentage in relation to the density. In this case, we noticed that the *Side by Side* formation appears less often if compared to the Same Flow scenario (Figure 4.9), and also, the *River-Like* and groups with no formation curves rises earlier (and higher for groups without formations) in Figure 4.11. This happens because of the interference caused by the crossing between other groups and individuals, restricting the space for the group to perform more comfortable formations.

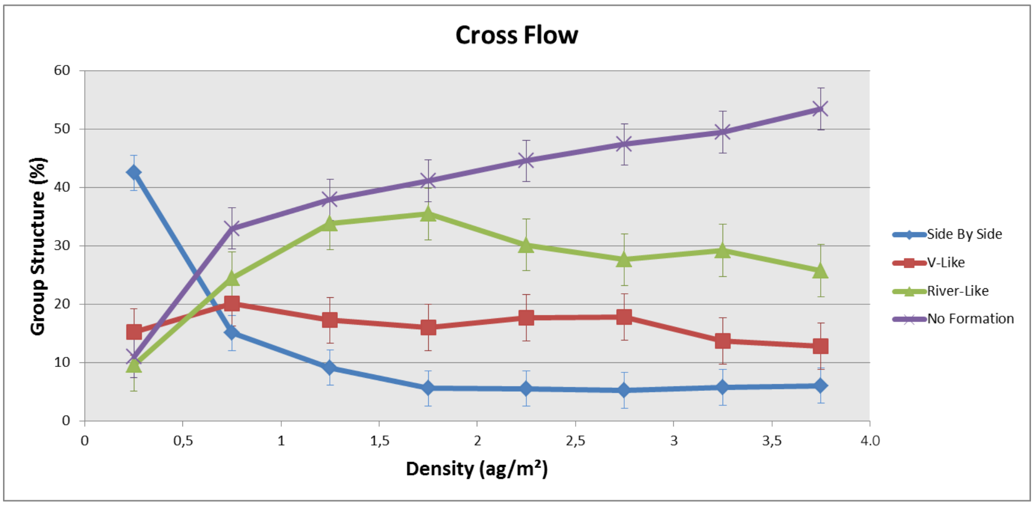


Figure 4.11 – Group structure percentage for each formation in relation to the density of people for the Cross Flow scenario.

As described in Section 4.4.1, we also measured the group linkage for the Cross Flow scenario. Figure 4.12 demonstrates the groups linkage average percentage over density of people. It is possible to notice that most groups keep linked in low and medium density situations, but the link starts to disrupt for higher densities, descreasing up to 80% - 10% less if compared to the Same Flow scenario (Figure 4.10).

4.4.3 Circular Flow

In this simulation scenario, we disposed the goals forming a circle shape, where the agents keep moving from one goal to another, creating a circular trajectory. Gradually, individuals and groups are added to the simulation, always walking from one goal to the next one located in clockwise, until we have 100 agents in total. Next, we start to increase

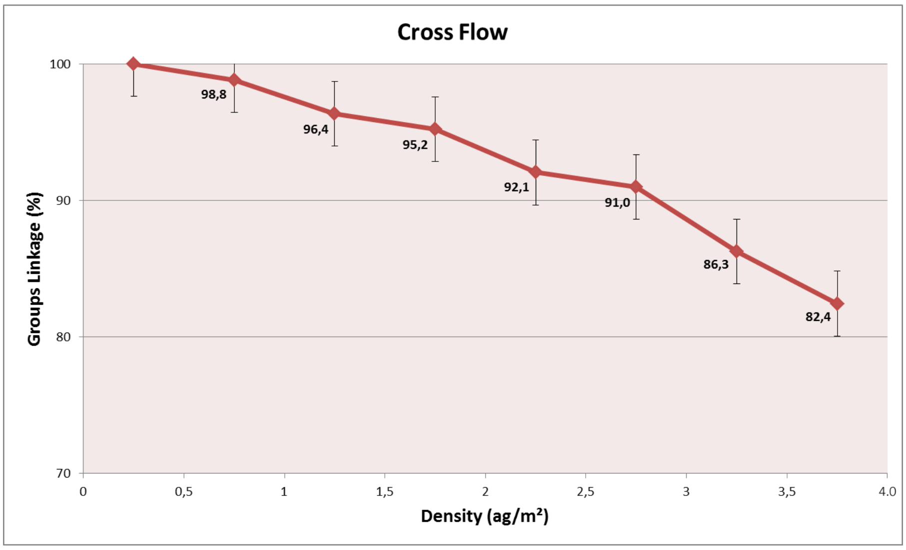


Figure 4.12 – Groups linkage percentage in relation to the density of people for the Cross Flow scenario.

gradually the speed of all agents, making they walk faster and faster. The simulation finishes when it reaches the desired average speed of $4\ m/s$. Figure 4.13 demonstrates the simulation scenario, where the agents are moving clockwise while trying to perform the most suitable formation. The simulations achieved an average density of people of approximately $0.58\ agents/m^2$. The objective of this simulation case is to observe if the agents' speed could influence the group linkage in low density situations.

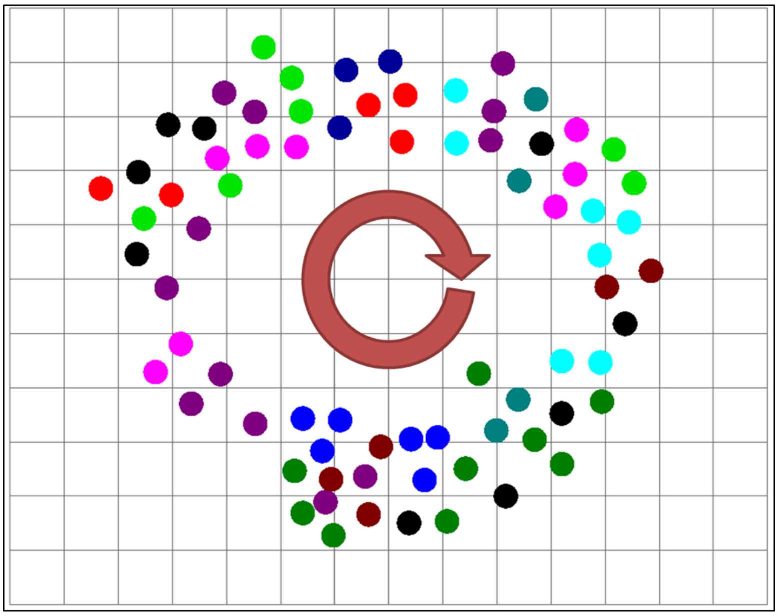


Figure 4.13 – Circular Flow simulation scenario, where all individuals and groups move clockwise.

Figure 4.14 shows the groups linkage percentage average in relation to the average desired speed of all individuals and groups. The simulation sample presented an average error of 2.3%. According to the chart presented in Figure 4.14, the groups linkage stay elevated for regular speeds, but starts to decrease as the average speed increase until nearing 70%. This means that as the speed increase, the more difficult is for the groups members stick together, since other agents could interfere in other groups structure while trying to perform their own.

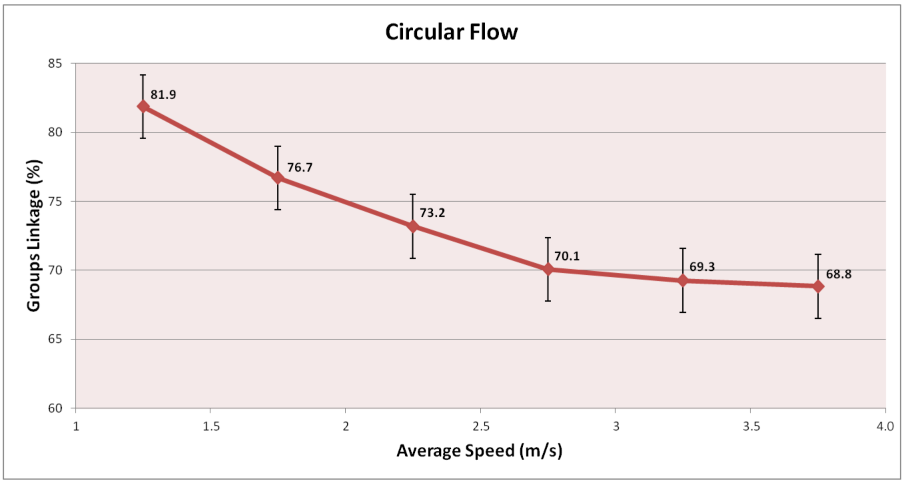


Figure 4.14 – Groups linkage percentage in relation to the desired average speed for the Circular Flow scenario.

5. FINAL CONSIDERATIONS

We presented a model able to provide procedurally coherent group behaviors in a semantic 3D environment. Only the environment is manually created, while all agents and groups behaviors can be automatically generated. Each group is able to perform different behaviors (*Side by Side*, *V-Like*, *River-Like*) according to the situation, while avoiding collision with other groups and objects in the space. Obtained results can be easily visualized in real-time using the 3D framework.

The main contributions of this work are to generate simulations with few user intervention, that allow the animator to be focused on the big picture and in the first plan characters, and provide group behaviors considering characteristics of the environment. Besides that, our model can be applied in games in order to coherently populate several sorts of environments. Another contribution is that our model is able to reproduce virtual humans moving in groups in order to allow social interactions between the members, while most studies analysed the crowd as a collection of isolated individuals.

Results showed that our model can deal with many different situations: from simpler cases, like pedestrians passing through a door, to more complex cases, like simulating a party with several groups of people. We noticed that group structures have direct correlation with the density of people, confirming the assertions presented in the study of Moussaid *et al.* [35]. Our results showed that the *Side by Side* formation is more likely to appear when the density of people is low, once the group members have enough space to perform it. When the density starts to raise, the *V-Like* formation is performed more often, and for higher densities, *River-Like* formation and groups with no formation at all appears with high frequency.

Another finding is that when people are walking all in the same flow, the group structure do not suffer many interference, once the others pedestrians are heading in the same direction, without cross each other. However, when groups of pedestrians are moving in opposite directions, the group structure is compromised by the other pedestrians coming from the contrary flow, also making difficult for the members to stick together. We also noticed that the speed in which the crowd is moving could influence the group structure, and it gets more difficult for the group members to stick together as speed increases. For instance, in an emergency situation, members of a family could be separated easier if the people in the crowd are trying to run toward the exit due to a fire.

Currently, our model is only able to simulate formations for groups up to 3 agents. We intend in a future work to include formations that involves more than 3 agents, either while moving or arranged in conversational groups. We also want to reproduce behaviors based on agent's type, i.e. group behaviors can vary according to the type of agents that compose each group, e.g. a married couple with a kid have a different behavior then a group of teenagers or a group of older people.

Finally, we are able to create a model capable of simulating inter and intra group behaviors using semantic informations in the environment, that is also useful for generating visual results in our latest researches.

During the master program experience, the author of this dissertation was involved in several researches in computer graphics, resulting in five published papers (see Appendix A). The research provided the opportunity of visiting the University of South Brittany (UBS) facilities for three month period (December to February, 2012/2013).

REFERENCES

[1] ANTONINI, G., BIERLAIRE, M., AND WEBER, M. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological 40*, 8 (2006), 667 – 687.

[2] AVENI, A. F. The Not-So-Lonely Crowd: Friendship Groups in Collective Behavior. *Sociometry 40*, 1 (1977), 96–99.

[3] BON, G. *Psychologie des foules*. 9. ed. Paris, França: Édition Félix Alcan, 1905. 192 p., 1895.

[4] BRAUN, H. Crowdvis: A framework for real time crowd visualization. Master's thesis, Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS, 2012.

[5] CASSOL, V. J., MARSON, F. P., VENDRAMINI, M., PARAVISI, M., BICHO, A. L., JUNG, C. R., AND MUSSE, S. R. Simulation of autonomous agents using terrain reasoning. In *Proc. the Twelfth IASTED International Conference on Computer Graphics and Imaging (CGIM 2011)* (Innsbruck, Austria, 2011), vol. 1, IASTED/ACTA Press, pp. 101–108.

[6] CHAMPAGNE, J., AND TANG, W. Real-time simulation of crowds using voronoi diagrams. L. M. Lever and M. McDerby, Eds., Eurographics Association (Canterbury, United Kingdom, 2005), pp. 195–201.

[7] CHAPPLE, E. D. Quantitative analysis of the interaction of individuals. *Proceedings of the National Academy of Sciences of the United States of America 25*, 2 (1939), pp. 58–67.

[8] COLEMAN, J. S., AND JAMES, J. The equilibrium size distribution of freely-forming groups. *Sociometry 24*, 1 (1961), 36–45.

[9] DAPPER, F., PRESTES, E., AND NEDEL, L. P. Generating steering behaviors for virtual humanoids using bvp control. In *Proc. of Computer Graphics International* (RJ, Brazil, 2007), pp. 105–114.

[10] DE LIMA BICHO, A. *From Plants to Crowd Dynamics: A bio-inspired model(in portuguese)*. PhD thesis, State University of Campinas, Campinas, Brazil, 2009.

[11] DE LIMA BICHO, A., RODRIGUES, R. A., MUSSE, S. R., JUNG, C. R., PARAVISI, M., AND MAGALHÃES, L. P. Simulating crowds based on a space colonization algorithm. *Computers & Graphics 36*, 2 (2012), 70 – 79. Virtual Reality in Brazil 2011.

[12] FRUIN, J. J. *Pedestrian and planning design*. Metropolitan Association of Urban Designers and Environmental Planners, Elevator World Inc., New York, NY, USA, 1971.

[13] GUTIERREZ, M., VEXO, F., AND THALMANN, D. Semantics-based representation of virtual environments. *International Journal of Computer Applications in Technology 23*, 2-4 (2005), 229–38. Virtual Reality Lab., Ecole Polytech. Fed. de Lausanne, Switzerland.

[14] HALL, E. T. *The hidden dimension / Edward T. Hall*, [1st ed.] ed. Doubleday, Garden City, N.Y. :, 1966.

[15] HARE, A. *Handbook of small group research*. Free Press of Glencoe, 1962.

[16] HART, P. E., NILSSON, N. J., AND RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. *SIGART Bull.* (1972), 28–29.

[17] HELBING, D. Pedestrian dynamics and trail formation. In *Traffic and Granular Flow '97* (Singapore, 1997), M. Schreckenberg and D. E. Wolf, Eds., Springer-Verlag, pp. 21–36.

[18] HELBING, D., AND MOLNÁR, P. Social force model for pedestrian dynamics. *Phys. Rev. E 51*, 5 (May 1995), 4282–4286.

[19] HENDERSON, L. F. The statistics of crowd fluids. *Nature 229*, 5284 (1971), 381–383.

[20] HENDERSON, L. F. On the fluid mechanic of human crowd motions. *Transportation Research 8*, 6 (1974), 509–515.

[21] JAMES, J. The Distribution of Free-Forming Small Group Size. *American Sociological Review 18*, 5 (1953), 569–570.

[22] JOHANSSON, A., HELBING, D., AND SHUKLA, P. K. Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems 10* (2007), 271–288.

[23] KALLMANN, M., AND MATARIC, M. Motion planning using dynamic roadmaps. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans, Louisiana, 2004), pp. 4399–4404.

[24] KAMPHUIS, A., AND OVERMARS, M. H. Finding paths for coherent groups using clearance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 19–28.

[25] KARAMOUZAS, I., AND OVERMARS, M. Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (2012), 394–406.

[26] KENDON, A. Spacing and orientation in co-present interaction. In *Proceedings of the Second international conference on Development of Multimodal Interfaces: active Listening and Synchrony* (Berlin, Heidelberg, 2010), COST'09, Springer-Verlag, pp. 1–15.

[27] KESSING, J., TUTENEL, T., AND BIDARRA, R. Services in game worlds: A semantic approach to improve object interaction. In *Proceedings of the 8th International Conference on Entertainment Computing* (Berlin, Heidelberg, 2009), ICEC '09, Springer-Verlag, pp. 276–281.

[28] LAVALLE, S. Rapidly-exploring random trees: A new tool for path planning. Tech. Rep. TR98-11, Dep. of Computer Science, Iowa State University, October 1998.

[29] LEE, K. H., CHOI, M. G., HONG, Q., AND LEE, J. Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '07)* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 109–118.

[30] LIEN, J.-M., RODRÍGUEZ, S., MALRIC, J.-P., AND AMATO, N. M. Shepherding behaviors with multiple shepherds. In *Proceedings of the IEEE Inter. Conf. on Robotics and Automation* (April 2005), pp. 3402–3407.

[31] MARSON, F. P. *Geração Procedural de Ambientes Virtuais Semânticos*. PhD thesis, Pontifícia Universidade Católica do Rio Grande do Sul, 2012.

[32] METOYER, R. A., AND HODGINS, J. K. Reactive pedestrian path following from examples. *The Visual Computer 20*, 10 (2004), 635–649.

[33] MILLS, T. M. *The sociology of small groups / Theodore M. Mills*. Prentice-Hall, Englewood Cliffs, N.J., 1967.

[34] MOUSSAID, M., HELBING, D., GARNIER, S., JOHANSSON, A., COMBE, M., AND THERAULAZ, G. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B: Biological Sciences 276* (2009), 2755–2762.

[35] MOUSSAÏD, M., PEROZO, N., GARNIER, S., HELBING, D., AND THERAULAZ, G. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE 5*, 4 (04 2010), e10047.

[36] MUSSE, S. R., AND THALMANN, D. A model of human crowd behavior: Group interrelationship and collision detection analysis. In *Proc. Workshop of Computer Animation and Simulation of Eurographics'97* (1997), pp. 39–51.

[37] O'ROURKE, J. *Computational Geometry in C (Second Edition)*. Cambridge University Press, Cambridge, UK, 1998.

[38] OTTO, K. A. Semantic virtual environments. In *Special interest tracks and posters of the 14th international conference on World Wide Web* (New York, NY, USA, 2005), ACM, pp. 1036–1037.

[39] RODRÍGUEZ, S., LIEN, J.-M., AND AMATO, N. M. A framework for planning motion in environments with moving obstacles. In *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems* (November 2007), pp. 3309–3314.

[40] RODRIGUES, R. A., DE LIMA BICHO, A., PARAVISI, M., JUNG, C. R., MAGALHAES, L. P., AND MUSSE, S. R. An interactive model for steering behaviors of groups of characters. *Appl. Artif. Intell. 24* (July 2010), 594–616.

[41] RUNIONS, A., FUHRER, M., LANE, B., FEDERL, P., ROLLAND-LAGAN, A.-G., AND PRUSINKIEWICZ, P. Modeling and visualization of leaf venation patterns. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM, pp. 702–711.

[42] SACHS, T. Polarity and the induction of organized vascular tissues. *Annals of Botany 33*, 2 (1969), 263–275.

[43] SIGHELE, S. *A Multidão Criminosa - Ensaio de Psicologia Coletiva*. eBooksBrasil, 1954.

[44] STEPHAN, F. F., AND MISHLER, E. G. The distribution of participation in small groups: An exponential approximation. *American Sociological Review 17*, 5 (1952), pp. 598–608.

[45] STILL, G. K. *Crowd Dynamics*. PhD thesis, University of Warwick, Coventry, UK, August 2000.

[46] THALMANN, D., AND MUSSE, S. R. *Crowd Simulation*. Springer-Verlag London Ltd, 2007.

[47] WOO, M., NEIDER, J., DAVIS, T., AND SHREINER, D. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

Appendix A. Publications

BRAUN, H., CASSOL, V., HOCEVAR, R., MARSON, F., AND MUSSE, S.; *CrowdVis: A Framework for Real Time Crowd Visualization*. To appear in ACM 28th Symposium On Applied Computing (2013).

FLACH, L. ; Dill, V. ; HOCEVAR, R. ; PINHO, M. S. ; MUSSE, S.; *Evaluation of the Uncanny Valley in CG Characters.* In: Brazilian Symposium on Computer Games and Digital Entertainment, 2012, Brasília. Proceedings of SBGames, 2012.

HOCEVAR, R., MARSON, F., BRAUN, H., CASSOL, V., BIDARRA, R., AND MUSSE, S. R.; *From their environment to their behavior: A procedural approach to model groups of virtual agents*. In IVA '12: Proceedings of the 12th International Conference on Intelligent Virtual Agents (2012).

DILL, V., FLACH, L., HOCEVAR, R., LYKAWKA, C., PINHO, M., AND MUSSE, S.; *Evaluation of the Uncanny Valley in CG Characters*. Lecture Notes in Computer Science. 12ed.: Springer Berlin Heidelberg, 2012, v. 7502, p. 511-513.

BRAUN, H., HOCEVAR, R., QUEIROZ, R. B., COHEN, M., MOREIRA, J. L., JACQUES JÚNIOR, J. C., BRAUN, A., MUSSE, S. R., AND SAMADANI, R.; *VHCVE: A collaborative virtual environment including facial animation and computer vision*. In Proceedings of the 2009 VIII Brazilian Symposium on Games and Digital Entertainment (Washington, DC, USA, 2009), SBGAMES '09, IEEE Computer Society, pp. 207-213.