Universidade Federal do Ceará

Departamento de Computação

Mestrado e Doutorado em Ciência da Computação

CRAb - Computação Gráfica, Realidade Virtual e Animação

Teófilo Bezerra Dutra

Gradient-based Steering for Vision-based Crowd Simulation Algorithms

Fortaleza

2015

Teófilo Bezerra Dutra

Gradient-based Steering for Vision-based Crowd Simulation Algorithms

Tese apresentada ao Departamento de Computação da Universidade Federal do Ceará como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Joaquim Bento Cavalcante Neto

Coorientadores: Prof. Dr. Creto Augusto Vidal
Dr. Julien Pettré

Fortaleza

2015

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

D978g Dutra, Teófilo Bezerra.
Gradient-based steering for vision-based crowd simulation algorithms / Teófilo Bezerra Dutra. – 2015.
112 p. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Pós-Graduação em Ciência da Computação, Fortaleza, 2015.
Orientação: Prof. Dr. Joaquim Bento Cavalcante Neto.
Coorientação: Prof. Dr. Creto Augusto Vidal
Coorientação: Dr. Julien Pettré

1. Simulação por computador. 2. Realidade virtual. 3. Computação – métodos de simulação. I. Título.

CDD 005

Teófilo Bezerra Dutra

Gradient-based steering for vision-based crowd simulation algorithms.

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como requisito para a obtenção do grau de Doutor em Ciência da Computação.

Aprovada em 16 de junho de 2015

BANCA EXAMINADORA

Joaquim Bento Cavalcante

Prof. Dr. Joaquim Bento Cavalcante Neto
Presidente
Universidade Federal do Ceará – UFC

Creto Augusto

Prof. Dr. Creto Augusto Vidal
Coorientador
Universidade Federal do Ceará – UFC

Prof. Dr. Julien Pettré
Institut National de Recherche en Informatique et en Automatiqu - INRIA

Profa. Dra. Soraia Raupp Musse
Pontifícia Universidade Católica do Rio Grande do Sul - PUC

Emanuele Marques dos Santos

Profa. Dra. Emanuele Marques dos Santos
Universidade Federal do Ceará – UFC

Fortaleza, 16 de junho de 2015

Acknowledgements

Aos meus pais, Teófilo (Francisco) e Inês, por todo o incentivo que me foi dado a educação, o apoio em todos os momentos da minha vida e por terem me proporcionado sempre tudo o que fosse necessário para o meu bem-estar. À minha irmã, Natália, por ser sempre uma ótima irmã e amiga. À minha amada esposa, Akemi, por todo seu amor, carinho, companheirismo e cumplicidade; e, acima de tudo, por me compreender e me apoiar durante todos os momentos difíceis enfrentados durante a minha jornada acadêmica.

Aos meus amigos de longa data, Thiarley, Sofia, Dácio, Vanessa, Bosco, Janaína, Felipe, Marcela, Leo, Thyci, Artur, Clarissa, Tales, Carcarah, Paulo, Tica, Rafael Carmo, Gilvan, só para citar alguns. Aos amigos do grupo de Computação Gráfica, Realidade Virtual e Animação (CRAb), sempre dispostos a ajudar, Markos, Martha, Roberto, Rubens, Yuri, Lílian, Laise, Arnaldo, Bustamante, Lenz, Daniel Cavalcante, Daniel Siqueira, Rafael Ivo, Rafael Vieira, Suzana, Daniel Teixeira, Danilo e todos outros que não citei porque é muita gente, mas saibam que vocês são todos são ótimos e foram essenciais para essa conquista.

Aos meus amigos em Rennes, pessoal dos grupos MimeTIC e Hybrid, mas não somente, Steve, Ferran, Ana, Fabien, Fanny, Bruneau, Helen, Anne-Hélène, Ana Lúcia, David, Tristan, Kevin, Panayiotis, Merwan, Da'rya, Orianne, Carl, Thomas, só para citar alguns, que tornaram meu ano em Rennes muito mais agradável. Aos amigos brasileiros em Rennes, Regina, Luciano, Lara e especialmente, Júlio, que teve um papel fundamental neste trabalho ao me ajudar bastante com minhas limitações matemáticas durante o desenvolvimento do mesmo.

Aos colaboradores do trabalho. Ao Ondřej, não somente pela inspiração que me foi dada pelo seu trabalho, mas também por sua disponibilidade em me ajudar no entendimento do mesmo. E ao Ricardo, por toda a ajuda (e não foi pouca), paciência e amizade; acredito que tudo teria sido bem mais difícil sem a sua ajuda.

Aos meus orientadores, prof. Joaquim Bento e prof. Creto Vidal, por terem acreditado no meu potencial e por todos os ensinamentos passados durante esses anos. Ao meu orientador, Dr. Julien Pettré, por também ter acreditado no meu potencial, por ter me recebido no grupo MimeTIC para o sanduíche e por todos os ensinamentos passados desde que iniciamos essa colaboração. O agradeço ainda por ter dedicado parte do seu tempo para vir a Fortaleza participar pessoalmente da banca examinadora da minha defesa. À profa. Soraia Musse por ter aceitado participar da banca examinadora e, com isso, ter engrandecido a discussão sobre o trabalho com toda a sua experiência de vários anos pesquisando na área de simulação de multidão. E ainda, à profa. Emanuele Santos também por ter aceitado participar da banca examinadora.

Ao Departamento de Computação e ao Programa Mestrado e Doutorado em Ciência da Computação (MDCC) da Universidade Federal do Ceará pela oportunidade.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

E a todos os outros que não foram citados, mas que foram importantes de alguma forma para o desenvolvimento deste trabalho.

"O cientista não estuda a natureza por sua utilidade;
ele o faz porque se deleita com ela, e esse deleite vem de sua beleza.
Se a natureza não fosse bela, não valeria a pena conhecê-la,
e se não valesse a pena conhecê-la, não haveria por que viver esta vida."
(Henri Poincaré)

Resumo

Alguns dos algoritmos mais recentes para simulação de multidão equipam agentes com um sistema visual sintético para auxiliá-los em sua locomoção. Eles oferecem perspectivas promissoras ao imitarem de forma mais realista a forma como os humanos navegam de acordo com o que eles percebem do seu ambiente. Nesta tese, é proposto um novo laço de percepção/ação para dirigir agentes ao longo de trajetórias livres de colisões que melhoram significativamente a qualidade dos simuladores de multidão baseados em visão. Em contraste com abordagens anteriores - que fazem agentes evitarem colisões de maneira puramente reativa - é sugerida a exploração de toda gama de adaptações possíveis e a retenção da que for ótima localmente. Para isto, é introduzida uma função de custo, baseada em variáveis de percepção, que estima a situação atual do agente considerando tanto os riscos de futuras colisões como o destino desejado. São então computadas as derivadas parciais dessa função com respeito a todas adaptações de movimento possíveis. O agente adapta seu movimento de forma a seguir o gradiente descendente. Esta tese possui assim duas principais contribuições: a definição de um esquema de controle de propósito geral para a orientação de agentes baseados em visão sintética; e a proposição de funções de custo para avaliar o perigo da situação atual. As melhorias obtidas com o modelo são demonstradas em diversos casos.

Palavras-chaves: Simulação de multidão. Visão sintética. Prevenção de colisão.

Abstract

Most recent crowd simulation algorithms equip agents with a synthetic vision component for steering. They offer promising perspectives by more realistically imitating the way humans navigate according to what they perceive of their environment. In this thesis, it is proposed a new perception/motion loop to steer agents along collision free trajectories that significantly improves the quality of vision-based crowd simulators. In contrast with previous solutions - which make agents avoid collisions in a purely reactive way - it is suggested exploring the full range of possible adaptations and to retain the locally optimal one. To this end, it is introduced a cost function, based on perceptual variables, which estimates an agent's situation considering both the risks of future collision and a desired destination. It is then computed the partial derivatives of that function with respect to all possible motion adaptations. The agent adapts its motion to follow the steepest gradient. This thesis has thus two main contributions: the definition of a general purpose control scheme for steering synthetic vision-based agents; and the proposition of cost functions for evaluating the dangerousness of the current situation. Improvements are demonstrated in several cases.

Keywords: Crowd simulation. Synthetic vision. Collision avoidance.

List of Figures

Contents

1 Introduction

A crowd can be defined as: "*a large number of people gathered together in a disorganized or unruly way*"[1]. The disorganization, in this case, is related to the lack of a previous organization. However, when people start interacting with each other in a crowd the emergence of self-organized patterns can be observed. A virtual crowd is composed of several moving entities, the so-called agents. Each of those agents needs to reach a goal inside a virtual environment while avoiding collision with each other and with other static and moving obstacles. The field of Crowd Simulation deals with the synthesis of such virtual crowds in order to reproduce the self-organized behavior observed in real crowds. That field is rapidly extending over various application fields, such as civil engineering, architectural design and the entertainment industry to populate games and movie scenes (Figure 1). All those fields demand high-quality and realistic simulations, where the notion of realism, however, can take different meanings. For movies, high-quality animation (visual and behavioral) are demanded and performance is not important since the results are played off-line. Whereas for games and other virtual reality applications, besides the animation's quality, the performance has an important role, since these kind of applications need to run in real-time. In civil engineering or architectural design, the visual quality of the animation and the application's performance are less important. In this case, the interest lies in modeling the individuals' behavior properly to mimic human behavior in emergence situations, for example.

Despite all the recent advances on the field of crowd simulation, there is still a long way to pursue regarding the understanding of the human's perception/motion so as to reproduce more realistic behaviors. It is know from the literature, for example, that the optic flow has an important role on locomotion (GIBSON, 1958; CUTTING et al., 1995). Reproducing human senses on a virtual environment is a challenging task. Since on computer graphics everything is purely visual, focusing on the human vision is very important to understand the role of this sense and how to reproduce it synthetically for going deeper on reproducing more believable human behavior.

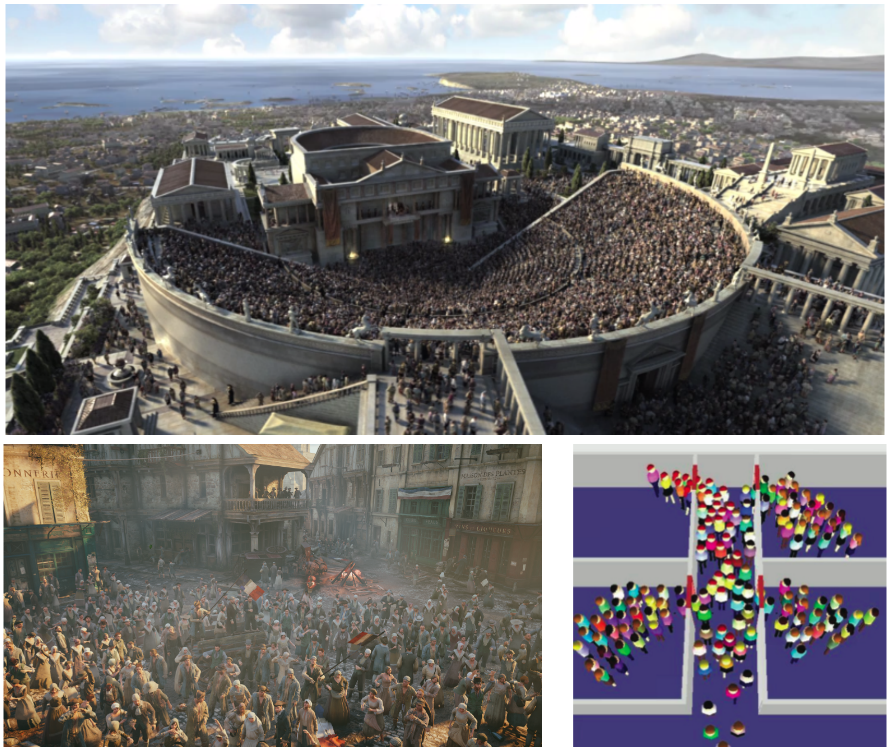
1.1 Contextualization

The Crowd Simulation field is a multidisciplinary one, involving several subfields which need to cooperate with one another so that the virtual crowds may be able to reproduce the behaviors observed in real crowds. One of the most important subfields, and consequently, one of those which receives more attention from the research community is the one which deals with how the agents plan their paths towards their goals, a task which is called *path planning*.

Path planning for crowd simulation is an area of extensive research, and the approaches used in this area can be split into two categories: local and global (Figure 2). Methods for local path planning can also be referred to as methods for collision avoidance, since they are intended to make the agents avoid collisions locally, at short range. Global path planners divide the environment into waypoints which help the agents to traverse complex environments. Usually, global and local path planners are used together, where waypoints provided by global path planners are used as goals for local path planners.

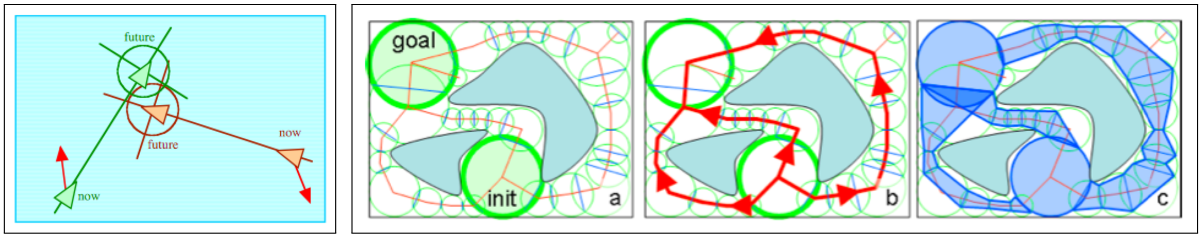
[1] http://www.oxforddictionaries.com/definition/english/crowd

Figure 1 – Examples of virtual crowds in a movie (top), in a game (bottom-left) and in an evacuation scenario (bottom-right).



Source: Hercules (PARAMOUNT PICTURES, 2014) (top), Assassin's Creed Unity (bottom-left) (UBISOFT, 2014) and Pelechano et al. (2007) (bottom-right).

Figure 2 – Examples of local (left) and global (right) path planning.



Source: Reynolds (1999) (left) and Pettré et al. (2005) (right).

Regarding local path planners, a recent class of agent-based algorithms, called velocity-based algorithms, allowed crowd simulators to make significant progresses in the last few years towards new levels of realism and robustness. These algorithms use the velocity of the agents and of the surrounding obstacles to predict the risks of future collision, and allow the agents to react to this risk with anticipation, as real humans do. A specific category of velocity-based algorithms pushed the notion of realism even further by equipping agents with a synthetic vision component. A number of motion variables are perceived by agents through a virtual retina. These, so-called perceptual variables, serve as input to a motion control loop that steers the agents through dynamic environments.

The use of synthetic vision for crowd simulation, once viewed with suspicion by the community given its high computational cost, opens a new range of possibilities. The results obtained by the first approach based on synthetic vision for simulating crowds, proposed by Ondřej et al. (2010), showed a good prospect for this new branch of crowd simulators.

In (ONDŘEJ et al., 2010), the agents react to the visually perceived collision threats (danger) by turning or by slowing down when collision becomes imminent. This simple perception action loop suffers from important drawbacks because of the nature of its proof of concept. In particular, that algorithm focuses on the most threatening obstacles only, decelerating and turning to avoid the most imminent collision threat, and disregarding the other obstacles. Moreover, it does not inspect the effects of maneuvers. Therefore, avoidance trajectories can actually lead to collision with some other nearby obstacles. For example, an agent that walks along a wall on its right side may collide with it when trying to escape a collision danger coming from its left.

1.2 Objectives

This work aims at further developing algorithms based on synthetic vision, motivated by several of their interesting properties. They are able to steer agents using a visual apparatus similar to what real humans do when they walk. They abstract moving and static obstacles with no distinction of nature. They consider the real shape of obstacles. They implicitly solve the question of combining and filtering several interactions when projecting them on the virtual retina. As a result, more natural trajectories are expected in comparison with other approaches that rely on information of a different nature. They actually demonstrate their ability to simulate the emergence of self-organized structures of agents under specific traffic conditions, typically observed in the real world. Our developments can bridge crowd simulation to extended application fields, such as Neuroscience, to decipher how humans behave in crowds. Given the wide attention received, developing this new generation of algorithms is very important.

The key idea of this thesis is to revisit the motion control scheme for algorithms based on synthetic vision and to propose a more developed technique with all the advances mentioned previously. The objectives of this work are:

- To develop a new control loop that is more robust to the complex situations which are often met when performing collision avoidance;
- To specify cost functions which use the visual input to evaluate the situation that the agent is in; and
- To propose and evaluate a new model for steering agents equipped with a synthetic vision mechanism, based on the control loop and on the cost functions previously defined. Such a model should consider all visible obstacles and explore all possible kinds of motion adaptations.

1.3 Proposal

In this work, it is proposed a motion control loop that is more robust to complex situations of collision avoidance. A locally optimal steering approach is adopted. A cost function that is used to characterize the current situation according to the visual information is defined. This cost function accounts for both the dangerousness of all visible obstacles and the agent's goal. At each time-step, the cost function and its gradient are evaluated. Agents adapt their motion by resorting to a gradient in order to minimize the cost function. Since the cost function accounts for both the goal and all the visible obstacles, this is equivalent to selecting a motion that corresponds to the best trade-off between reaching the agent's goal and reducing the dangerousness of the situation.

Compared to the previous algorithms based on synthetic vision, the proposed technique considers *all* visible obstacles whereas only dangerous obstacles were previously considered in other approaches. It explores *all* possible kinds of motion adaptations whereas agents previously reacted always the same way when escaping the risk of collision. The proposed method significantly improves the quality of crowd simulation results, while all the interesting properties of previous algorithms are preserved.

This work has two main contributions:

- A new motion control loop scheme for simulating crowds in a microscopic fashion; and
- The cost function to evaluate the situation of each agent with respect to its goal and risk of collision with nearby obstacles.

The new control scheme opens new directions for techniques based on synthetic vision and, more generally, for velocity-based crowd simulation models. As the proposed control scheme is deployed in the frame of algorithms based on synthetic vision, its principles can be reused in a different context. The cost function is one key-component of the proposed approach, where agents are able to perform locally optimal maneuvers by moving in accordance with the gradient of the proposed cost function. Such cost function can be integrated to other simulation algorithms or to an evaluation framework to estimate the relevance of avoidance maneuvers.

1.4 Organization

The remainder of this thesis is organized as follows. In Chapter 2, the most relevant related works are presented and discussed. Important approaches for crowd simulation are categorized and explained in details in this chapter. The importance of social behavior and methods for evaluating and validating virtual crowd behavior is also discussed. Finally, Ondřej's work on simulating crowds based on synthetic vision is detailed and discussed, for a better understanding of the improvements achieved with the model proposed in this thesis.

The contributions of this work are detailed in Chapter 3 where the proposed model is described. First, the new 3-phase control loop scheme is introduced as well as the mathematical formalism. In the following sections, the three phases of the loop (*perception*, *evaluation* and *action*) are presented in details.

Chapter 4 gives insights on technical aspects such as the model implementation. Moreover, the characterization of the agents' vision, i.e., the camera resolution, the field of view, etc., are detailed. Still in that chapter, the parameterization of the model is discussed. The model has four parameters which allow the agents' characterization with respect to speed adaptation, orientation adaptation, anticipation time and distance to keep from obstacles. First, it is shown how those parameters can affect the agent's behavior and, then, it is detailed how the model could be tuned according to experimental data.

After parameterizing, the results are presented and discussed in Chapter 5. In this chapter, the scenarios used for the model's evaluation are defined. The model is then evaluated and compared with a previous approach based on synthetic vision and a representative of geometrical velocity-based models. The evaluation is both qualitative and quantitative. Finally, the performance is also measured and discussed.

A summary of this work is made in Chapter 6. It is followed by a discussion about the current limitations of the model as well as the future directions to be taken.

2 Related work

This chapter is divided in two parts. In the first part, a review of the most relevant approaches for simulating crowd behavior found in the literature is performed. Those approaches are categorized and explained in separated sections. Then, a brief review about social behavior is presented. Planning the motion of crowds is not purely to set goals for the agents and to make them traverse the environment avoiding collisions. It is necessary to make these trajectories look as visually pleasant as possible. In the end of this part, it is discussed the problem of evaluating qualitatively crowd behavior and some existing solutions for this problem. In the second part, the model for simulating crowds based on synthetic vision introduced by (ONDŘEJ et al., 2010) is presented with more details. Then, the main drawbacks of this simple approach for simulating crowds using synthetic vision are discussed. This last discussion is followed by the proposals of this work to overcome the drawbacks found in (ONDŘEJ et al., 2010).

2.1 Crowd simulation approaches

The main objective of simulating crowds is to compute the motion of many characters which results from collective behaviors. This objective has received a wide attention from various disciplines and many solutions can be found in the literature as overviewed by recent books (PELECHANO et al., 2008; THALMANN; MUSSE, 2013; ALI et al., 2013). A full animation of a virtual crowd involves not only dealing with motion planning but also with several other aspects such as: avatars' animation and characterization; techniques for rendering crowds; and development of interfaces for manipulating crowds.

This work focuses on motion planning, or path planning. As briefly presented in the previous chapter, there are local and global path planners. For simple environments, just locally reacting to potential collisions can be enough for the agents to avoid collisions and to reach their goals, and this is what local path planners do. However, when the environment becomes more complex, local path planners can lead the agents to undesirable paths. A global path planner defines a set of waypoints between the initial position of the agent and its final goal. To reach its final goal, the agents must thus pass through each of the waypoints. The navigation between two waypoints is then managed by the local path planner.

In short, a global path planner is in charge of planning a route from the starting point until the goal considering the peculiarities of the environment, and a local path planner is in charge of adapting the route between waypoints so as the agents avoid collisions with local obstacles not tracked by the global path planner. For a better understanding of path planning, this section starts by addressing some literature regarding global path planning.

2.1.1 Global path planning

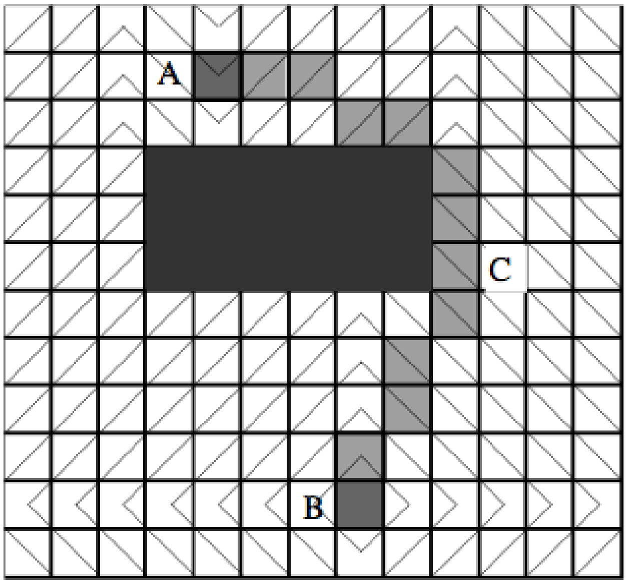
According to (KAPADIA; BADLER, 2013): "Navigation in arbitrarily large, complex environments requires an agent to be equipped with a mental model that provides some semantically meaningful geometric representation of the world around it". The agent needs a mental model, or map, of the environment that surrounds it and needs to be able to understand it and to use it to reach its goal through the best possible path. This mental map is created through the processing of the environment, where the existing navigable paths are established.

Most of the existing approaches for global path planning consist of computing the shortest path between vertices in a graph. What differs the approaches from each other are the details related to: the graph formation; the positioning of its vertices; the functions assigned to the vertices and edges; and the algorithm used for computing the shortest path. The approaches discussed in the following subsections were classified into five categories based on the literature (PELECHANO et al., 2008; THALMANN; MUSSE, 2013; ALI et al., 2013): discrete motion planning, probabilistic roadmaps, cell and portal graphs, reactive methods and environment modeling.

2.1.1.1 Discrete motion planning

As stated by (THALMANN; MUSSE, 2013): "Discrete methods are probably the most popular and simple in practice. The basic idea is to use a discrete representation of the environment: a 2D grid lying on the floor of an environment for navigation planning.". In this grid, the state of a cell is assigned according to the area which it represents within the environment. Thus, a cell can, basically, be marked as free or occupied by an obstacle. The movement is allowed between adjacent free cells: the problem of reaching a goal is reduced to a search for the shortest path from a given cell to this goal (THALMANN; MUSSE, 2013). An example of a path computed by this method can be seen in Figure 3.

Figure 3 – Example of a discrete method. The path between A and B is demonstrated by the cells in light gray in the grid, whereas the dark cells represent an obstacle.



Source: Bandi and Thalmann (1998).

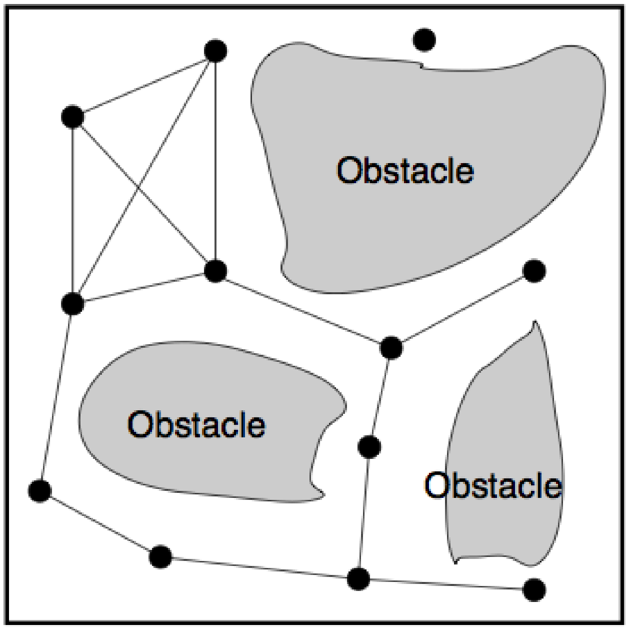
The search for the shortest path can be made by the well-known Dijkstra algorithm (DIJKSTRA, 1959), or by more efficient algorithms, such as the A* (HART et al., 1968), for example. Such algorithms are simple to implement and have strict guarantees of optimality and completeness of solution. Because of this, discrete methods are very popular and widely used for path planning, specially in games (KAPADIA; BADLER, 2013). However, according to (PELECHANO et al., 2008): "Although A* can find the shortest path to a goal and several improvements have been added to achieve fast solutions, it is still necessary to run the algorithm again to find a new path for each new goal and for each agent in the group".

The resolution of the discretization, i.e., the number of cells in the grid, affects both the performance and the quality of the trajectories. Coarse resolutions produce low-quality paths, whereas fine resolutions have a high computational cost, sometimes prohibitive for real-time applications (KAPADIA; BADLER, 2013). A solution for this problem of resolution could be the use of a hierarchical discretization, where the grid is refined according to the environment's geometry (THALMANN; MUSSE, 2013).

2.1.1.2 Probabilistic roadmaps

The Probabilistic Roadmap Method (PRM) (KAVRAKI et al., 1996) is intended to represent the environment in a very simplistic way, which consists of randomly distributing some points in free spaces within the environment and then connecting them creating a graph. A vertex must be connected to a neighbor vertex if and only if the straight line between them is collision free (Figure 4). With the generated graph, it is used a shortest path algorithm (such as in discrete methods) for obtaining the best path between an origin vertex and a destination vertex.

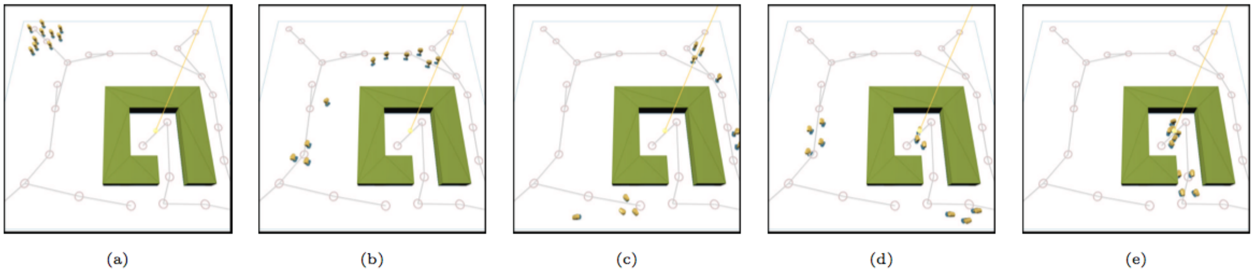
Figure 4 – Example of a roadmap.



Source: Bayazit et al. (2003).

PRMs are very popular and have been used to generate navigation paths for large groups of autonomous agents (PELECHANO et al., 2008). Several works have proposed improvements to this method, such as: heuristics or strategies for the roadmap creation (SIMÉON et al., 2000); pseudo-random sampling techniques (KUFFNER, 2004); or by manipulating PRMs to deal with group behavior such as homing, shepherding and exploring (Figure 5) (BAYAZIT et al., 2002; BAYAZIT et al., 2003; LIEN et al., 2005).

Figure 5 – Ten flock members are searching for an unknown goal. (a) The flock faces a branch point. (b) Since both edges have the same weight, the flock splits into two groups. (c) After dead ends are encountered in the lower left and upper right, edge weights leading to them are decreased. (d) As some members find the goal, edge weights leading to it are increased. (e) The remaining members reach the goal.

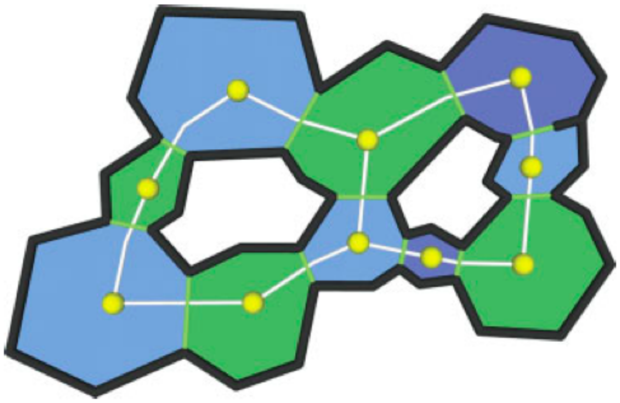


Source: Bayazit et al. (2003).

2.1.1.3 Cell and portal graphs

A Cell and Portal Graph (CPG) (TELLER, 1992) represents a method of abstracting the geometry of virtual environments. Generally, within indoor environments, the graph nodes (cells) indicate navigable regions such as rooms, whereas the portals represent entrance/exit points such as doors. For outdoor environments, cells can represent pedestrian pathways and portals are placed between pathways and crossings (LERNER et al., 2006; PELECHANO et al., 2008). After setting up the environment, the problem of navigating in a CPG is reduced to getting from a cell to another through a sequence of cells and portals (PETTRÉ et al., 2005; PELECHANO; BADLER, 2006; PELECHANO et al., 2008). Some examples of CPG can be seen in figures 6 and 7.

Figure 6 – An example of CPG. The portals are in green and the cells are represented by the colored polygons. A vertex is placed in each one of the cells and two cells are connected if they share a portal.

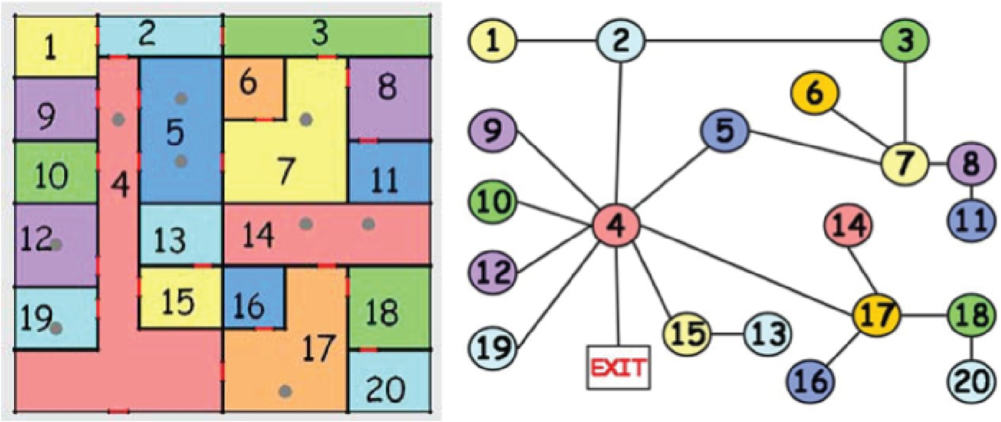


Source: Lerner et al. (2006).

2.1.1.4 Guidance fields

Guidance fields are used to steer agents through grids of vectors. These vectors help the agents to avoid obstacles and to move toward the goals specified as cells in the grids. Potential fields methods (KHATIB, 1985; WARREN, 1989; WARREN, 1990; SHIMODA et al., 2005) are based on guidance fields. In such an approach, a field of vectors composes a gradient which indicates the path to the goal. This field is generated

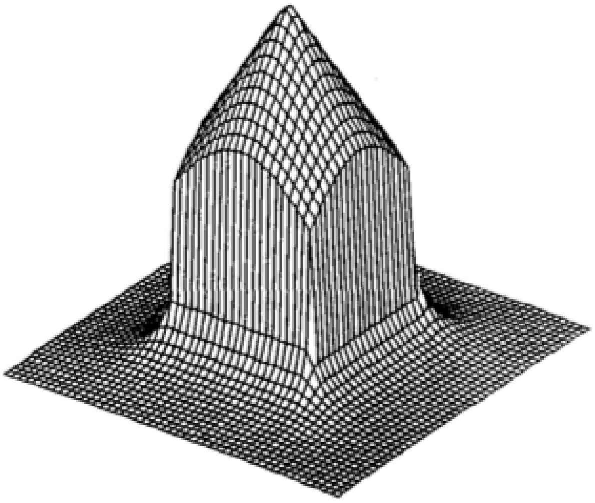
Figure 7 – Floor plan of a building and its corresponding CPG.



Source: Pelechano et al. (2008).

taking into account the repulsion caused by the obstacles and the attraction caused by the goal. Generally, in a potential field method, the environment is discretized in a regular grid, where each cell has a potential value which corresponds to the attraction and repulsion forces acting on it. Once the gradient is computed based on the potential values of each cell, it is possible to follow it to reach low potential values, usually representing a goal. This method has the same problems regarding discretization as those found in discrete methods. Figure 8 shows a representation of a potential field in an environment with a central obstacle.

Figure 8 – Potential field representing an environment with a central obstacle.

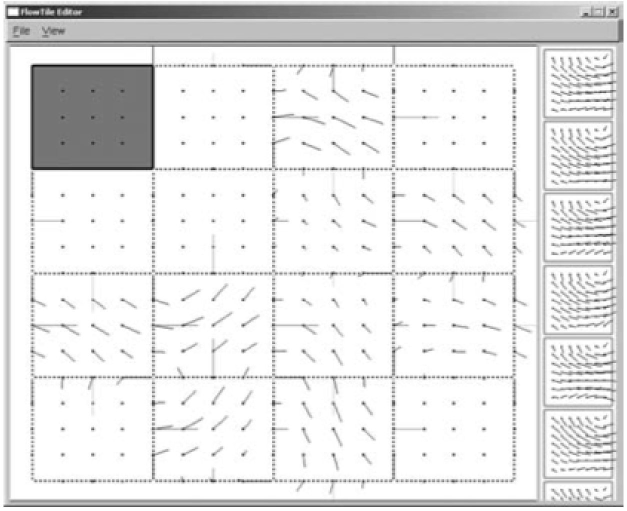


Source: Warren (1989).

In (CHENNEY, 2004), the author proposed a similar approach where tiles with precomputed vector fields are used to compose the field of vectors representing large flows. In other words, through the production of tiles with different configurations it is possible to create scalable scenarios with varied flows. Figure 9 shows the tool used for composing scenarios with tiles.

Dynamic potential fields (TREUILLE et al., 2006) have been used for integrating global navigation with dynamic obstacles and agents, solving the problem of moving crowds in an efficient way without the need of

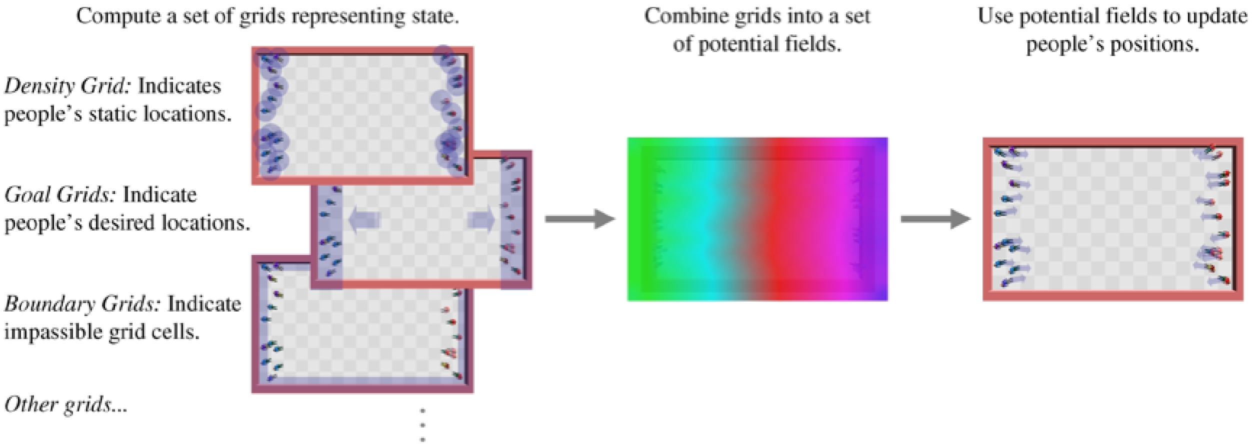
Figure 9 – Tool used for composing scenarios with precomputed fields of vectors.



Source: Chenney (2004).

explicit collision avoidance. For this method, the grids' discretization highly affects the performance because the potential fields are dynamic and need to be computed for every simulation's time-step. Figure 10 shows an overview of the algorithm proposed in (TREUILLE et al., 2006).

Figure 10 – Overview of the algorithm proposed for creating dynamic potential fields.


Compute a set of grids representing state.
Combine grids into a set of potential fields.
Use potential fields to update people's positions.
Density Grid: Indicates people's static locations.
Goal Grids: Indicate people's desired locations.
Boundary Grids: Indicate impassible grid cells.
Other grids...

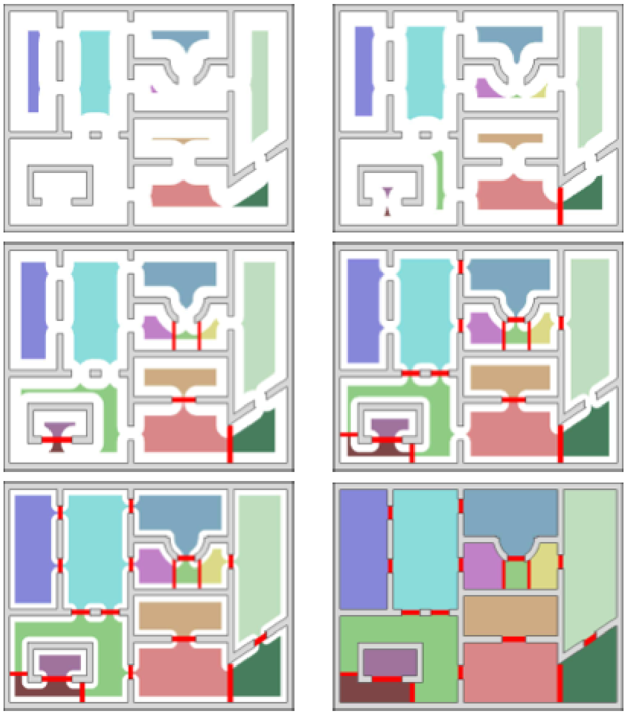
Source: Treuille et al. (2006).

2.1.1.5 Environment modeling

In this approach, the better the topological representation of the environment for the method used for path planning, the better the paths will be. Manual methods can be used to represent the environment where an artist or level designer can distribute waypoints along the path within the scenario. However, a more accurate alternative is to define navigation meshes, for example. Nevertheless, as stated by (GRÖSCHEL, 2011): "although the process of manually creating the world representation is versatile, it is considered to be very labor intense since great care must be taken to ensure validity".

For that reason, techniques for automatic creation of topological representations have been developed. In (HAUMONT et al., 2003), the authors presented an algorithm for generating volumetric CPGs for indoor scenarios based on an adaptation of the 3-D watershed transform algorithm, computed on a distance-to-geometry sampled field. The environment is "flooded" from the local minima, and each minimum produces a region (room). Portals are created between regions when they get in contact during the flooding process (Figure 11). The algorithm classifies, automatically, each room as a cell and the openings (doors and windows) as portals, thus, being able to generate the CPG of any indoor environment.

Figure 11 – Flooding process used to generate the environment's CPG.

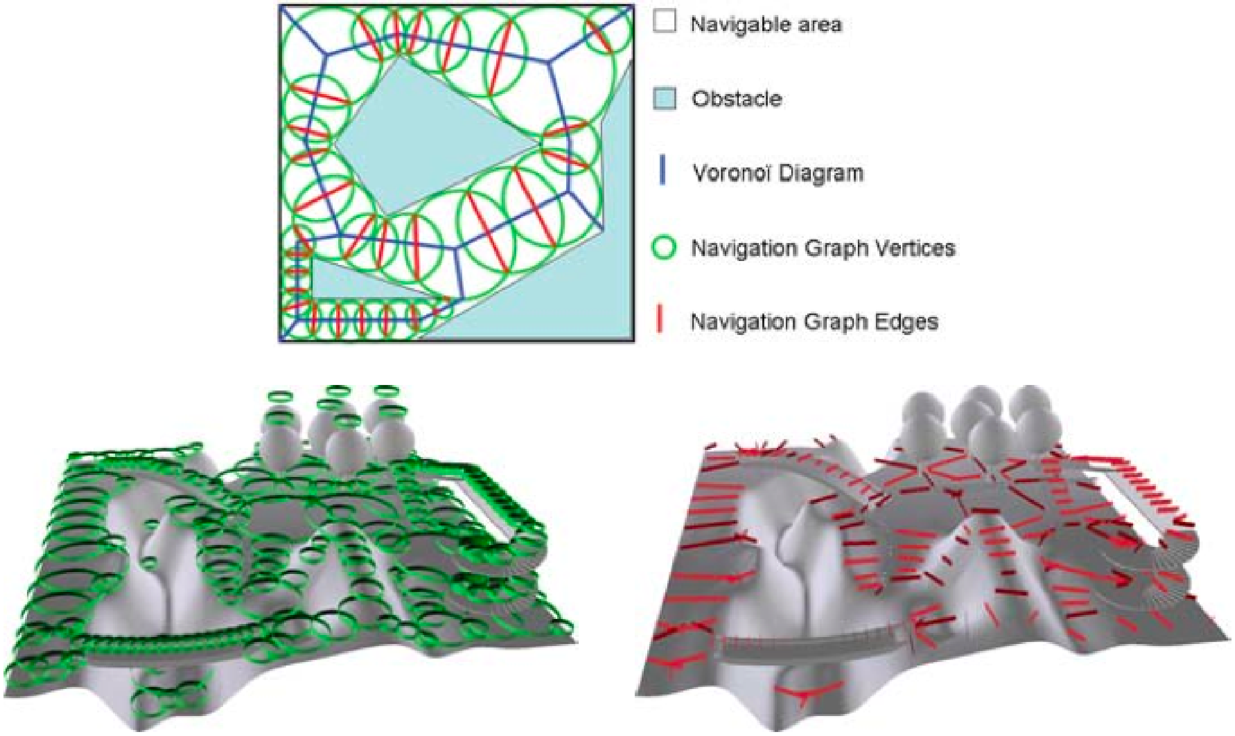


Source: Haumont et al. (2003).

In (PETTRÉ et al., 2005), the authors proposed an approach based on a spatial structuring technique that automatically decomposes multilayered or uneven terrains into corridors giving rise to a navigation graph to be used for path planning. In this method, the space is divided into free spaces and obstacles to be avoided. At first, it computes a Voronoi diagram of the free-space and, then, it builds a set of collision-free convex cells along the diagram. The navigation graph is obtained from the adjacency graph of the cells (Figure 12). The novelty of this work was to extend a basic navigation graph for terrains with multiple layers by classifying some areas with free spaces as obstacles based on the terrain's slope.

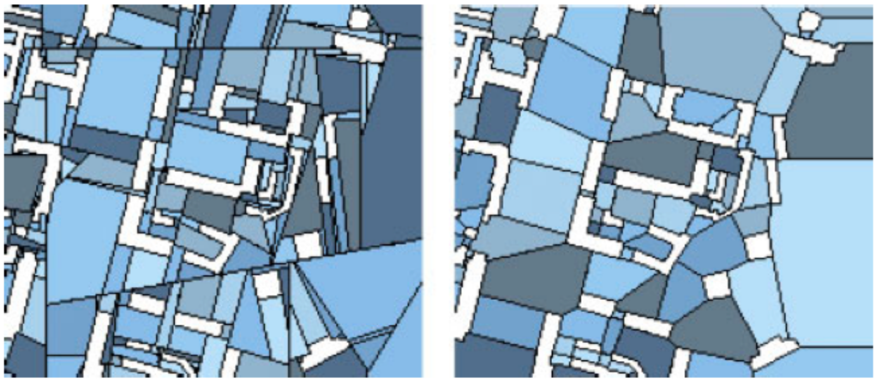
Lerner et al. (2006) defined an effectiveness measure for a cells-and-portals partition. With this metric, they introduced a two-pass algorithm for computing CPGs, in which the first pass creates an initial partition and the second pass refines it. The algorithm uses a simple heuristic that strives to create small portals as a means for generating an effective partition. The algorithm's input is a set of half edges (in 2-D) which can be extracted from a complex polygonal model. The method supports incremental changes of the model by locally recomputing and updating the partition, and can be used for generating CPGs for indoor and outdoor environments. In Figure 13, an example of partition produced by this method can be seen.

Figure 12 – Navigation Graphs principles. Top: example of a Navigation Graph in a 2-D academic example. Bottom: Vertices (left) and Edges (right) of a Navigation Graph computed for a natural scene.



Source: Thalmann and Musse (2013).

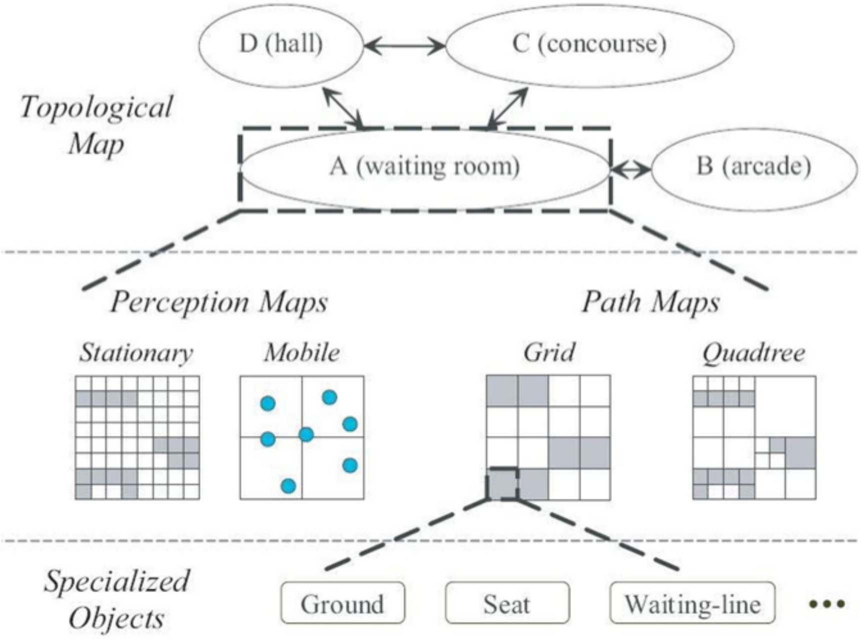
Figure 13 – Example of partitioned scenario, where the buildings are in white and partition cells are in blue. On the left, a BSP (Binary Space Partitioning) partition and, on the right, a partition produced by the algorithm proposed by (LERNER et al., 2006)



Source: Lerner et al. (2006).

Shao and Terzopoulos (2007) represented virtual environments using a hierarchical collection of maps (Figure 14). Each map was created for a specific purpose and the combination of them helps to manage crowds in an efficient way. The model comprises: a topological map, which represents the topological structure of the virtual environment; a set of perception maps, associated to the topological map, which provides relevant information for perceptual queries; and a set of path maps which enables online path planning for navigation. The topological map contains nodes which correspond to regions within the environment and edges representing accessibility between regions. The path maps include a quadtree map which supports global, long-range path planning and a grid map which supports short-range path planning.

Figure 14 – Hierarchical representation of a building.



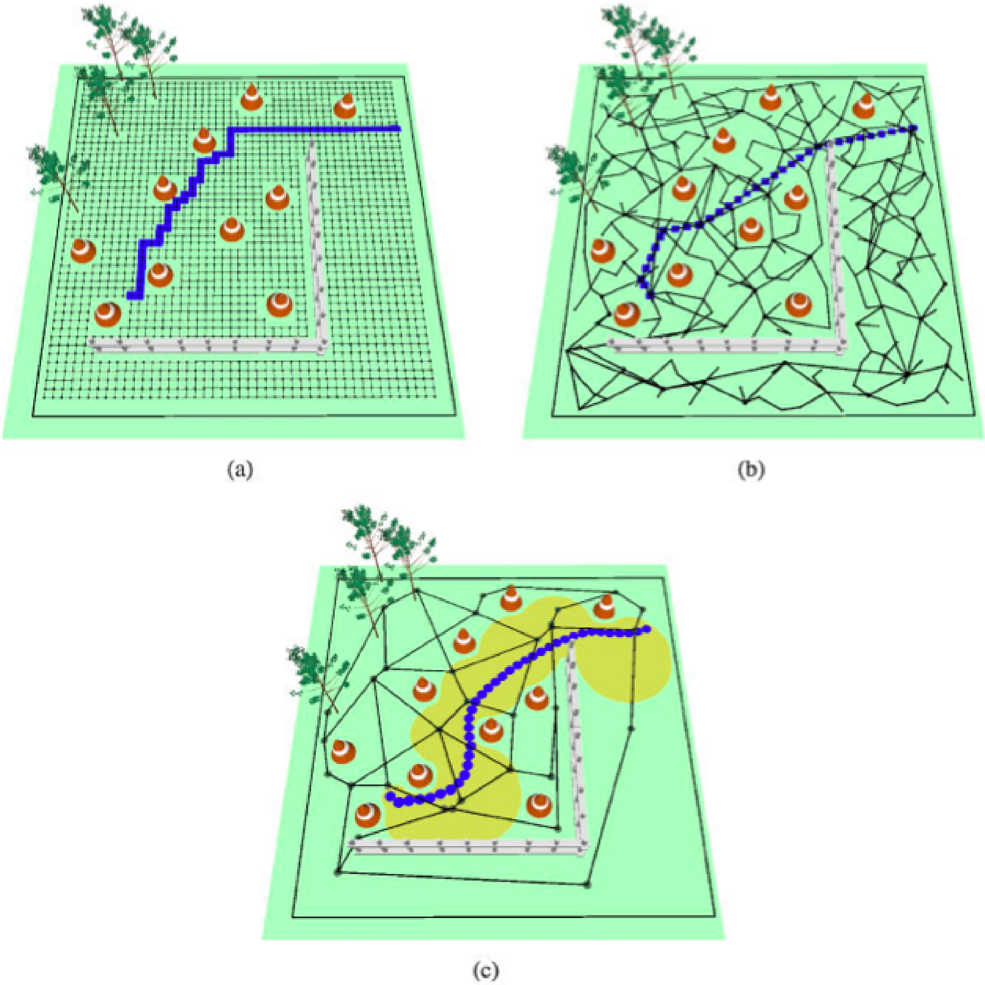
Source: Shao and Terzopoulos (2007).

In (GERAERTS; OVERMARS, 2007), the authors presented the Corridor Map Method (CMM). That method, in an offline construction phase, creates a system of collision-free corridors for the static obstacles in an environment, and then, in the query phase, it plans paths inside the corridors for different types of characters that avoid dynamic obstacles. The authors use a method based on medial axis to represent free-space as a graph where the edges correspond to collision-free corridors. Each edge of the graph encodes a local path together with a maximum clearance radius which can be used to find paths with arbitrary clearance. Figure 15 shows a comparison of the CMM with two other methods.

Several methods aiming at reproducing free-space within environments by using meshes have been proposed. In (LAMARCHE; DONIKIAN, 2004), the authors used the Delaunay triangulation to compute a subdivision of convex cells of the free-space while maintaining the information about the local bottlenecks to represent the topological connectivity of 3-D environments. Other approaches explore the Voronoi diagram, as in (HOFF III et al., 1999) where a Voronoi representation of the static environment is precomputed for path planning; and in (SUD et al., 2007a) where the authors presented the Multi-agent Navigation Graph (MaNG) which is built dynamically using discrete Voronoi diagrams. The MaNG is used to compute simultaneously the maximum clearance paths for a set of agents which move with independent goals. In (KALLMANN, 2010), the author proposes a method for finding optimal paths with arbitrary clearance directly from a triangulation. The method proposed introduces a new local clearance property which facilitates the efficient computation of the path clearance in a triangulated mesh.

Recently, adaptive elastic roadmaps (SUD et al., 2007b; GAYLE et al., 2009) have been used to compute near-optimal paths within virtual environments, where these roadmaps adapt themselves according to the

Figure 15 – Three methods for path finding. (a) The A* algorithm finds the shortest path in the displayed grid, consisting of 1792 nodes and 3321 edges. (b) The PRM-graph is almost six times as small. (c) The CMM-graph is the smallest one containing 44 nodes and 50 edges.



Source: Geraerts and Overmars (2007).

dynamic obstacles and the interaction forces among the agents.

2.1.2 Local path planning

Local path planning can be defined as:

> The layer of intelligence that interfaces with navigation to move an agent along its planned path by performing a series of successive local searches, taking into consideration locomotion constraints such as turning capabilities and limits on movement velocity, as well as dynamic objects in the environment such as other agents (KAPADIA; BADLER, 2013).

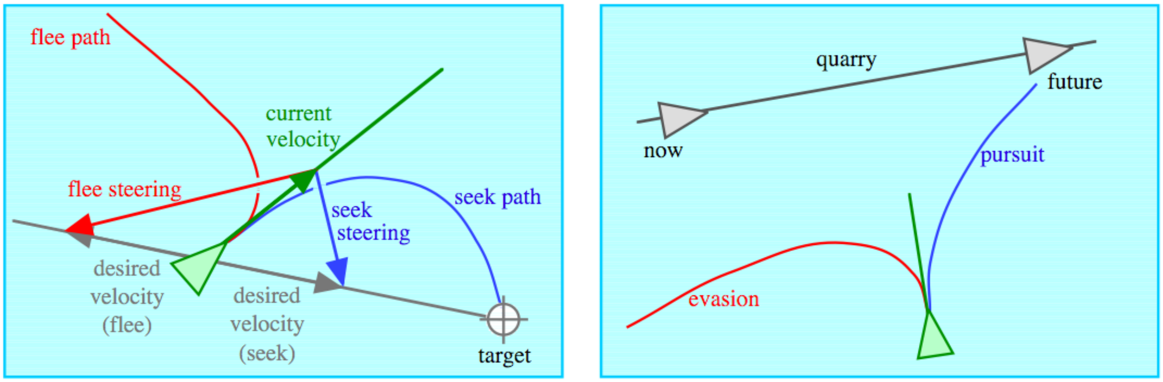
This is one of the most explored subareas in Crowd Simulation given the importance of the agents' motion visual quality. The need for agents behaving as similarly as possible to real crowds motivates the large number of research works on this field. The more realistic the agents' animated trajectories look, the more believable, or immersive, the virtual world will be.

The various existing solutions for this problem can be classified in several ways according to their common features. Some authors choose to classify them according to the way the agents are managed. In this case the approaches can be classified as: macroscopic or microscopic. Macroscopic approaches are concerned with the global crowd flow, regardless of the local interactions between agents, whereas microscopic approaches model local interactions between agents which influence each other's motion. In the case of microscopic approaches, collective behaviors and global patterns emerge as a consequence of the agent's local interactions. Other authors prefer to classify approaches according to their abilities of collision prediction. In this case, the different approaches are classified as predictive or reactive, depending on whether they allow anticipated reactions based on motion prediction or purely reactive motion. In (ZHENG et al., 2009), the authors present some features to classify approaches for crowd evacuation simulation. In this section, the algorithms were classified in the following categories based on the techniques used for collision avoidance: rule-based, particle-based, cellular automata, guidance fields, velocity-based, data-driven, based on synthetic vision and hybrid.

2.1.2.1 Rule-based models

In 1987, Reynolds presented a rule-based model, where the concept of *boids* ("bird-oid" contraction) was introduced. The author made these agents called boids behave as a flock through the combination of three simple individual rules, where the boids should: avoid collisions (separation), keep the same velocity (alignment) and stay close to each other (cohesion). Rule-based models allow characterizing the agents individually with unique behaviors not only focusing on collision avoidance, making it possible to model heterogeneous agents with complex behaviors formed by the combination of simple behaviors. In 1999, Reynolds expanded his previous work by defining steering behaviors for autonomous agents (e.g., seek, flee, pursuit, evasion) (Figure 16).

Figure 16 – Examples of rules. Seek and flee (left). Pursuit and evasion (right).



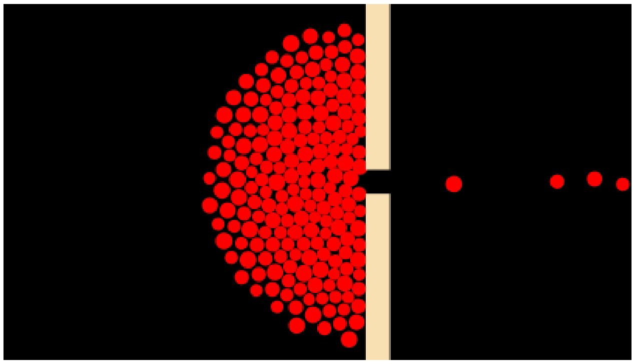
Source: Reynolds (1999).

Rule-based models have the advantage of allowing the modeling of heterogeneous agents. However, in some cases, it can be difficult to combine rules, leading to the emergence of unpleasant artifacts in the agent behavior (contradictory rules, for example). Moreover, it can be computationally expensive to handle crowds with several behaviors. For achieving better performance results, allowing real-time simulations with several agents, some techniques have been proposed, such as: spatial partitioning (REYNOLDS, 2000; REYNOLDS, 2006), hierarchical grouping (MUSSE, 2000; MUSSE; THALMANN, 2001) or the use of hierarchical structures (LAMARCHE; DONIKIAN, 2004; SHAO; TERZOPOULOS, 2007).

2.1.2.2 Particle-based models

The interest in particle-based models arose given their ability to simulate crowd evacuation from buildings considering physical aspects. In (HELBING; MOLNÁR, 1995), the authors proposed a model where the agents move according to repulsive forces (exerted by other agents, objects, walls, etc.) and attractive forces (exerted by friends, objectives, etc.) trying to reach the position of their goals within the environment as comfortably as possible. This model naturally simulates agents' self-organization in collective phenomena. Combining socio-psychological and physical forces, Helbing et al. (2000) proposed a particle-based model capable of simulating crowds in panic situations. The algorithm managed to reproduce many observed phenomena including: clogging effects at bottlenecks (Figure 17) and the corresponding increase of pressure, jamming at widenings, the "faster-is-slower" effect, inefficient use of alternative exits and initiation of panics by couterflows and impatience.

Figure 17 – Clogging effect at a bottleneck reproduced with Helbing's model.



Source: Helbing et al. (2000).

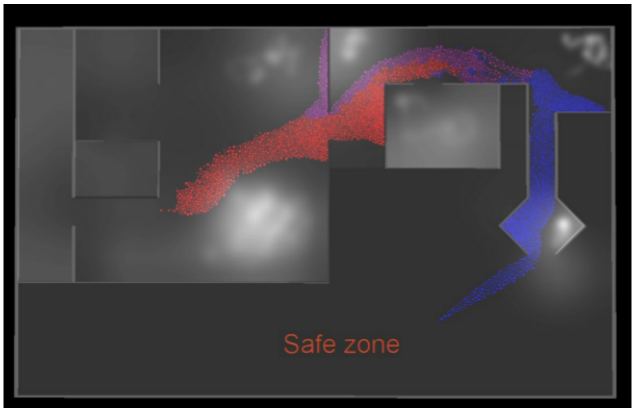
This type of model can be adapted to many situations just by adding new attractive and repulsive forces, according to the behavior expected for the crowd. In (COURTY; MUSSE, 2005), a repulsion force was added to Helbing's model (HELBING et al., 2000) so as to make the crowd avoid places with smoke while trying to evacuate a building (Figure 18). In (BRAUN et al., 2003), the authors extended Helbing's model to include individualism. Particle-based models have also been used for studying group behavior and their effects on crowd dynamics (MOUSSAÏD et al., 2010; XU; DUH, 2010). Despite being particularly suitable for simulating emergency situations, this type of model can also be adapted for simulating common situations as can be seen in (PELECHANO et al., 2007).

A different approach was proposed by (HEIGEAS et al., 2003). In this case, the agents' interactions were modeled as a mass-spring-damper system, where stiffness and viscosity terms change with respect to the relative distance between the agents. Particle-based models are inherently reactive, i.e., agents do not anticipate their motion with respect to other agents in collision course, resulting in visually unpleasant artifacts in sparse environments. However, it is possible to minimize this drawback by adding an evasion force so as to make the agents react to future collisions in advance, as presented in (KARAMOUZAS et al., 2009).

2.1.2.3 Cellular automata models

Cellular automata are discrete dynamic systems which consist of a regular two-dimensional grid of cells. A cell can be occupied (by an obstacle), free or it can have another state depending on the behavior to be

Figure 18 – Crowd being repulsed by places with smoke while evacuating a building.



Source: Courty and Musse (2005).

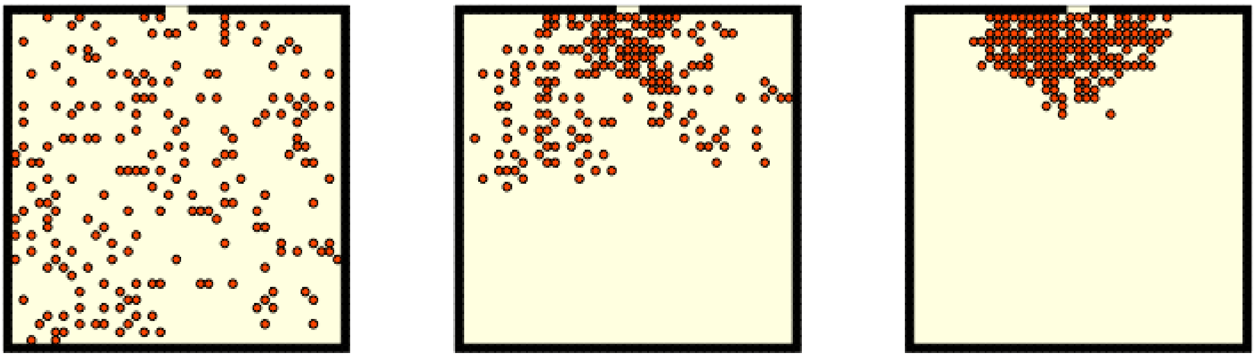
simulated.

> A cellular automaton evolves in discrete time steps, with the value of the variable at one cell being affected by the values of variables at the neighboring cells. The variables at each cell are updated simultaneously based on the values of the variables in their neighborhood at the previous time-step and according to a set of local rules (PELECHANO et al., 2008; WOLFRAM, 1983).

When used for simulating crowds, those rules are defined to control the agents' behaviors during the simulation.

Although fast and simple to implement, cellular automata models (DIJKSTRA et al., 2001; SCHADSCHNEIDER, 2001; TECCHIA et al., 2001; BURSTEDDE et al., 2001; KIRCHNER; SCHADSCHNEIDER, 2002; KIRCHNER et al., 2003) do not allow physical contact between agents, because each cell can be occupied by only one agent each time. This restriction causes this type of model to reproduce unrealistic results in high density situations (Figure 19), making them impracticable for some applications such as entertainment. Aiming at creating more realistic behavior in high density situations, in (LOSCOS et al., 2003), the authors adapted their model for collision avoidance to deal with several situations, enabling the cooperation between agents for decision-making, allowing the emergence of pedestrian flow in the crowd.

Figure 19 – Room evacuation simulation using a cellular automata model. It is possible to notice the organization of the agents in a regular grid.

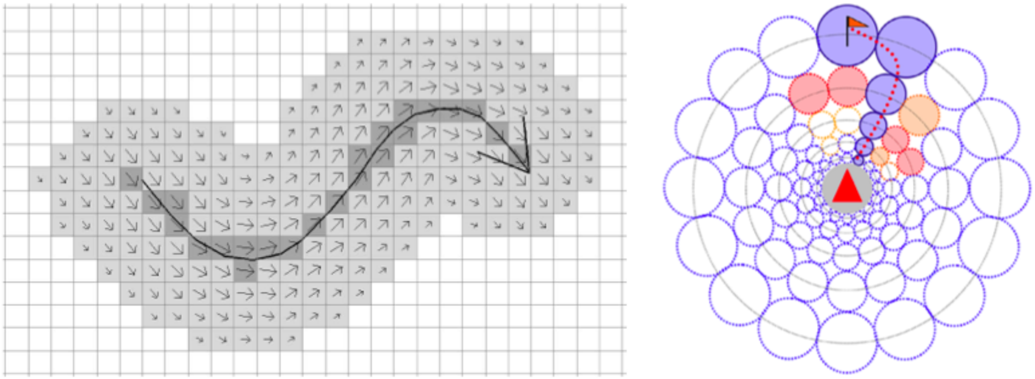


Source: Burstedde et al. (2001).

2.1.2.4 Guidance field models

Models based on guidance fields can be used for both global (Section 2.1.1.4) and local path planning. As presented before, guidance fields can be manually defined by setting flow tiles to compose the field (CHENNEY, 2004). In (PATIL et al., 2011), the field is generated procedurally, but it can be influenced by user interaction (Figure 20 left) or by motion flow fields extracted from crowd video footage. Kapadia et al. (2009b) use egocentric fields to determine the optimal path that an agent can take at short-term (Figure 20 right). The main drawback of using local guidance fields is that they can easily lead agents to local minima.

Figure 20 – Example of guidance fields. Field updated by user interaction (left) and egocentric field (right).



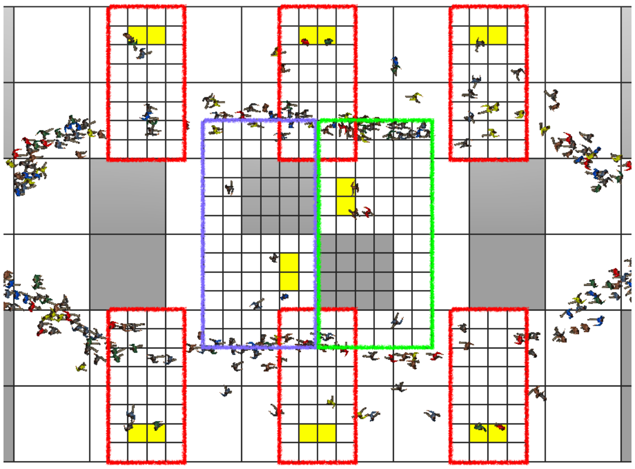
Source: Patil et al. (2011) (left) and Kapadia et al. (2009b) (right).

As it was briefly presented in Section 2.1.1.4, guidance fields can also be generated by using potential fields. In this case, each cell in the grid has a potential positive value which decreases according to the proximity of the goal. Goal cells must have a zero value so as to attract the agents, whereas obstacle cells must have high values to repulse them. This way, a gradient is formed indicating the path from each cell to the goal. Hughes (2002, 2003) presented a model where partial differential equations were defined to describe crowd dynamics. In this model, the agents are converted to a density field which is used to generate the potential field. Treuille et al. (2006) improved Hughes's model to reach more realistic crowd behavior. Their model separates agents in groups with common goals, where these groups represent the environment as a grid of cells. Then, for each group, a potential field is computed. In (JIANG et al., 2010), the authors adapted Treuille's model to deal with more complex environments. Models based on dynamic potential fields have as main advantage the possibility of simulating realistically the behavior of several agents in real-time. Nevertheless the process of generating potential fields is computationally expensive and, consequently, only a small number of groups (goals) is supported. Recently, Dutra et al. (2013) proposed a multipotential field method for allowing scalable behaviors in Treuille's model (Figure 21).

2.1.2.5 Velocity-based models

Recently, velocity-based approaches have received increased attention. Such models anticipate the danger by extrapolating the agents' trajectories to detect potential collisions in a near future. Extrapolating trajectories for collision prediction is not new. In 1999, Reynolds presented the unaligned collision avoidance behavior for this purpose and, in the following years, other works in this direction have been presented. In (PARIS et al., 2007), the authors presented a model which represented a great advance to the early work of Reynolds, specially by solving major existing drawbacks such as oscillations and jams due to the lack of anticipation. Moreover, the model was calibrated and validated from real data.

Figure 21 – Example of multipotential fields. Eight potential fields are used to change momentarily the agents' main goals.



Source: Dutra et al. (2013).

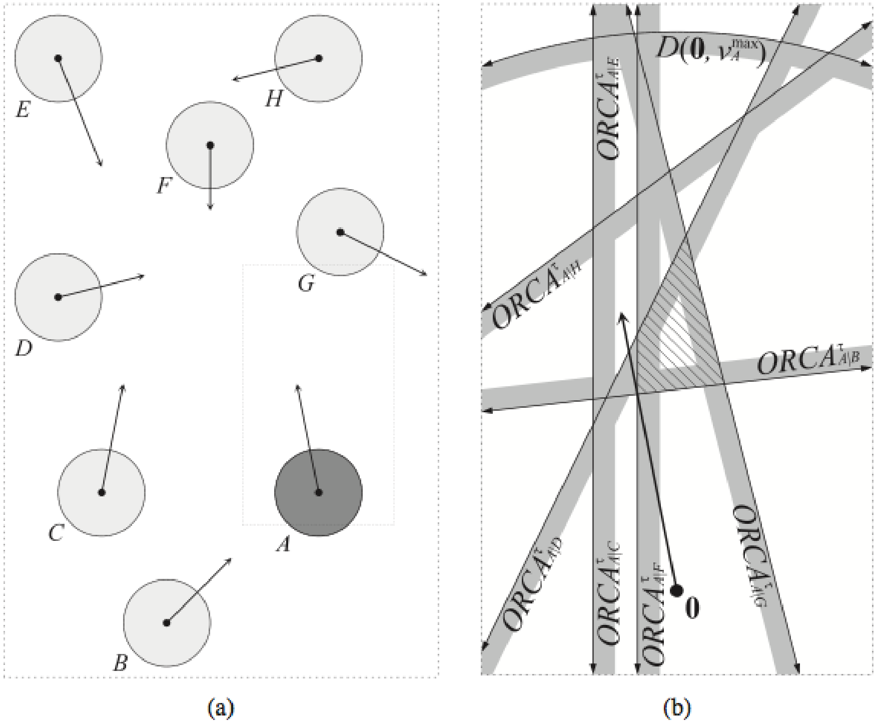
In the field of robotics, there is the concept of Velocity Obstacle (VO), introduced by Fiorini and Shiller (1998), where a robot is capable of avoiding collisions with obstacles based on their velocities. The method consists of selecting a velocity in the velocity-space which allows the robot to avoid collisions with the static and moving obstacles based on their positions and velocities. This concept has been widely used by velocity-based models for simulating crowd behavior. In 2008, van den Berg et al. (2008a, 2008b) presented the concept of Reciprocal Velocity Obstacle (RVO) which extends the VO concept to guarantee safe and oscillation-free navigation among agents by considering the reactive behavior of the other agents, assuming that the agents avoid each other in the same way. Since then many contributions have been made to this approach. In (GUY et al., 2009), the authors presented a parallel algorithm which uses a discrete optimization method. The most recent evolution of velocity-based obstacles is represented by the Optimal Reciprocal Collision Avoidance (ORCA) approach (BERG et al., 2011), which efficiently computes the optimal solution (maximum collision free velocity closer to the comfort velocity) in the velocity-space, hence reciprocally avoiding collisions between agents in a near future (Figure 22).

This type of model can treat collisions among several agents and obstacles efficiently (thanks to its parallelizable nature) allowing to simulate crowds with visually pleasant trajectories in real-time; however this realism decreases when the crowd density increases. They can also expose artifacts when dealing with symmetric situations.

2.1.2.6 Data-driven models

Data-driven models are those that use some kind of data to determine the virtual crowd behavior. This data can be real crowd footage, trajectories extracted from experiments, the resulting output of some other model, or even generated by the user. In (METOYER; HODGINS, 2004), the user can adjust the agents' motion in a simulation in real-time and, according to these interactions, the simulator learns how to treat the adjusted situations. Lee et al. (2007) presented a model where the behavior's patterns of each agent are learned from real crowd footage (Figure 23). Lerner et al., in 2007, described a model where, given a set of trajectories extracted from a real crowd footage, a database of examples is built and the agents' trajectories are incrementally synthesized considering spatio-temporal relationships with nearby agents and obstacles,

Figure 22 – (a) A configuration with eight agents. Their current velocities are shown using arrows. (b) The half-planes of permitted velocities for agent A induced by each of the other agents. The dashed region contains the velocities for A that are permitted with respect to all other agents. The arrow indicates the current velocity of A.



Source: Van den Berg et al. (2011).

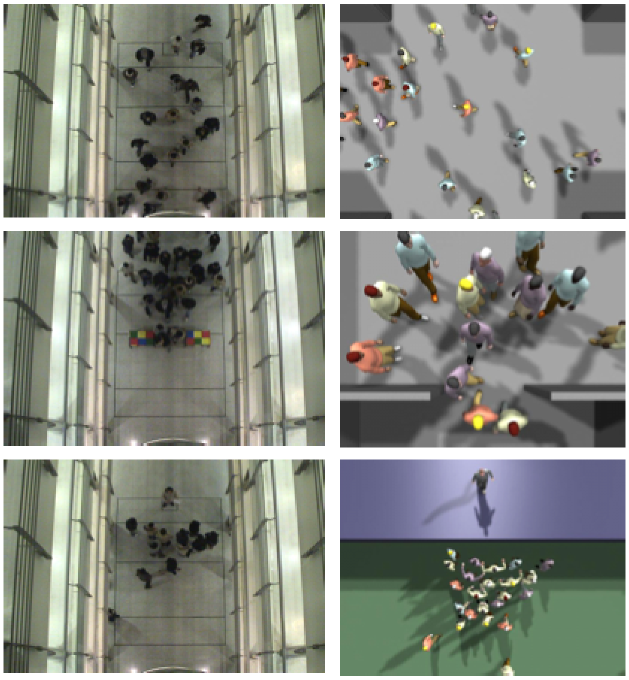
and according to similar scenarios found in the database.

Pettré et al. (2009) presented a model able to simulate and reproduce experimental trajectories obtained from observations of real interactions between walkers. In (LERNER et al., 2009), the authors developed a method which, given a database of behaviors extracted from real crowd footage, is able to select from the database the behavior which is the most representative of the agent's current situation. Paravisi et al. (2008) adapted Treuille's model (TREUILLE et al., 2006) to reproduce crowd and group behavior extracted from video footage. The authors, in (JU et al., 2010), presented a method which combines different data from several existing crowds to generate a new crowd animation. This data can be obtained from real crowds or virtual ones simulated by other models (Figure 24).

As stated initially, the input data need not necessarily be obtained through computer vision techniques as in (GUY et al., 2011), where the authors use perceptual studies to affect the agents' personalities, thus generating heterogeneous crowds. Recently, in (CHARALAMBOUS; CHRYSANTHOU, 2014), the authors introduced a structure called perception-action graph (PAG) for accelerating and improving the quality of data-driven crowds. The PAG handles the input examples as a graph, which is used at run-time to efficiently synthesize believable virtual crowds.

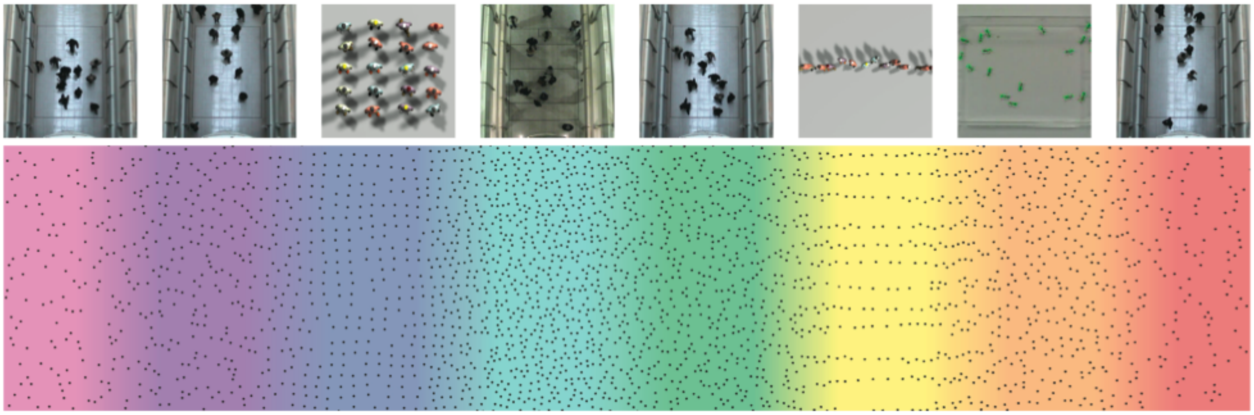
This type of model has some drawbacks, such as: the need of maintaining a database of the recorded behaviors, which can occupy plenty of space according to the number of behaviors to reproduce and increase the searching time for a specific behavior; or, the need of input data according to the situation to reproduce,

Figure 23 – Real data (left) and reproduced behavior (right).



Source: Lee et al. (2007).

Figure 24 – Real data (top) and reproduced behavior (bottom).



Source: Ju et al. (2010).

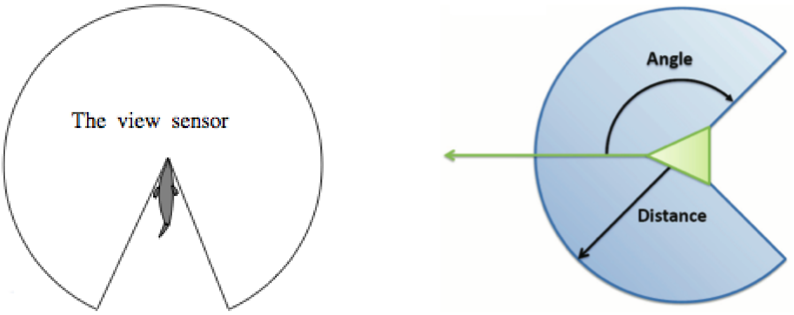
i.e., a behavior recorded in a specific situation may not work in a different context.

2.1.2.7 Models based on synthetic vision

A specific category of approaches is based on a Synthetic Vision (SV) system which equips each agent. Synthetic vision models mimic how humans visually perceive risk of future collision and how they react accordingly. These algorithms are inspired by literature which acknowledges the role of the human vision system in the locomotion perception-action loop (CUTTING et al., 1995; WARREN; FAJEN, 2004; RIO et al., 2014). Early attempts modeled the agents' field of view as a geometrical area (REYNOLDS, 1987; TU; TERZOPOULOS, 1994) (Figure 25) or as a volumetric representation of the scene (SILVA et al., 2010).

Agents would only interact if falling in such field of view.

Figure 25 – Vision as a geometrical area. A vision of a fish (left) and a vision of a boid (right).

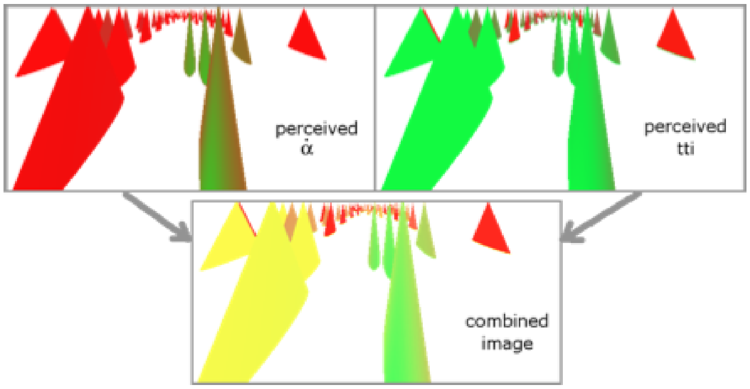


Source: Tu and Terzopoulos (1994) (left) and Silva et al. (2010) (right).

The first explicit simulation of locomotion using the agents' vision was introduced in (RENAULT et al., 1990). In (NOSER et al., 1995), the authors used the agents' vision to identify objects within an environment. Kuffner and Latombe (1999) used synthetic vision to allow agents to explore and to navigate within unknown environments. The authors in (PETERS; O'SULLIVAN, 2003) used synthetic vision to find interesting areas in the agent's vision which could attract its attention and thus improve the feeling of presence within the virtual environment.

Ondřej et al. (2010) proposed a novel synthetic-vision approach for crowd simulation. Their model transforms the visual input of each agent into images containing information which allows detecting risk of collisions with any obstacle or agent in the scene (Figure 26). Agents react to the stimuli by turning to avoid a future collision when detected with anticipation and slowing down to avoid an imminent collision. Despite the good results reported, Ondřej's model suffers from important drawbacks. This model and the model proposed by this thesis are strongly related, for this reason Section 2.2 is dedicated to present and to discuss the Ondřej's model with more details.

Figure 26 – Representation of the agent's visual perception.



Source: Ondřej et al. (2010).

In 2011, Moussaïd et al. presented a vision-based approach for simulating pedestrian behavior based on two rules. This model is purely reactive as well as Ondřej's. The first rule is used to adapt the agent's orientation in order to allow a small deviation to the goal according to the obstacles' positions whereas the second rule is used to adapt speed according to the agent's reaction time and the first obstacle in the walking direction. Body collisions are avoided by using a particle-based algorithm. Despite resorting to the agent's vision, it is not clear in the article whether this vision is a synthetic one or a geometrical representation. Recently, Rio et al. (2014) investigated the optical information used to control walking speed in pedestrian following. From the results of this investigation, the authors could derive a visual control law for one-dimensional following. This law is based on the optical expansion of the follower, where the follower accelerates if the leader's visual angle is decreasing, decelerates if it is increasing, and maintains the current speed if visual angle is constant.

Models based on synthetic vision are very promising, allowing reproducing behaviors more closely to reality than those produced by traditional models, provided that the data obtained through the vision is interpreted properly. Those results come with a high computational cost, although, the rapid evolution of graphic cards suggests this approach will soon become more popular for real-time applications such as games.

2.1.2.8 Hybrid models

Each type of model has advantages and disadvantages. Thus, the idea of combining the best features of the existing models gave rise to the hybrid models. In (PELECHANO et al., 2007), the authors captured the best aspects of rule-based and particle-based models, and created a new model which used psychological, physiological and geometrical rules combined with physical forces to reproduce heterogeneous agents and to guide them within an environment. Yersin et al. (2008) proposed an architecture for simulating crowds which divides the environment in three regions of interest (ROI) according to the distance to the camera (the closer, the more important). Each ROI is ruled by a different technique of path planning. In regions of no interest, the planning is ruled by a navigation graph and collisions are not avoided; in regions of low interest, the planning is also ruled by a navigation graph and collisions are avoided resorting to the Reynolds's concepts (1999); and finally, in high interest regions, both path planning and collision avoidance are ruled by potential fields, similar to (TREUILLE et al., 2006).

In (XIONG et al., 2010), the authors proposed an architecture where two models (a macroscopic and a microscopic) coexist in a simulation and work in a collaborative way, resorting to the benefits of each when necessary. The environment is divided in partitions and each of them is ruled by one of the models at a given instant. Narain et al. (2009) presented a model based on potential fields, but which solves local collisions using a geometrical model. First, the path of the agents is planned globally and their comfort velocities are set; then, locally, their velocities are adapted according to the density of the cell where the agents lie; after defining the velocities, the minimum distance among the agents is assured. In (SINGH et al., 2011), the authors described a framework that integrates multiple models which are used according to the agent's current situation. Finally, the authors, in (GOLAS et al., 2014), proposed an approach which blends results from continuum and discrete algorithms, based on local density and velocity variance. Their hybrid method has seamless transitions between the continuum and discrete representations.

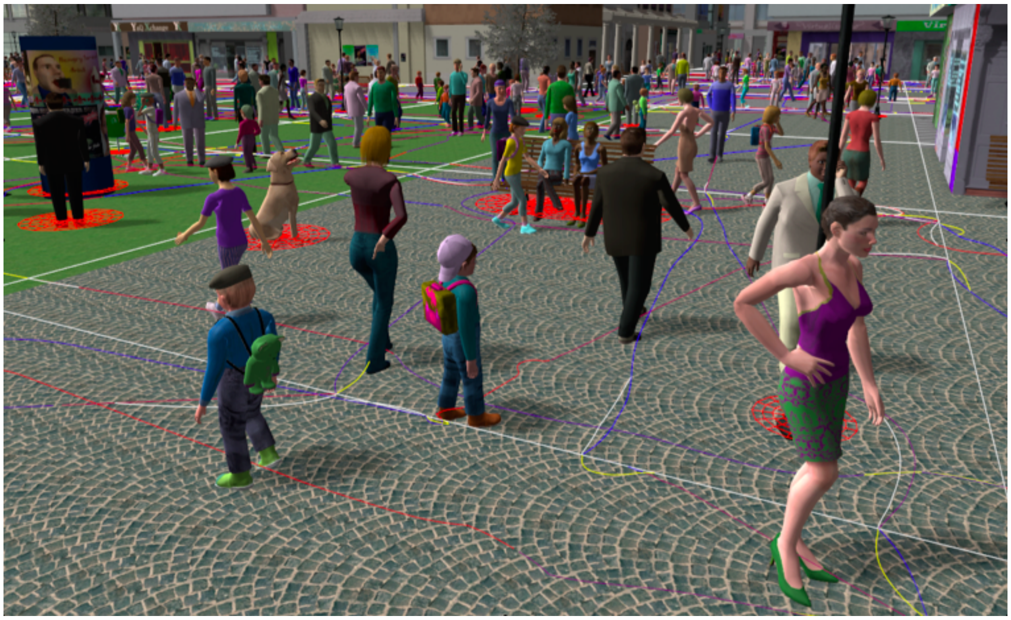
Hybrid models are able to deal with more varied situations given the flexibility acquired by integrating models which solve different problems related to crowd path planning. However, when working with different models, it is necessary to take into account the complexity of integrating them so that they can work together properly.

2.1.3 Path planning remarks

As it was stated at the beginning of this chapter, path planning is a research field widely explored, motivated by several problems found in the robotics field, entertaining industry and civil engineering, just to name a few. The works presented here are just some of the most relevant works that can be found in the literature. Most of the works can fit into more the one classification, for example, models based on synthetic vision can also be considered velocity-based approaches, since the agents take into account the movement of the obstacles to anticipate their motion. On the other hand, some models need specific classification given their very specific properties. Bicho et al. (2012), for example, proposed an algorithm for simulating crowds based on the modeling of leaf venation patterns and the branching architecture of trees. This model uses the concept of space colonization to model crowd behavior.

In some cases, animating crowds in huge and complex environments for long periods can be very challenging and difficult with the traditional crowd simulators. This need of handling dense populations in large-scale environments gave rise the models based on crowd patches. A crowd patch is a block containing precomputed local crowd simulation. Several patches can be designed with different animations. Then, for animating a crowd, it is necessary just to connect patches in space and time (Figure 27). Crowd patches provide endless animation in real-time, nevertheless interactivity is not allowed since the agents cannot adapt their precomputed paths (LEE et al., 2006; YERSIN et al., 2009; JORDAO et al., 2014).

Figure 27 – Crowd patches used for crowd simulation.



Source: Yersin et al. (2009).

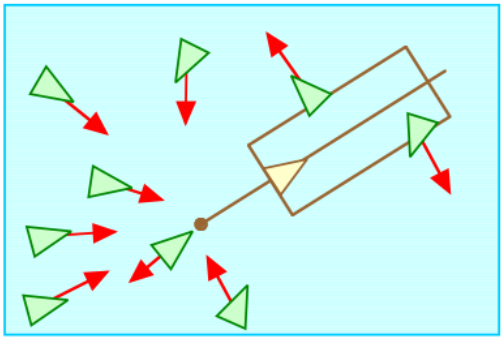
Finally, in addition to those techniques presented here, there are also several commercial solutions, such as: Massive (2015), Legion (2015), Golaem (2015), just to cite a few. Popular tools used for 3D modeling and animation, such as Autodesk 3ds Max (2015a), Autodesk Maya (2015b) and Blender (2015), have either plugins for simulating crowds or inherent support. Generally, these solutions present limited crowd behavior, but exhibit visually pleasant animations.

2.1.4 Social behavior

Several techniques have been proposed over the years focusing on navigation and collision avoidance in crowds represented as conglomerations of agents with global goals. However, in real crowds there are several social interactions, since people interact with the environment and with each other. While most of the crowd behavior studies consider only interactions among isolated individuals, a recent study demonstrated that up to 70% of the pedestrians observed in crowds walk in groups (MOUSSAÏD et al., 2010). In this case, groups in the sociological sense, i.e., not only referring to the proximity of individuals, but the individuals with social relationships intentionally walking together, such as friends or members of the same family.

Reynolds (1987) in his pioneering work simulated a flock of boids, which is a group formation, using a set of rules (alignment, cohesion and separation). Musse and Thalmann (1997) used sociological concepts to describe a rule-based model for simulating the relationship of groups in a crowd. An agent interacts with a group according to its emotional status, the level of relationship with the group, and its level of dominance. With these characteristics and the rules established by the model, the authors could simulate some sociological effects in crowds, such as grouping, polarization and adding. In 1999, Reynolds introduced more rules for steering agents and, through the combination of some of them, he described the leader following behavior (Figure 28). In (QIU; HU, 2010), the authors simulated intra-group and inter-group relationships in a model based on (REYNOLDS, 1999). In this model, two-dimensional matrices were used to establish the relationships among the agents in the same group and the relationships among groups. Recently, Lemercier and Auberlet (2015) presented behavioral rules based on perception in which agents analyze the situation to adopt different behaviors accordingly (following and group collision avoidance behavior).

Figure 28 – Leader following behavior.

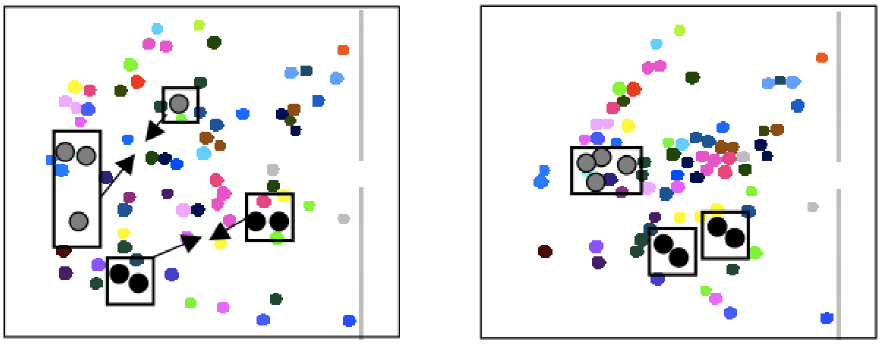


Source: Reynolds (1987).

In (BRAUN et al., 2003), the authors added an altruism force to the particle-based model proposed by Helbing et al. (2000). This force is used to keep the agents belonging to the same family together. The experiments performed with this model showed that the addition of the altruism force made altruist agents tend to rescue dependent agents (Figure 29). Xu et al. (2010) developed a particle-based model to simulate groups of two agents. Then, they studied the impact of these bonding effects on crowd behavior.

In (MOUSSAÏD et al., 2010), the authors, based on observations, drew some conclusions related to the group formation in real crowds. First, they observed that up to 70% of the people in crowds walk in groups. Moreover, groups with two to four people are more frequent, whereas groups with five or more people are rare. Another observed aspect was the relation between the agents' speed and the group size, where the first decreases when the second increases. Finally, it was observed that, in low density, the group members

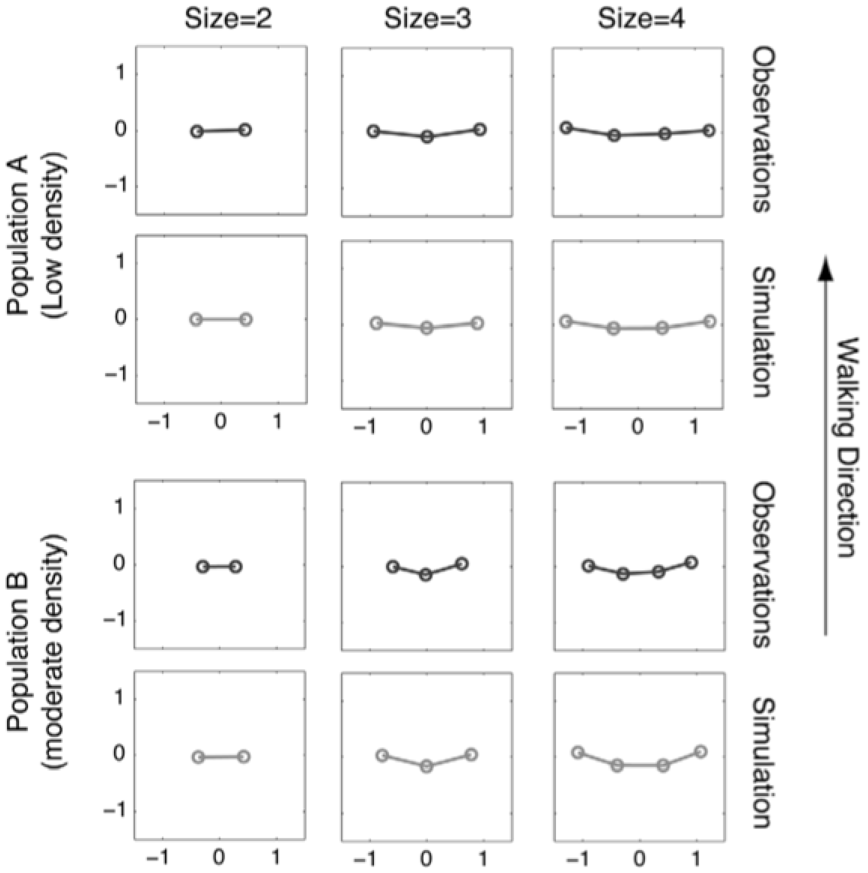
Figure 29 – Altruism force being used to keep groups together.



Source: Braun et al. (2003).

tend to walk side by side, forming a line perpendicular to the motion direction; whereas, in high density, the members tend to form a V-like pattern (Figure 30). The authors demonstrated, by adding a group force to a particle-based model, that the V-like pattern facilitates social interactions within the group, but reduces the flow because of its "non-aerodynamic" shape. The authors conclude that: "crowd dynamics is not only determined by physical constraints induced by other pedestrians and the environment, but also significantly by communicative, social interactions among individuals".

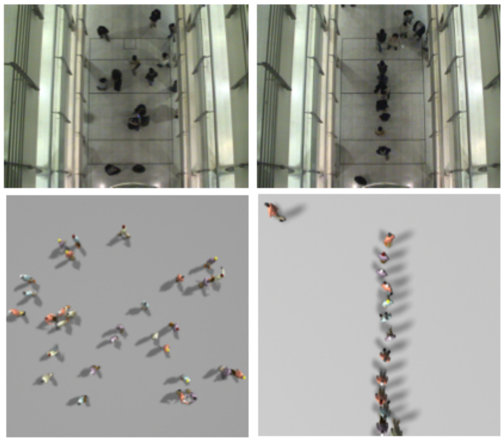
Figure 30 – Average patterns of organization of groups with two to four individuals.



Source: Moussaïd et al. (2010).

Social behavior can also be extracted from real crowd footage, through computer vision techniques, and then be used as input for data-driven models. In (LEE et al., 2007), the authors presented a model able to simulate group behavior, among others, from behavior patterns learned from video footage (Figure 31). Lerner et al. (2007) also simulated group behavior using a data-driven model. In this case, the trajectories of the agents are synthesized incrementally by considering its spatio-temporal relationships with other nearby agents and obstacles, and searching for similar scenarios in the database.

Figure 31 – Social behavior simulated by a data-driven model.



Source: Lee et al. (2007).

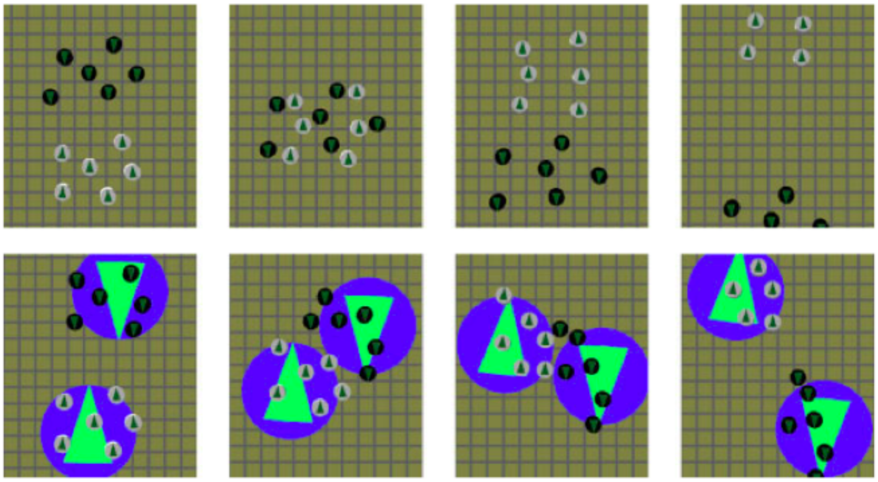
In 2010, Schuerman et al. proposed a new approach which reduces the complexity of standard agents by externalizing the specialized steering logic required for some situations to a new class of agents called “situation agents”. According to the authors: “These agents exist solely to influence the agents within their vicinity by modifying their preferred velocities and boldness parameters”. They demonstrated that this type of agent can be used for the maintenance of regular groups and groups in formation (Figure 32).

Karamouzas and Overmars (2010, 2012) proposed a model to simulate the local behavior of small groups based on Moussaïd’s observations (MOUSSAÏD et al., 2010). In this model, the agents’ current position in a group is interpolated according to the most adequate formation at the moment (among the possible formations) (Figure 33). After the group maintenance, the final motion of each agent is computed by an agent-based model.

Ricks and Egbert (2012) proposed a model which uses transactional analysis to simulate social interactions between agents in a crowd. The model focuses on the evolving social needs of agents and allows the agents to join and to leave different groups (of two) as desired. The model allows simulating two types of social interactions: agents that stop to talk and agents that walk together for some time.

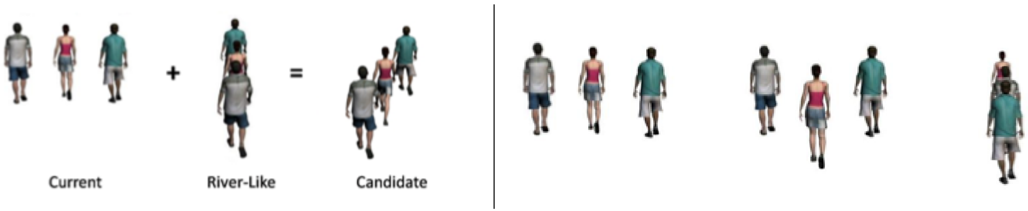
In (LEMERCIER et al., 2012), the authors made an experimental study about following behavior in crowds (Figure 34 left) and, with the obtained results, developed a numerical model to simulate the observed interactions. In the developed model, the acceleration of an agent is controlled as a function of relative speed and distance to the target. The model is able to simulate following behavior regardless the crowd density and to reproduce the experimentally observed stop-and-go waves when calibrated correctly. Bruneau et al. (2014b, 2014a) went further on simulating following behaviors by taking into account physical, social and psychological aspects to adjust the following distance. In their model, the follower evaluates the leader’s

Figure 32 – Top left: two groups of six agents (marked in brown and white, respectively) are placed in front of each other. The groups have opposing trajectories and must thus cross each other. The top sequence shows the agents' behavior simulated by a velocity-based model. The bottom sequence shows the interaction between the same groups of agents, with the difference that this time each of the groups has been assigned a situation agent (large and marked in blue).



Source: Schuerman et al. (2010).

Figure 33 – Simulation of small groups. Example of interpolation between a current and a river-like formation (left) and the group formations (from left to right): line-abreast, V-like, river-like (right).



Source: Karamouzas and Overmars (2010).

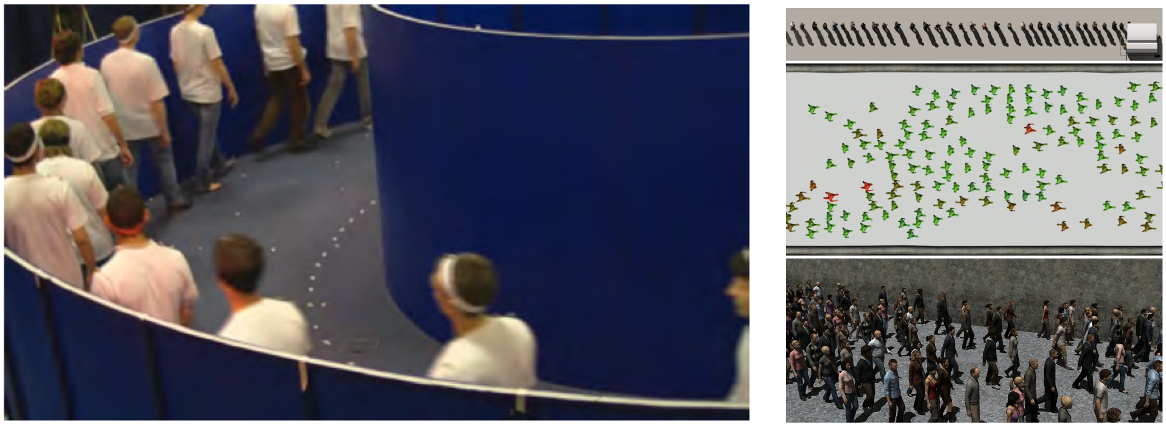
motion, predicts the leader's next move, and then, based on this prediction, the agent adjusts its speed to keep a safe distance from the leader (Figure 34 right).

Rio et al. (2014) presented and discussed several models for human following behavior simulation. They also discussed the importance of the visual information on following behavior and introduced several hypothesis about the information used for visual control. Finally, in (WU et al., 2013), the authors combined Ondřej's model based on synthetic vision (ONDŘEJ et al., 2010) with the formation characteristics and deformation mode of small groups presented in (KARAMOUZAS; OVERMARS, 2012) to reproduce local group behavior.

2.1.5 Evaluation and validation

An important aspect for crowd simulation is the realism of the results, but *is it possible to evaluate and to validate virtual crowd behavior*? This is a pertinent question in this area, since the term realism is subjective. What looks realistic for a person may not seem so realistic for another. Some researchers have devoted attention to this question given the need of comparing the results obtained through simulators with reality.

Figure 34 – Experimental study on following behavior (left) and simulated following behavior on 1-D and 2-D scenarios (right).



Source: Lemercier et al. (2012) (left) and Bruneau et al. (2014b) (right).

Crowd behavior can be evaluated quantitatively or qualitatively. Quantitative metrics are interested in comparing quantities extracted from crowd formations. Examples of these quantities are: how fast agents can evacuate a room, the mean of distances between agents, the agents' average velocity, the number of collisions, etc. Qualitative metrics, on the other hand, are intended to compare visual aspects of the crowds. Usually, a qualitative evaluation consists in checking whether the results of the simulation can mimic patterns observed in real crowds. A key aspect to evaluate virtual crowd behavior is to have real data for comparison and validation.

In (KAPADIA et al., 2009a), the authors proposed a framework for detecting anomalies on steering behavior through predefined rules or user defined sketches. Singh et al. (2009) presented a benchmark framework, composed of metrics of evaluation and a scoring method, for objectively evaluating steering behaviors, so as to compare different steering algorithms. In (2011), Kapadia et al. introduced a method of automatically generating and sampling the representative space of challenging scenarios and, in addition, they propose a method of determining coverage and quality of a steering algorithm in this space.

Musse et al. (2012) proposed a model to quantitatively compare global flow characteristics of two crowds (real or synthetic). The approach compares distances in 4-D histograms which take into account the agents' velocities (speed and orientation). In (GUY et al., 2012), the authors introduced the *Entropy Metric* to evaluate the predictability of crowd simulation techniques in terms of similarity to real-world crowd data. Charalambous et al. (2014) presented a framework for visual crowd analysis. The proposed framework can detect potentially erroneous behaviors in a simulation given a collection of arbitrary, user-selected evaluation metrics.

Recently, frameworks for parameter estimation and comparative evaluation have been proposed. Berseth et al. (2014) introduced a framework which searches for optimal parameters for a given model according to pre- or user-defined metrics. The model's parameters can be evaluated and optimized according to metrics such as the coverage of the algorithm, the distance quality, time quality, pedestrian least-effort quality, computational efficiency, similarity to ground truth, among others. In (2014), Wolinski et al. presented a framework which has the objective of estimating the parameters of crowd simulators by fitting the simulated steering behavior to real situations. This estimation is modeled as an optimization problem. The framework also supports a variety of metrics to compare reference data with simulation outputs.

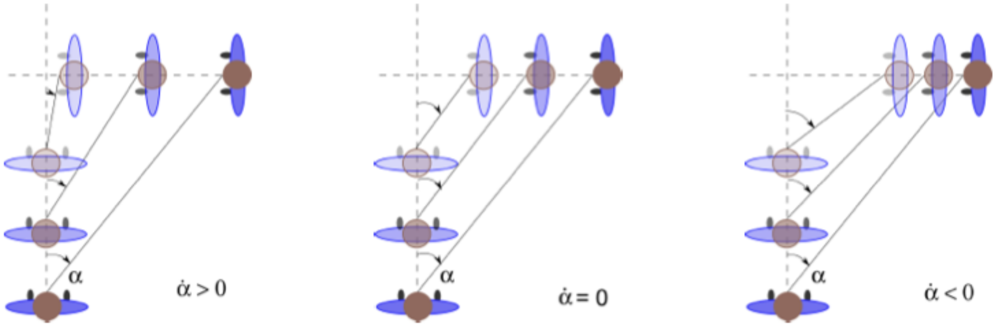
2.2 A closer look to Ondřej's model

The category of crowd simulators based on synthetic vision has an important representative that was introduced in (ONDŘEJ et al., 2010), as presented in Section 2.1.2.7. In this thesis, it is proposed a novel model that fits in this category. For this reason, it is very important to make a detailed presentation of Ondřej's model. This will allow not only a better understanding of Ondřej's model, but also to clearly identify how the model proposed in this work overcomes some of its drawbacks.

The use of synthetic vision is motivated by the possibility of mimicking the human perception as it is. During locomotion, humans are always resorting to their senses to adapt the steering. It is known by the literature that the visual sense has a strong role on locomotion (CUTTING et al., 1995; WARREN; FAJEN, 2004; RIO et al., 2014).

Cutting's work (1995) on human locomotion in the field of cognitive science stated that humans are successively answering two questions during interactions with moving and static obstacles: *Will a collision occur?* and *When will a collision occur?* Through experiments, the authors could observe that these two questions are answered by extracting two indicators from the perceived optic flow. The first indicator is the $\dot{\alpha}$ which represents the time-derivative of the bearing angle (α) under which obstacles are perceived (Figure 35); and the second indicator is the time-to-collision, or *ttc*, which is deduced from the rate of growth of obstacles in successively perceived images.

Figure 35 – "The bearing angle and its time-derivative, respectively α and $\dot{\alpha}$, allow detecting future collisions. From the perspective of an observer (the walker at the bottom), a collision is predicted when α remains constant in time. **(left)** $\alpha < 0$ and $\dot{\alpha} > 0$: the two walkers will not collide and the observer will give way. **(center)** the bearing angle is constant ($\dot{\alpha} = 0$). The two walkers will collide. **(right)** $\alpha < 0$ and $\dot{\alpha} < 0$: the two walkers will not collide and the observer will pass first".

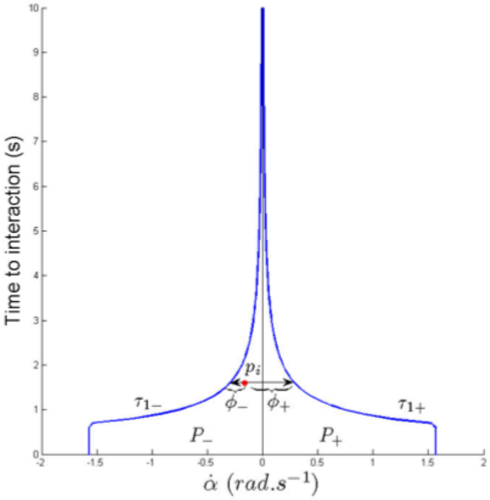


Source: Ondrej et al. (2010).

Still according to Cutting, the relevant information necessary to achieve collision-free locomotion is entirely described by the pair ($\dot{\alpha}$, *ttc*), where a collision is predicted when $\dot{\alpha}$ is close to zero and it is imminent if *ttc* is low. Based on these observations, Ondřej et al. developed a model for simulating crowds where the agents react to their visual stimuli to avoid collisions with other agents and the environment. In their work, the only difference to the Cutting's statements is the use of the time-to-interaction concept, *tti*, instead of *ttc*, for convenience, where *tti* means the remaining time until a collision and *ttc* was referred to the increasing rate of the obstacles' size perceived through the vision.

In their model, at each time-step and for each agent, the visual perception is represented as a set of pixels representing the static and moving obstacles from the agent's point of view. For each of those pixels a pair ($\dot{\alpha}$, tti) is computed. Then, the agent's reaction to visual stimuli is twofold. First, for each pixel indicating a future collision ($\dot{\alpha}$ close to zero), a turning is computed based on a threshold function (Figure 36). The turning which avoids all the obstacles and deviates less from the goal is the chosen one. And second, when the collision is imminent (i.e., there is at least a $tti < 3s$), the speed is adjusted according to the minimum tti among all $tti < 3s$, so as to slow down the agent.

Figure 36 – Threshold function. Future collision is detected when the pair ($\dot{\alpha}$, tti) is below the function and ttii > 0. The plot also illustrates that the lower the tti value, the higher the agent's reaction.



Source: Ondrej et al. (2010).

Despite the good results reported, Ondřej's model suffers from important drawbacks. These are mostly consequence of a basic and unique response to the visual stimuli. Such features drastically limit the range of possible solutions explored to solve risks of future collisions. For instance, the possibility of accelerating to avoid an obstacle is not considered. Furthermore, collisions are often observed, since the model is purely reactive. The agents only focus on avoiding the most imminent danger (most of the information generated by the optical flow is discarded), neglecting the consequence of their action on other potential dangers. Finally, although their model yields emergent patterns under specific traffic conditions (as expected), those patterns often seem too strong and unnatural, probably because the agents tend to act similarly.

2.3 Final considerations

This chapter started with an overview of global and local path planning techniques. Their characteristics and differences were briefly explained and discussed. Next, it was discussed the importance of going further on planning trajectories by incorporating social behavior. Then, the important, and still underexplored, topic of evaluation and validation was shortly overviewed. Finally, Ondřej's model for simulating crowds with synthetic vision was detailed for a better understanding.

In this thesis, it is presented an innovative model based on synthetic vision, which overcomes most of the limitations presented by Ondřej's model. Instead of systematically using the same rules to react to danger of collision as in (ONDŘEJ et al., 2010), the proposed model performs locally optimal adaptations with respect to a cost function. This cost function is one of the core contributions of this work and is based on the perceptual variables of the agent's synthetic vision system. It considers both the risk of future collisions as well as the relative heading toward the goal. Agents follow the gradient of the function to perform collision-free locomotion up to their goal. Since it accounts for *all* visible obstacles and not only dangerous ones, the resulting adaptation also accounts for the consequences of the motion change. This is the key contribution of this work with respect to previous synthetic vision techniques.

In other words, the full set of possible adaptations is considered by the proposed method instead of applying predefined reactions to future collisions. For this reason, in some sense the presented contributions represent the same advance with respect to Ondřej's model as the contribution of the first complete velocity-based models (PARIS et al., 2007; BERG et al., 2008a) represented with respect to the early work of Reynolds (REYNOLDS, 1999). In the next chapter, the proposed model is presented in details.

3 Gradient-based model

The objective of this work is to develop a new model for steering crowds based on the evaluation of a cost function in real-time. The cost function is used to evaluate the agent's current situation. This situation, in its turn, is composed by the obstacles perceived by the agent as well as by its properties, such as its orientation regarding its goal position and its current speed.

The first question explored by this work is: "Is it possible to define such cost function given an arbitrary situation?". As it is shown in the course of this chapter, this question is positively answered as soon as the cost function is proposed in Section 3.4. After verifying that it was possible to define a cost function for evaluating the agent's current situation, a second question was raised "How could this function be used to adapt the agent's motion?". The answer for this question is presented in Section 3.5. In short, the agent's motion adaption is modeled as a minimization problem, i.e., at each simulation's time-step each agent will try to minimize the cost function which evaluates its current situation.

The cost function defined in this chapter is written as a function of the agent's velocity (vector) decomposed in speed (scalar) and orientation. In other words, it is defined a two-dimensional function. The proposed model is described as a *gradient-based* one given the need of working with the gradient of the function for minimizing it.

This chapter is organized into six sections. It starts by introducing the new variables used for vision-based collision avoidance. These new variables satisfy the requirements for vision-based locomotion stated by Cutting's work (1995). Next, it is made an overview of the model in Section 3.2 where the simulation's control loop is introduced as well as the mathematical characterization of the agent's state. This section is followed by three sections which detail each phase of the simulation's control loop:

***Perception* (Section 3.3)** describes the agent's visual perception;

***Evaluation* (Section 3.4)** introduces the cost function and describes how it evaluates the perceived information; and

***Action* (Section 3.5)** describes how the agent's motion is adapted according to the cost function's gradient.

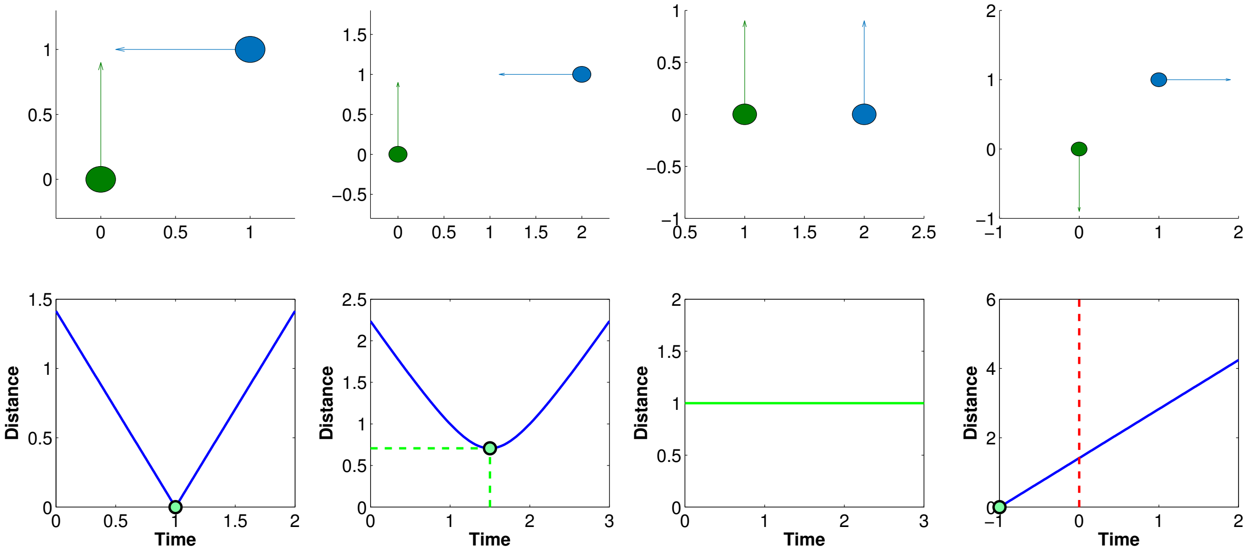
Final considerations are draw in the last section.

3.1 Variables for vision-based collision avoidance

It is acknowledged by the literature, as it was shown in Chapter 2, that the visual perception has an important role on locomotion. Moreover, Cutting's work (1995) states that humans, when walking, are successively answering two questions: "*Will a collision occur?*" and "*When will a collision occur?*" And those questions are answered resorting to two indicators in the optic flow ($\dot{\alpha}$ and ttc, respectively).

For the proposed model, two quantities are defined to answer those questions and to act analogously to the pair ($\dot{\alpha}$, ttc). Considering two agents moving at constant velocity, the four situations depicted in Figure 37 can be observed. If they are converging, the distance between them will decrease and at some point, they will reach the minimum distance between them (figures 37 (a) and (b)). Otherwise, if they are diverging, the distance will only increase (Figure 37 (d)). And, finally, if their motion is parallel, that distance will remain constant (Figure 37 (c)). In the situation of convergence, the minimum reachable distance gives a hint of a possible collision. In this work, this minimum distance is referred to as distance at closest approach (or *dca*). A potential collision is predicted if *dca* is positive and smaller enough to make the agents' body envelopes intersect. The quantity *dca* answers the first Cutting's question. Similarly to Ondřej's work (2010), a collision remaining time is computed to answer the second question. In this case, the remaining time to reach the minimum distance, the so-called time to closest approach (or *ttca*) is computed. The terms *ttca* used here and *tti* used by Ondřej might sound the same, but there is a difference between them. While both quantities indicates a time to collision, Ondřej's *tti* is just an approximation, whereas *ttca* represents the exact remaining time to collision. The pair (*dca*, *ttca*) is used throughout this thesis to describe interactions between agents and static and moving obstacles.

Figure 37 – These figures show four distinct situations with two agents. The top line illustrates the agents' positions and velocities and the bottom line illustrates the distance between them over time, if they maintain a constant velocity. Still in the bottom line, the green point represents the pair (*dca*, *ttca*). (a) and (b) represent situations where the agents are converging. In (a), the minimum distance between the agents (*dca*) is 0 at time 1, which represents a risk of collision, and, in this case, the *ttca* is equal to 1 for the initial situation at the top. Whereas in (b), the pair (*dca*, *ttca*) does not represent a risk of collision, since *dca* is greater than the sum of the agents' body radius. In (c), since the agents are moving in parallel, *dca* is constant (illustrated as the green line) and *ttca* is undetermined. The situation in (d) shows the agents with a divergent motion, in this case *ttca* is negative, i.e., *dca* lies behind the agents.



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

Source: the author.

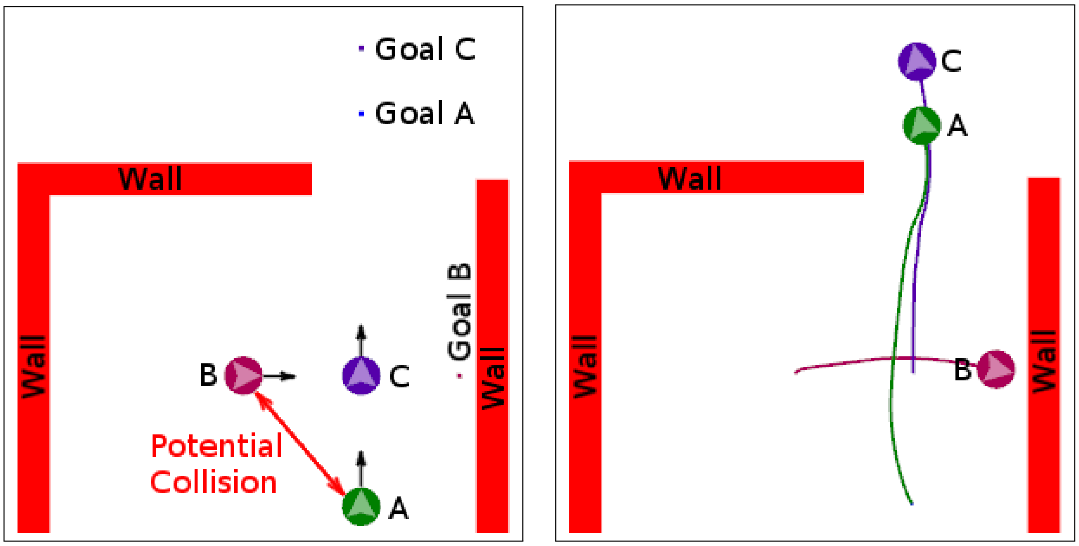
3.2 Overview

The proposed approach is an agent-based approach. Agents are equipped with a synthetic vision system, that allows them to perceive their environment, which consists of static and moving obstacles. This section first provides an overview of the simulation loop controlling the agents' motion according to their visual perception by means of a simulation example (Section 3.2.1). Then, the related mathematical formalism is introduced in Section 3.2.2.

3.2.1 Control loop

The first contribution of this thesis is a new control loop to steer agents in crowded environments with static and moving obstacles. Before introducing and examining the control loop, let us take a look at an example of a simulation's result, which is depicted in Figure 38. On the left of Figure 38, it is shown an environment and the initial states of three agents that should move towards their goals. The current motion of agents A and B presents a risk of collision in the future. By resorting to the proposed model, agent A changes its linear trajectory and adapts its velocity in order to pass behind agent B (Figure 38 (right)).

Figure 38 – Example scenario. The **left** image shows the obstacles of the environment, the agents (A, B and C) at their initial positions as well as their goals and velocities. Their current velocities raise a risk of collision between agents A and B. The **right** image shows the trace of the collision-free trajectories resulting from our vision-based control loop.

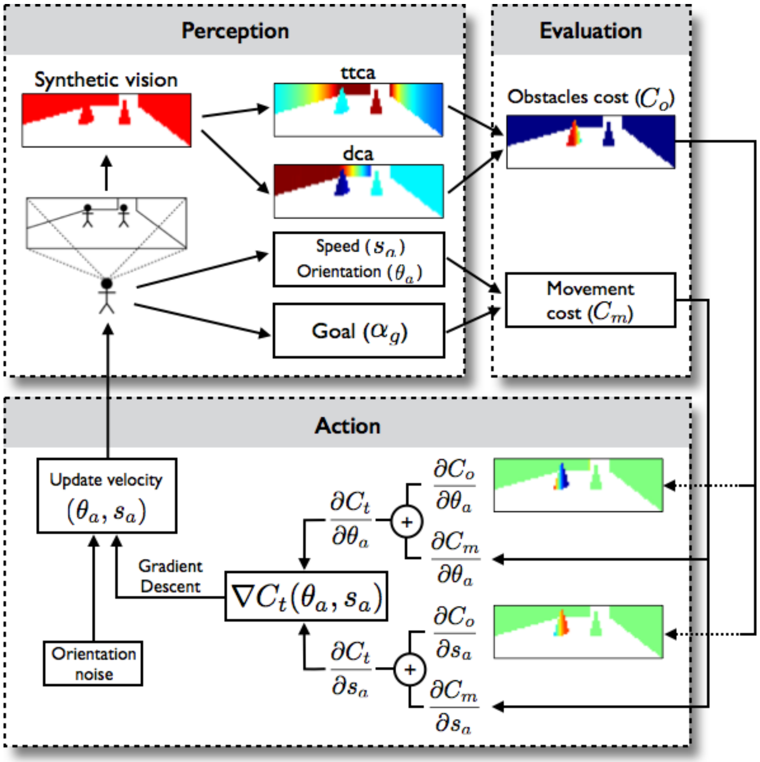


Source: the author.

This proposal has some resemblances to Kapadia's model (KAPADIA et al., 2009b) given that both approaches resort to a control loop for steering agents. In the latter case, the control is composed of a sensory, affordance and selection phases. This could be associated to the perception, evaluation, action phases introduced in this chapter. However, the resemblances end here. Kapadia's perception phase is based on egocentric circular discrete fields with a limited radius, in which agents must have a local representation of the environment in the form of several scalar fields. Their model plans a local trajectory according to the best affordance in the field resorting to unrealistic assumptions such as not taking into account occlusion and a perception of 360°. This is a very important contrast with the proposed model in which is intended to more realistically simulate a human-like perception.

In the proposed model, agents are steered to reach their goals and to avoid collisions according to the loop shown in Figure 39. The *control loop* is composed of three phases: *perception*, *evaluation* and *action*. The complete 3-phase loop is performed for each time-step of the simulation and for each agent. A central element of this loop is the agent's synthetic vision system: a *perceptual space* which is a set of three 2-D matrices of pixels values (perceived obstacles, *ttca* map, *dca* map). Figures 40 (a), (b) and (c) illustrate those three matrices for the perceptual space of agent A at the initial state of Figure 38 (left).

Figure 39 – The 3-phase control loop. **Perception:** the agent perceives the surrounding environment resorting to its synthetic vision. **Evaluation:** the agent assesses the cost of the current situation. **Action:** the agent takes an action to reduce the cost of the situation.


Perception
Evaluation
Synthetic vision
ttca
dca
Obstacles cost (C_o)
Speed (s_a)
Orientation (θ_a)
Movement cost (C_m)
Goal (α_g)
Action
Update velocity
(θ_a, s_a)
$\frac{\partial C_o}{\partial \theta_a}$
$\frac{\partial C_t}{\partial \theta_a}$
+
$\frac{\partial C_m}{\partial \theta_a}$
Gradient Descent
$\nabla C_t(\theta_a, s_a)$
$\frac{\partial C_o}{\partial s_a}$
$\frac{\partial C_t}{\partial s_a}$
+
$\frac{\partial C_m}{\partial s_a}$
Orientation noise

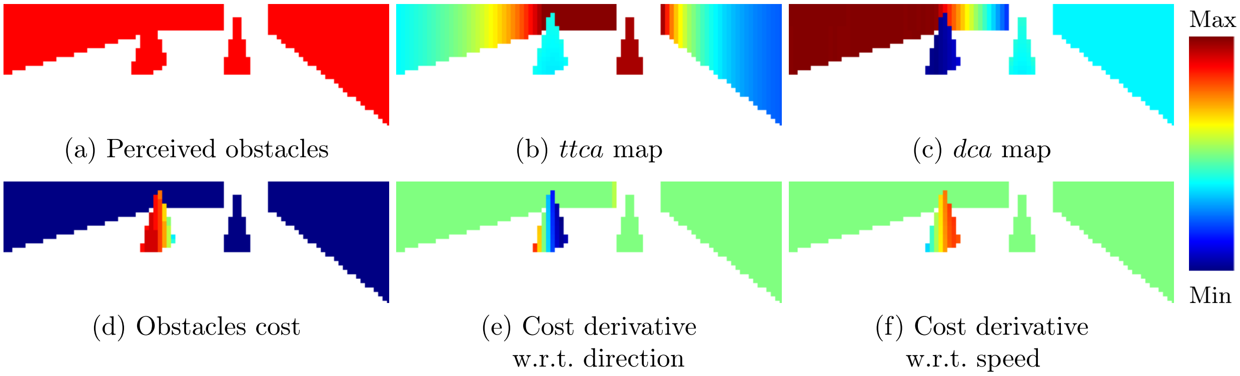
Source: the author.

The role of the ***perception*** phase is to project the visually perceptible obstacles of the environment to the perceptual space. This step is similar to performing a graphical rendering of the scene from the agent's point of view. However, instead of rendering the texture and color of obstacles, pixels capture some geometrical information: the time to closest approach (*ttca*) and the distance at closest approach (*dca*). The perception phase is detailed in Section 3.3. Figure 40 (a) illustrates what agent A perceives of the surrounding obstacles and agents, and figures 40 (b) and (c) show the *ttca* and *dca* maps.

The role of the ***evaluation*** phase is to estimate how 'good' the agent's current velocity is, given the risk of collision with the perceived obstacles and the alignment with the goal. That evaluation is made through the definition of a cost function C_t, which consists of two components: the obstacles cost C_o, which captures the risk of collision based on the perceived matrices of *ttca* and *dca*; and the movement cost C_m, which considers the task of reaching a goal based on the agent's speed s_a, orientation θ_a and angle with the goal α_g. The definition of this cost function is another important contribution of this work. The evaluation phase is detailed in Section 3.4. Figure 40 (d) illustrates the C_o associated with each pixel. The risk of collision with agent B is well estimated as the corresponding pixels have a high cost.

The role of the ***action*** phase is to update the agent's velocity to minimize the cost function C_t. To this end, the partial derivatives of C_t with respect to the agent's motion variables are computed and the locally

Figure 40 – Images representing the vision of the agent A in Figure 38 (left). (a) Obstacles detected by agent A for the situation. (b) and (c) show the time to closest approach (*ttca*) and distance at closest approach (*dca*) for each perceived obstacle (blue encodes a low value, while red corresponds to a high value). Obstacles with low *ttca* and *dca* convey a significant risk of collision, which leads to high cost in (d). To solve the problem, the agent determines the partial derivatives of the obstacles cost with respect to direction and speed ((e) and (f), respectively). Collision is avoided by descending the gradient so as to reduce the cost. The resulting trajectories are shown in Figure 38 (right).



(a) Perceived obstacles

(b) *ttca* map

(c) *dca* map

(d) Obstacles cost

(e) Cost derivative w.r.t. direction

(f) Cost derivative w.r.t. speed

Source: the author.

optimal move is deduced. The action phase is detailed in Section 3.5. Figures 40 (e) and (f) illustrate the partial derivatives of C_o for each pixel.

The agent's change of orientation and speed is determined through the collective information of the gradient of the cost functions computed at each pixel. Thus, the values illustrated in figures 40 (e) and (f) induced agent A to perform a left turn and to reduce its speed in order to avoid collision with agent B. A small noise ε is added to the agent's new direction θ_a so as to disrupt symmetry as shown in Figure 39.

3.2.2 Mathematical characterization of the agent's state

The current state of an agent a is defined by its position $\mathbf{p}_a$, orientation θ_a and speed s_a which are used to compute its velocity vector $\mathbf{v}_a$ given by:

$$\mathbf{v}_a = (v_{xa}, v_{ya}) = (s_a \cos\theta_a,\, s_a \sin\theta_a) = s_a \hat{\mathbf{v}}_a \ . \qquad (3.1)$$

These quantities and others which are used throughout the thesis are summarized in Table 1 and illustrated in Figure 41. For the sake of simplicity, the parameters of the terms are omitted, for example, $\mathbf{v}_a(s_a, \theta_a)$ would be more adequate to refer to $\mathbf{v}_a$ as well as $dca_{o_i,a}(\mathbf{p}_{o_i,a}, \mathbf{v}_{o_i,a})$ and $ttca_{o_i,a}(\mathbf{p}_{o_i,a}, \mathbf{v}_{o_i,a})$ would be more adequate to refer to $dca_{o_i,a}$ and $ttca_{o_i,a}$. Other simplifications can be found in the text and they are appropriately referred when needed. A more detailed description of these quantities is given in the following sections.

3.3 Perception: Acquiring information

The *perception* phase (see top left of Figure 39) consists in gathering, for each agent, information about the surrounding environment. This is achieved by performing a graphical rendering of the scene from the agent's point of view. However, instead of rendering the visual aspect of the obstacles (e.g., color, texture, lighting),

Table 1 – Notation description. Bold face notation represents vectors, otherwise the notation represents scalar variables.

Symbol	Description
t_0	Current time-step.
$\mathbf{P}_a(t)$	Position of agent a at time t (2D point)
$\mathbf{p}_a$	Current position of agent a ($\mathbf{p}_a = \mathbf{P}_a(t_0)$)
$\mathbf{v}_a$	Velocity of agent a (2D vector)
s_a	Speed of agent a ($s_a = \|\mathbf{v}_a\|$)
$s_{a_{comf}}$	Comfort speed of agent a
θ_a	Orientation of agent a (angle measured with the x-axis)
α_g	Bearing angle with respect to the goal
$\mathbf{P}_{o_i}(t)$	Position of obstacle o_i at time t (2D point)
$\mathbf{p}_{o_i}$	Current position of obstacle o_i ($\mathbf{p}_{o_i} = \mathbf{P}_{o_i}(t_0)$)
$\mathbf{v}_{o_i}$	Velocity of obstacle o_i (2D vector)
$\mathbf{v}_{o_i\|a}$	Velocity of obstacle o_i relative to agent a (2D vector)
$\mathbf{p}_{o_i\|a}$	Current position of obstacle o_i relative to agent a
$ttca_{o_i,a}$	Time to closest approach between agent a and obstacle o_i
$dca_{o_i,a}$	Distance between agent a and obstacle o_i at the closest approach

Source: the author.

some kinematic properties are associated with each pixel. The agent's *synthetic vision* is used to identify the visible obstacles. The set of pixels through which an obstacle is detected (marked in red in Figure 40 (a)) composes the perceptual space $\mathcal{O}$. In the perceptual space, each pixel is treated as an independent obstacle, i.e., obstacles are abstracted as a set of pixels $\mathcal{O} = \{o_i\}$.

For each obstacle o_i perceived by a given agent, the relative position $\mathbf{p}_{o_i|a}$ and relative velocity $\mathbf{v}_{o_i|a}$ are computed as follows (see Table 1 for notations):

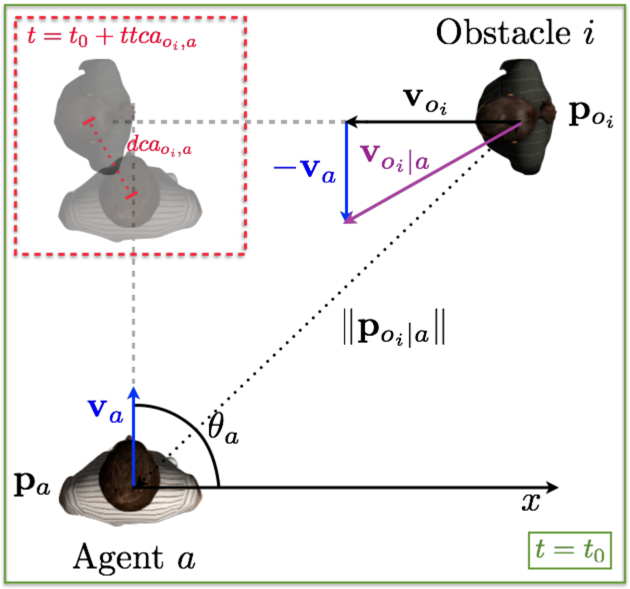
$$\mathbf{p}_{o_i|a} = \mathbf{p}_{o_i} - \mathbf{p}_a \ , \tag{3.2}$$

$$\mathbf{v}_{o_i|a} = \mathbf{v}_{o_i} - \mathbf{v}_a \ . \tag{3.3}$$

Note that the velocity $\mathbf{v}_a$ is not extracted from the analysis of the visual flow. Instead, the velocity of visible 3-D obstacles is directly retrieved from the simulation state. To compute the velocity of the obstacle's part which corresponds to the pixel, GPU interpolation is used, as will be described in Section 4.1. This approach allows treating objects of arbitrary shapes where different parts of the object move with different velocities.

$\mathbf{p}_{o_i|a}$ and $\mathbf{v}_{o_i|a}$ allow us to deduce $ttca_{o_i,a}$ and $dca_{o_i,a}$ (figures 40 (b) and (c), respectively). Assuming that the relative motion of a pixel is constant, $ttca_{o_i,a}$ quantifies the remaining time before agent a reaches the minimum distance to the obstacle o_i; and $dca_{o_i,a}$ is the distance between the agent and the obstacle at

Figure 41 – Illustration of the variables used to model interactions.



Source: the author.

the time of the closest approach.

Knowing that the squared distance, D^2, between agent a and obstacle o_i at time t is given by:

$$D^2(t) = \|\mathbf{p}_{o_i|a} + t\ \mathbf{v}_{o_i|a}\|^2 \ , \tag{3.4}$$

the time to closest approach ($ttca_{o_i,a}$) between agent a and obstacle o_i is given by the following equation:

$$\frac{d}{dt} D^2(ttca_{o_i,a}) = 0 \ , \tag{3.5}$$

where

$$\frac{d}{dt} D^2(ttca_{o_i,a}) = 2\left(\mathbf{p}_{o_i|a} + ttca_{o_i,a}\ \mathbf{v}_{o_i|a}\right) \cdot \mathbf{v}_{o_i|a} \ . \tag{3.6}$$

By solving Eq. (3.5), $ttca_{o_i,a}$ is given by:

$$ttca_{o_i,a} = \begin{cases} t \in \mathbb{R} & : \mathbf{v}_{o_i|a} = (0,0) \\ -\dfrac{\mathbf{p}_{o_i|a} \cdot \mathbf{v}_{o_i|a}}{\|\mathbf{v}_{o_i|a}\|^2} & : \mathbf{v}_{o_i|a} \neq (0,0) \end{cases} \ , \tag{3.7}$$

where t is any time value belonging to $\mathbb{R}$. Once the value of $ttca_{o_i,a}$ is known, the $dca_{o_i,a}$ can be easily computed:

$$\begin{align} dca_{o_i,a} &= \|\mathbf{dca}_{o_i,a}\| \tag{3.8} \\ &= \sqrt{D^2(ttca_{o_i,a})} \tag{3.9} \\ &= \|\mathbf{p}_{o_i|a} + ttca_{o_i,a}\mathbf{v}_{o_i|a}\| \ , \tag{3.10} \end{align}$$

given that $\mathbf{x}^2 = \mathbf{x} \cdot \mathbf{x} = \|\mathbf{x}\|^2$.

In Figure 40, the maps are encoded using a color code in which blue represents a value close to zero and red represents a large value. Because we assume a linear motion and the agent is oriented parallel to the right wall (see Figure 38 left), the *dca* of the agent with respect to the right wall is constant. The corresponding *ttca*, on the other hand, increases with the distance from a point on the wall to the agent.

The *ttca* and *dca* maps, along with the quantities defining the agents' motion (i.e., the speed, s_a; the orientation, θ_a; and the angle with respect to the goal, α_g) allow a full characterization of the current situation. This information is thus passed to the *evaluation* phase, which is described in detail in the next section.

3.4 Evaluation: A cost function to evaluate risk of collision

The goal of the *evaluation* phase (top right of Figure 39) is to estimate the risk of collision with obstacles while maintaining the agent heading toward the goal. To this end, for each agent a, we define a cost function C_t composed of two terms:

$$C_t = C_m + C_o \ , \tag{3.11}$$

where the movement cost C_m accounts for whether or not the agent is heading towards the goal; and the obstacles cost C_o evaluates the importance of risk of collision with obstacles. In the following sections, we give the details regarding C_m and C_o. Here, once again, the parameters of the functions were omitted for the sake of simplification. What is important to know is that those functions are written in terms of s_a and θ_a, which allows us to compute their gradients with respect to these terms which compose the agent's velocity. Another simplification made was to drop the index referring to the agent, then, from here on C_t, C_m and C_o are referring to $C_{t,a}$, $C_{m,a}$ and $C_{o,a}$, respectively.

3.4.1 Movement cost

The movement cost function C_m is defined so that it is minimal when the agent is heading towards the goal at its comfort speed:

$$C_m = 1 - \frac{C_\alpha + C_s}{2} \ , \tag{3.12}$$

where

$$C_\alpha = \exp\left(-\frac{1}{2}\left(\frac{\alpha_g}{\sigma_{\alpha_g}}\right)^2\right) \ , \tag{3.13}$$

$$C_s = \exp\left(-\frac{1}{2}\left(\frac{s_a - s_{a_{comf}}}{\sigma_s}\right)^2\right) , \tag{3.14}$$

α_g and s_a are the function arguments and σ_{α_g} and σ_s are parameters used to adapt the shape of the cost function.

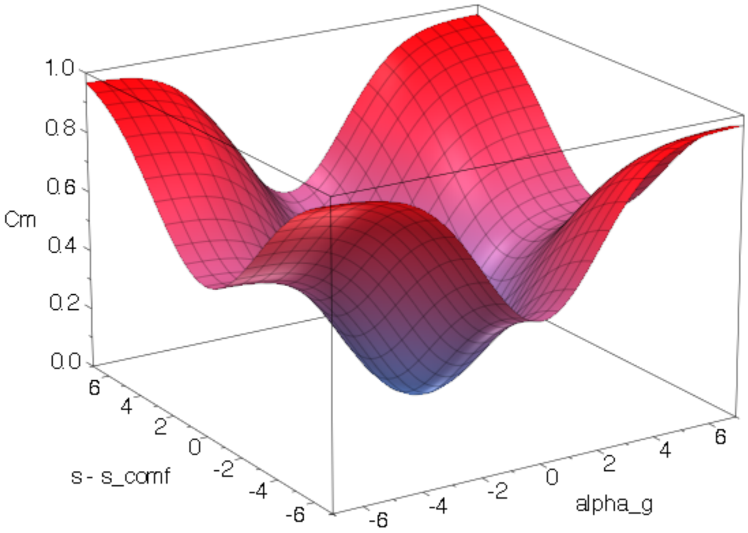
C_m is thus defined as a sum of two Gaussians functions (equations (3.12), (3.13) and (3.14)). The width of both Gaussians can be independently controlled through σ_{α_g} and σ_s. A plot of the movement cost function with $\sigma_{\alpha_g} = 2$ and $\sigma_s = 3$ is shown in Figure 42.

As it will be detailed in Section 4.2, changing these parameters will directly play on the agents' avoidance strategy, i.e., their preference to adapt their speed or their orientation to perform collision avoidance.

3.4.2 Obstacles cost

The obstacles cost $C_{o_i,a}$ accounts for the risk of collision between agent a and each perceived obstacle o_i. This cost is high when both the distance at the closest approach *dca* (which indicates an existing risk of collision)

Figure 42 – Plot of the movement cost function C_m (Equation 3.12) with $\sigma_{\alpha_g} = 2$ and $\sigma_s = 3$.



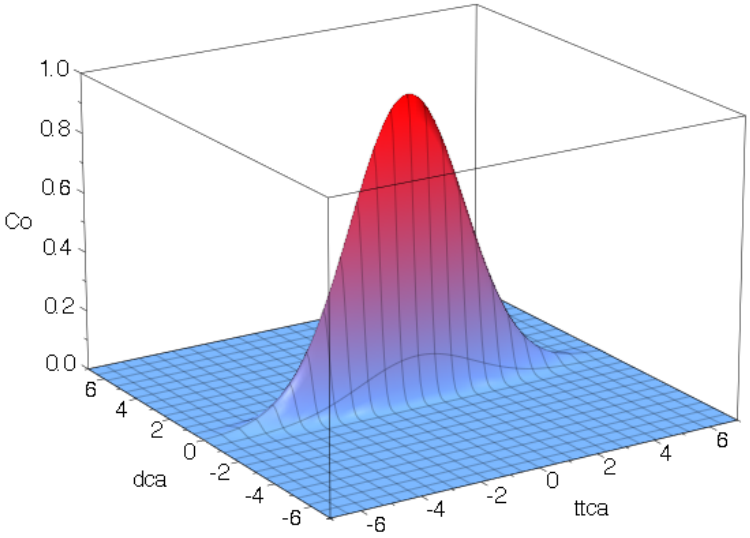
Source: the author.

and the time to closest approach *ttca* (which indicates how imminent the collision is) have low values. $C_{o_i,a}$ is defined as a two-dimensional Gaussian function:

$$C_{o_i,a} = \exp\left[-\frac{1}{2}\left(\left(\frac{ttca_{o_i,a}}{\sigma_{ttca}}\right)^2 + \left(\frac{dca_{o_i,a}}{\sigma_{dca}}\right)^2\right)\right], \tag{3.15}$$

where $ttca_{o_i,a}$ and $dca_{o_i,a}$ are the function arguments and σ_{ttca} and σ_{dca} are parameters used to adapt the shape of the function. σ_{ttca} controls avoidance anticipation time, whereas σ_{dca} controls the distance the agent must keep from obstacles. A plot of the obstacles cost function with $\sigma_{ttca} = 2$ and $\sigma_{dca} = 0.3$ is shown in Figure 43 (with these parameters, the cost is high only when *dca* is low).

Figure 43 – Plot of the obstacles cost function C_o (Equation 3.15) with $\sigma_{ttca} = 2$ and $\sigma_{dca} = 0.3$.



Source: the author.

Let us recall that the obstacle o_i corresponds to the obstacle seen by agent a through the pixel p_i (see Section 3.3). However, agent a can detect several obstacles from its visual flow (one for each pixel). The costs $C_{o_i,a}$ of each obstacle are combined by simply averaging the cost of all the visible obstacles, such that:

$$C_o = C_{o,a} = \frac{1}{n}\sum_{i=1}^{n} C_{o_i,a} \ , \tag{3.16}$$

where n is the number of visible obstacles.

Figure 40 (d) shows the obstacles cost for the situation depicted in Figure 38 (left). Note that the pixels for which the cost is high (marked in red in Figure 40 (d)) correspond to those which have a low value for both *ttca* and *dca* (i.e., marked in blue in figures 40 (b) and (c)).

3.5 Action: Gradient descent

Each agent aims at moving toward its goal while avoiding some risk of collision with obstacles. As the cost function C_t models these two criteria, agents continuously adapt their velocity in order to locally optimize the cost function C_t. Technically, this is efficiently done by computing the gradient of C_t and by updating the agent's velocity to follow the gradient (steepest descent). This operation is repeated at each time-step.

The gradient descent method uses the information provided by the gradient of a function to reach a local minimum. Given a function $F(x)$, defined and differentiable, and an initial guess x_n, the method says this function decreases fastest if one goes from x_n in the direction of the negative gradient of F at x_n, $\nabla F(n)$. Thus, it can be defined:

$$x_{n+1} = x_n - \lambda_n \nabla F(x_n),\ n \geq 0 \ , \tag{3.17}$$

where λ_n is the size of the step which can change at every iteration. The sequence should converge to a local minimum.

Assuming that an obstacle's motion is constant and given that the gradient of the cost function ∇C_t only depends on the agent's motion variables (s_a, θ_a), the agent's new motion, given by $(s_a^{(\text{new})}, \theta_a^{(\text{new})})$, is thus computed by giving a step of size λ_n such that:

$$(s_a^{(\text{new})}, \theta_a^{(\text{new})}) = (s_a, \theta_a) - \lambda_n \nabla C_t(s_a, \theta_a) + (0, \varepsilon) \ , \tag{3.18}$$

where a small noise value $\varepsilon \sim \mathcal{U}(-0.1, 0.1)$ is added to disrupt symmetric situations.

The λ_n value can have a strong impact in the optimization process, and it is thus worth discussing it before continuing. A too small λ_n might make the optimization take a long time to converge, whereas a too large one might make it diverge. Moreover, the ideal λ_n might not be a fixed value, but instead a value which is adapted according to the characteristics of the function we wish to minimize. There are ways of adapting the step size to assure convergence. The most straightforward solution (CAUCHY, 1847) is to perform a line search, choosing a step size which minimizes the cost function. Nevertheless, evaluating the proposed cost function for a set of steps is impractical, because it would require to render the agent's vision for each step size evaluated. Another known solution, proposed by Barzilai and Borwein (1988), defines an adaptive step size by approximating the secant equation. This solution requires the evaluation of the current and the previous steps. However, it is also not applicable for the proposed cost function, because the proposed function is changing every step according to the number of pixel in the agent's vision (Equation (3.16)). Therefore, evaluating the current step with the previous does not make sense. Finding the correct adaptive step size can thus be cumbersome in the context of our problem. The computational cost required for this type of solution could have a very negative impact in the model performance, while the benefits from such an approach would probably be very limited. This idea is confirmed by the experiments made which show that using a the fixed value of $\lambda_n = 1$ yields good results. Nevertheless, future research works could aim at solving the problem of efficiently finding an adaptive step size.

Then, ∇C_t is evaluated as follows:

$$\begin{aligned} \nabla C_t &= \begin{bmatrix} \dfrac{\partial C_t}{\partial s_a} & \dfrac{\partial C_t}{\partial \theta_a} \end{bmatrix} && (3.19) \\ &= \begin{bmatrix} \left(\dfrac{\partial C_m}{\partial s_a} + \dfrac{\partial C_o}{\partial s_a} \right) & \left(\dfrac{\partial C_m}{\partial \theta_a} + \dfrac{\partial C_o}{\partial \theta_a} \right) \end{bmatrix} . && (3.20) \end{aligned}$$

Note that the partial derivatives of both C_m and C_o can be explicitly evaluated as detailed in the Appendix A. The values of the partial derivatives of the obstacles cost function $C_{o_i,a}$, for the situation in Figure 38 (left), can be visualized in figures 40 (e) and (f). In those images, blue represents a negative value, green a value close to zero, and red a positive value. Since most of the obstacles in Figure 40 (e) with non-zero value have a blue color, the agent will tend to turn left. In Figure 40 (f), the obstacles have mostly a red color, causing the agent to reduce its speed.

3.6 Final considerations

In this chapter, the proposed model was presented and detailed. The new variables for vision-based collision avoidance were first introduced. Then, it was defined a control loop composed of three phases: *perception*, *evaluation* and *action*. This loop tries to mimic the real human perception/action loop. In the perception phase the agent's visual information is acquired. The following phase uses this information to evaluate the current situation of the agent, by resorting to cost functions. Finally, in the action phase, the gradient descent method is used to minimize the cost function by adapting the agent's velocity. The model represents a great advance when compared to the previous one (ONDŘEJ et al., 2010) with respect to the use of the visual information. Here, the entire agent's visual information is taken into account, and moreover, the evaluation of the gradient of the cost function provides the nearly optimal movement adaptation.

In the next chapter, the algorithm developed for the proposed model is described as well as the technical aspects are discussed.

4 Implementation and parameterization

This chapter starts by describing in details the algorithm developed for the proposed model as well as the technical aspects related to this implementation (Section 4.1). In the following section, the influence of each parameter of the model is analyzed. This analysis is followed by the description of a strategy for setting up the model's parameters in order to fit global features observed in experimental data.

4.1 Implementation

4.1.1 Gradient-based model for agents equipped with synthetic vision

Algorithm 1 shows a pseudocode of the proposed approach. It consists of two successive loops: computing C_t and its gradient $\nabla C_{t,a}$ for each agent a (lines 1 to 19); and a second loop to update the state of the agents in terms of $\nabla C_{t,a}$ (lines 20 to 22).

Algorithm 1 Gradient-based model implementation.

```
 1: for all agents $a$ do
 2:   $\mathbf{p}_a, \mathbf{v}_a \leftarrow$ get_state($a$)
 3:   $camera \leftarrow$ set_up_camera($\mathbf{p}_a, \mathbf{v}_a$)
 4:   $perc\_space \leftarrow$ render_environment()
 5:   for all pixels $p_i \in perc\_space$ do
 6:     if has_visible_obstacle($\mathrm{p_i}$) then
 7:       $o_i \leftarrow get\_obstacle(p_i)$
 8:       $\mathbf{p}_{o_i}, \mathbf{v}_{o_i} \leftarrow$ get_motion($o_i$)
 9:       $\mathbf{p}_{o_i|a}, \mathbf{v}_{o_i|a} \leftarrow$ relative_motion $(\mathbf{p}_a, \mathbf{v}_a, \mathbf{p}_{o_i}, \mathbf{v}_{o_i})$
10:       $ttca_{o_i,a} \leftarrow$ compute_ttca $(\mathbf{p}_{o_i|a}, \mathbf{v}_{o_i|a})$
11:       $dca_{o_i,a} \leftarrow$ compute_dca $(\mathbf{p}_{o_i|a}, \mathbf{v}_{o_i|a}, ttca_{o_i,a})$
12:       $\left(\frac{\partial C_{o_i,a}}{\partial s_a}, \frac{\partial C_{o_i,a}}{\partial \theta_a}\right) \leftarrow$ grad_pixel_cost $(ttca_{o_i,a}, dca_{o_i,a})$
13:       $perc\_space\,(p_i) \leftarrow \left(\frac{\partial C_{o_i,a}}{\partial s_a}, \frac{\partial C_{o_i,a}}{\partial \theta_a}\right)$
14:     end if
15:   end for
16:   $\nabla C_o = \left(\frac{\partial C_o}{\partial s_a}, \frac{\partial C_o}{\partial \theta_a}\right) \leftarrow$ grad_obstacle_cost $(perc\_space)$
17:   $\nabla C_m = \left(\frac{\partial C_m}{\partial s_a}, \frac{\partial C_m}{\partial \theta_a}\right) \leftarrow$ grad_movement_cost $(\Delta_s, \alpha_g)$
18:   $\nabla C_{t,a} \leftarrow$ grad_cost $(\nabla C_o, \nabla C_m)$
19: end for
20: for all agents $a$ do
21:   $\mathbf{v}_a \leftarrow$ adapt_agent_motion $(\nabla C_{t,a})$
22: end for
```

The first loop fetches the agent's current position $\mathbf{p}_a$ and the current velocity $\mathbf{v}_a$ (line 2) which are then used to set up its virtual camera (line 3). In line 4, environment obstacles are rendered to the perceptual space (a texture). The loop over perceived pixels starts in line 5. If an obstacle o_i is visible through pixel p_i (line 6), the corresponding obstacle is retrieved and its relative motion is deduced (lines 7 to 9). In line 10, $ttca_{o_i,a}$ is computed according to Equation (3.7). Simillarly, in line 11, Equation (3.10) is used to compute

$dca_{o_i,a}$. The partial derivatives of the current obstacles cost are then computed in line 12. Next, the gradient of the obstacles cost (line 16), the gradient of the movement cost (line 17) and the gradient of the total cost (line 18) are computed. The second loop (lines 20 to 22) updates the simulation by iterating and adapting the motion of each agent, using the gradient ∇C_t and Eq. (3.18).

4.1.2 Technical aspects

The algorithm implementation takes place in CPU and GPU. Per pixel operations in lines 5 to 15 are executed in parallel: they were implemented in OpenGL Shading Language (GLSL). More precisely, lines 7 to 9 were implemented in the vertex shader while lines 10 to 13 were implemented in the fragment shader. The computation of the gradient of the obstacles cost (line 16) is suitable for parallelization, but for now just a CPU version was implemented. The rest of the algorithm was implemented using C++.

As for the camera set up, it has been used the same settings as (ONDŘEJ et al., 2010): a field of view of 150°, a height of 80°, an orientation towards the ground with an angle of $-40°$(so that the upper clipping plane is horizontal), and a resolution of the vision texture of 256 × 48 pixels. The camera is positioned according to agent's eye level and oriented according to its motion direction.

The agents also have been represented in a similar fashion to (ONDŘEJ et al., 2010) so as to enable comparisons: cones with 0.3m of radius and 1.7m of height. Cones are used because of the simplified geometry and the similarity to human shape in the sense of usually being wider at the base than at the top. Cones also allow the agent to see behind each others 'shoulders'. However, the proposed approach allows agents and obstacles to have arbitrary shapes and sizes, as it is shown in Chapter 5.

In the next section, the role and influence of each model's parameter are detailed. A data-driven setup for fitting parameters is also presented.

4.2 Model parameterization

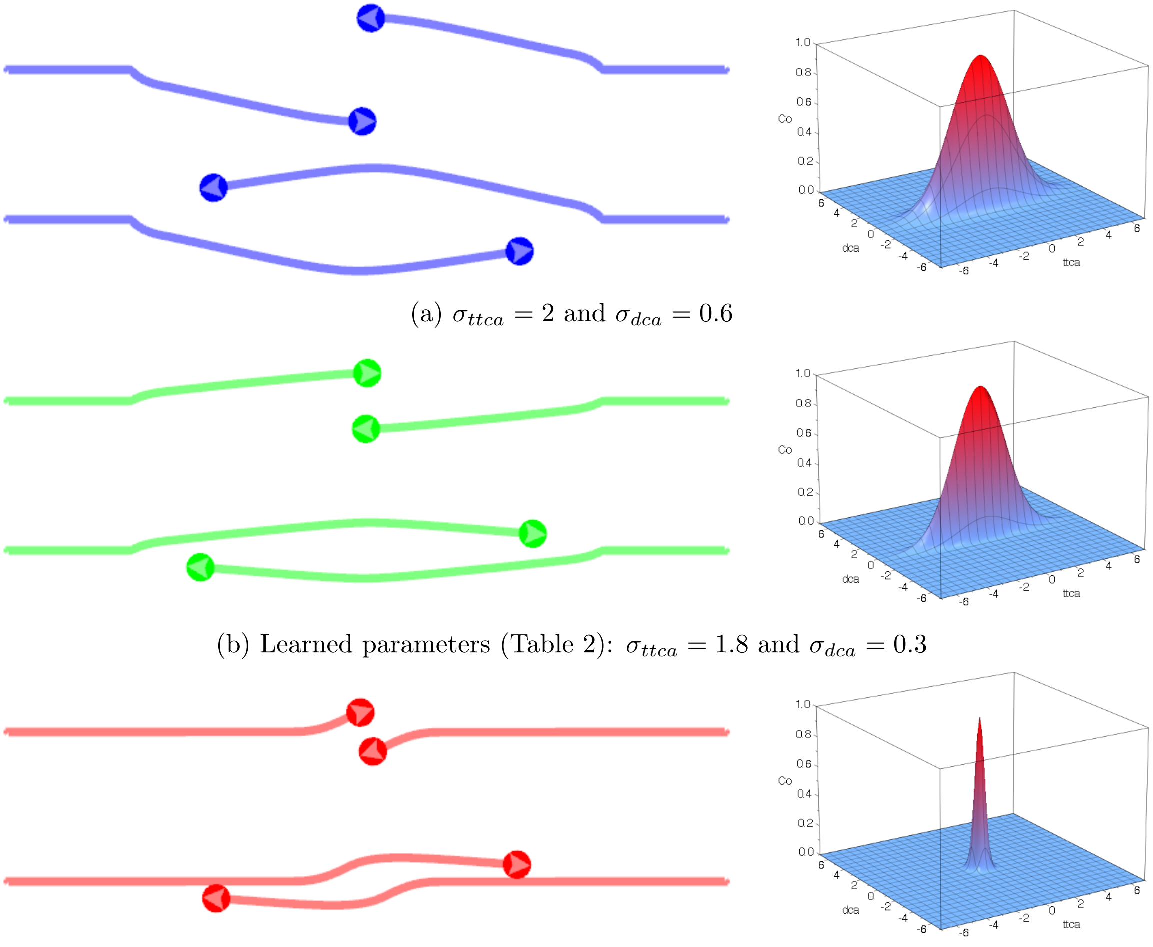
In Chapter 3, the cost function was introduced as a composition of Gaussian functions (equations (3.11), (3.15), (3.13) and (3.14)). The shape of that function depends on four parameters. The influence of these parameters on the agents' behavior is discussed in Section 4.2.1, where it is shown that they allow to control for each agent: the speed adaptation, the orientation adaptation, the distance to keep from obstacles and the anticipation to motion adaptation. These parameters can be defined empirically, but Section 4.2.2 details how to define a set of parameters according to real data obtained from experiments.

4.2.1 Influence of model parameters

The proposed algorithm has four parameters: σ_{ttca} and σ_{dca} for the obstacles cost function (Equation (3.15)); and σ_s and σ_{α_g} for the movement cost function (equations (3.13) and (3.14)). Their values can be intuitively adjusted by users. Indeed, σ_{ttca} directly affects the agent's anticipation time whereas σ_{dca} modulates the minimum distance to obstacles. Their effects on avoidance trajectories are shown in Figure 44. Increasing σ_{dca} results in a larger minimum distance between the agents whereas increasing the σ_{ttca} results in an earlier motion adaptation to avoid collisions.

The movement cost parameters have direct impact on avoidance strategies. Greater σ_s values result in a preference for adapting speed in order to avoid collisions, whereas, on the other hand, larger σ_{α_g} values

Figure 44 – Influence of the parameters on the agents' motion. Variation of the obstacles cost function's parameters σ_{ttca} and σ_{dca} (Equation (3.15)). Images on the left show trajectories and images on the right show the C_o plots.



(a) $\sigma_{ttca} = 2$ and $\sigma_{dca} = 0.6$

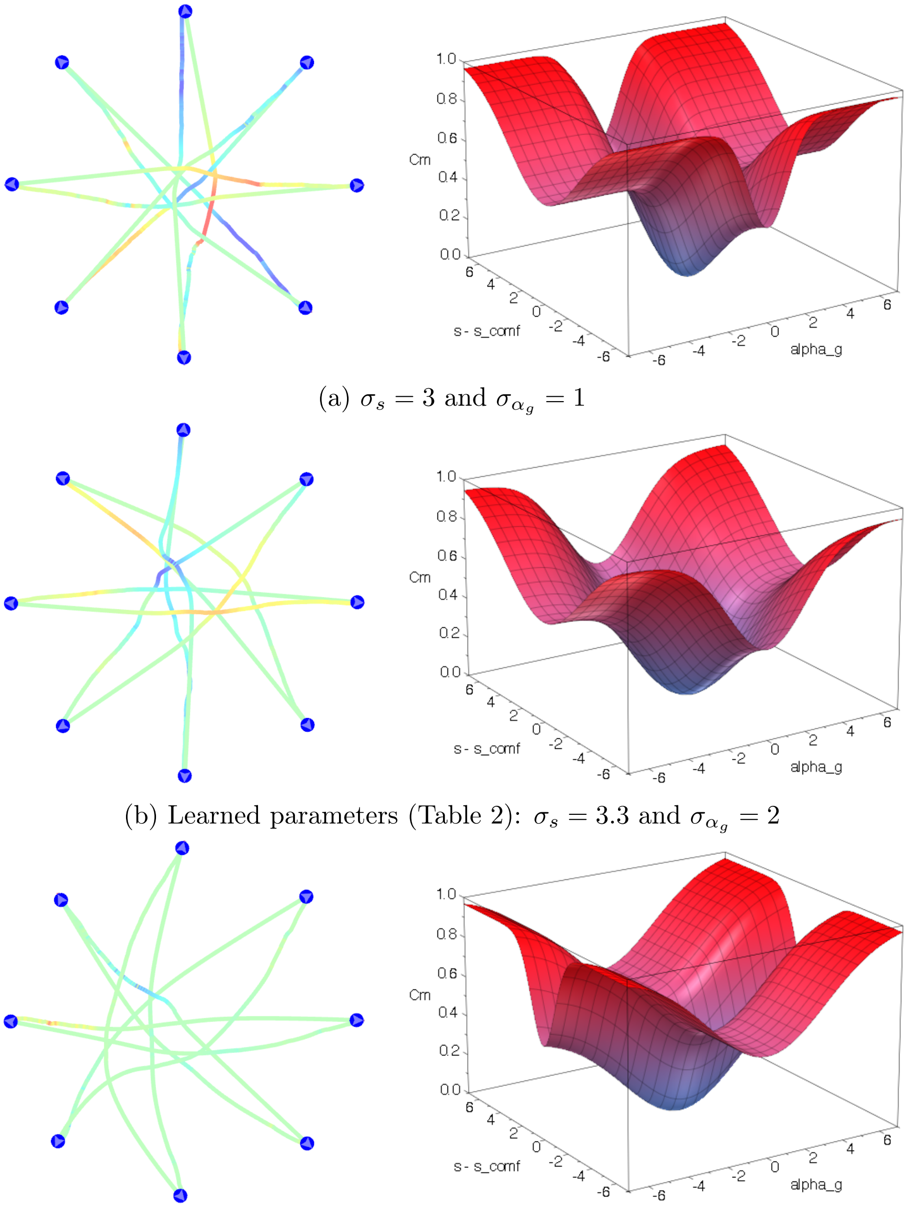
(b) Learned parameters (Table 2): $\sigma_{ttca} = 1.8$ and $\sigma_{dca} = 0.3$

(c) $\sigma_{ttca} = 0.3$ and $\sigma_{dca} = 0.3$

Source: the author.

favors the agents' reorientations. This effect is illustrated in Figure 45: eight agents are initially disposed in a circle and have to reach the diametrically opposite position. The top row (Figure 45 (a)) corresponds to a large value for σ_s and a small value for σ_{α_g} (respectively 3 and 1): speed adaptation is favored over changes of direction. Trajectories are mainly rectilinear, some agents wait and give way to others rather than going around the obstacles (speed is color coded). The bottom row (Figure 45 (c)) corresponds to the opposite configuration: a large value for σ_{α_g} and a small value for σ_s. It can be seen from the depicted trajectories that agents tend to prefer an adaptation of the motion direction. Figure 45 (b) shows results for an intermediate configuration (using the learned parameters of Table 2 discussed in the next section), in which the agents adapt both their speed and direction to avoid collisions. These four parameters discussed here can be set differently for each agent, allowing the simulation of heterogeneous agents.

Figure 45 – Influence of the parameters on the agents' motion. Variation of the goal cost function's parameters σ_{α_g} and σ_s (equations (3.13) and (3.14)). Images on the left show trajectories and images on the right show the C_m plots.



(a) $\sigma_s = 3$ and $\sigma_{\alpha_g} = 1$

(b) Learned parameters (Table 2): $\sigma_s = 3.3$ and $\sigma_{\alpha_g} = 2$

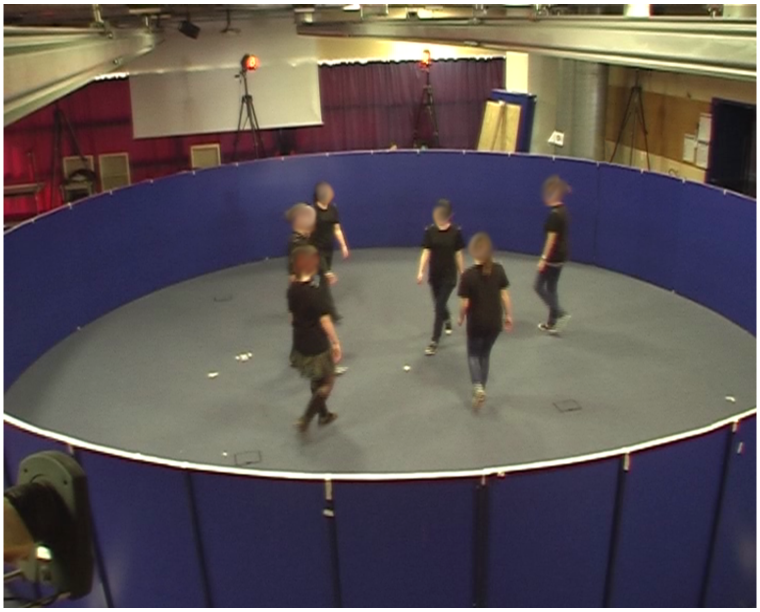
(c) $\sigma_s = 1$ and $\sigma_{\alpha_g} = 3$

Source: the author.

4.2.2 Data-based parameters setup

An objective strategy to find the parameters' setting consists in fitting the model to experimental data. To this end, the framework developed by (WOLINSKI et al., 2014) to learn the model's parameters from experimental data was used. The Wolinski's framework allows using several different metrics for comparing the trajectories synthesized by the model and those experimentally measured. The motion data used in this work is illustrated in Figure 46. In that experiment, six people were positioned on the border of a circle and needed to reach the diametrically opposite position.

Figure 46 – Photo of the experiment where six people were positioned on the border of a circle and needed to reach the diametrically opposite position.



Source: MimeTIC team at INRIA-Rennes Bretagne Atlantique.

For comparing real and simulated trajectories, it was used the progressive difference metrics, a variant of Guy's metrics (GUY et al., 2012) which is prone to capture global motion features and avoids overfitting problems. Table 2 shows the values of the model's parameters after calibration. The results depicted in Figure 47 show that, after calibration, our model produces trajectories with motion adaptations similar to the real data. Although trajectories do not exactly fit (as expected, given the used metrics), global features are similar: some agents adapt its direction, while others tend to go in a straight line; some accelerate, and others decelerate.

Table 2 – Model's parameters learned from data.

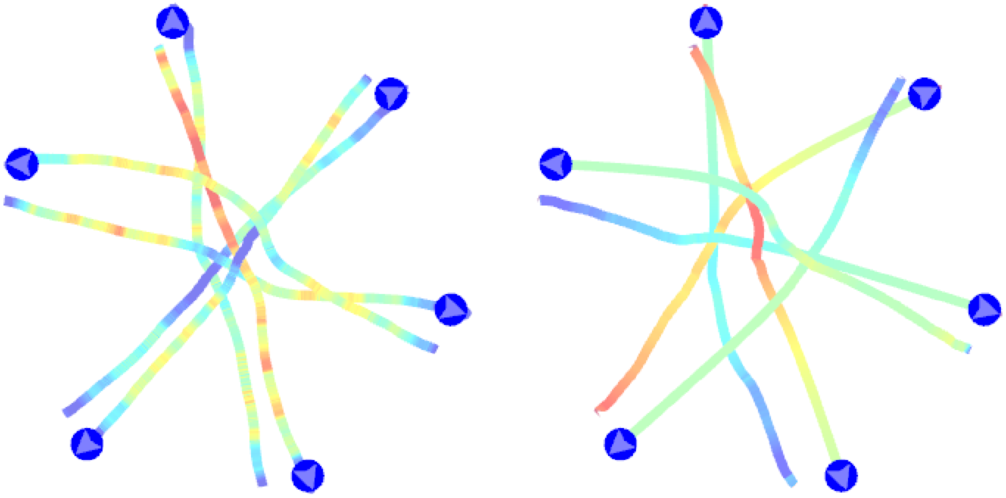
	σ_{α_g}	σ_s	σ_{ttca}	σ_{dca}	s_{comf}
Value	2.0	3.3	1.8	0.3	1.5
Variance	0.9	0.5	0.3	0.0	0.1

Source: the author.

4.3 Final considerations

In this chapter, the technical aspects of the proposed model were presented. Moreover, the influence of each parameter of the model was discussed and examples of resulting trajectories for different parameters setting were shown as well as the plots of the obstacles cost and movement cost functions. Then, Wolinski's framework (2014) was used to calibrate the model according to experimental data.

Figure 47 – Comparison of experimental trajectories (a) with the trajectories generated by the proposed model after fitting the model to data (c). It can be seen that, after calibration, the agents are able to follow the experimental data. Nevertheless, for some particular cases, a different parameter setting might be required.



(a) Experimental trajectories

(b) Our model trajectories

Source: the author.

Now that the model was introduced and the role of its parameters was defined, it is possible to present and to discuss the results obtained with it. This is done in the next chapter where the model is compared with other important approaches in several challenging scenarios.

5 Results

In this chapter, the results obtained with the proposed model are presented and discussed. For evaluating the new approach and for making comparisons among different approaches, several challenging scenarios were defined. Those scenarios are described as follows:

Room, illustrated in Figure 40. Three agents move in a room. This is the simple test-case example used to introduce our technique.

S-corridor, illustrated in Figure 48 (a). Five agents are placed at the beginning of a sinuous corridor. Their goal is set at the corridor's exit. Note that no intermediary waypoint is set.

Corridor, illustrated in Figure 48 (b). Agents are set in two opposite groups going to opposite directions in a corridor.

Opposite, illustrated in Figure 48 (c). Agents are set in two opposite groups going in opposite directions.

Columns, illustrated in Figure 48 (d). Six static obstacles (columns) are added to the **Opposite** scenario.

Multi-obstacle, illustrated in Figure 48 (e). Groups of agents need to traverse scenario composed of several obstacles.

H-corridor, illustrated in Figure 48 (f). A group of agents traverses the corridor from left to right. During the traversal the agents need to avoid moving obstacles crossing the corridor vertically.

Crossing, illustrated in Figure 48 (g). Agents are set in two groups which must cross orthogonally.

1-D Periodic Corridor, illustrated in Figure 48 (h). Agents are placed in a line in a periodic corridor, i.e., when an agent reaches the end of the corridor it reenters at the beginning.

Circle, illustrated in Figure 48 (i). Agents are set on the border of a circle and need to reach the diametrically opposite position.

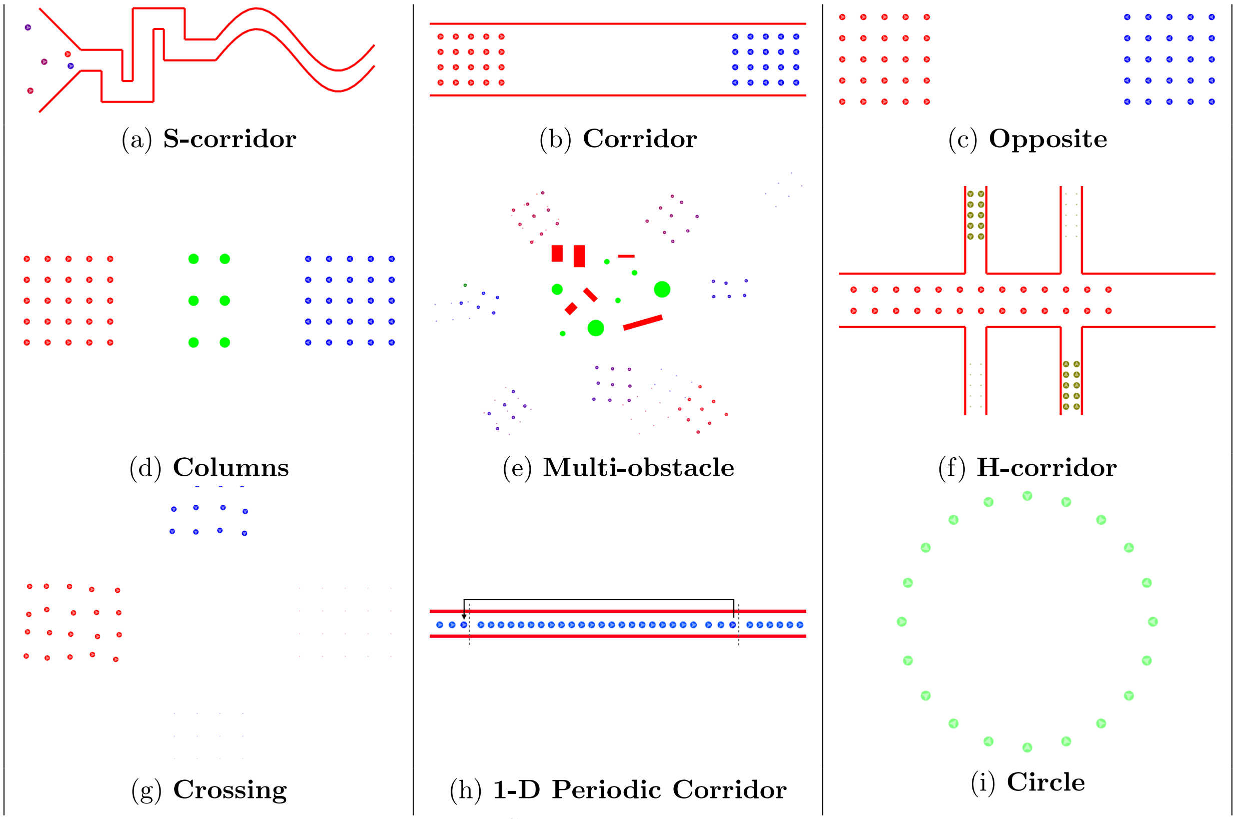
Some of those scenarios are changed during the chapter for experimental purposes. For example, the *Opposite* scenario is tested also with random initial positions to disrupt symmetry.

The results are described through the following sections. In sections 5.1 and 5.2, qualitative and quantitative evaluations are performed, respectively. In the following section, the performance is measured and a comparison between the models based on synthetic vision is made. Finally, some considerations are draw in the last section. The companion video[1] shows the corresponding animations with moving 3-D characters. The proposed model is compared with the following models:

OSV The original approach based on synthetic vision proposed by Ondřej et al. (ONDŘEJ et al., 2010), which was selected as a representative of previous vision-based algorithms;

[1] http://1drv.ms/1Bgd6JX

Figure 48 – The initial configuration of each scenario.


(a) **S-corridor**

(b) **Corridor**

(c) **Opposite**

(d) **Columns**

(e) **Multi-obstacle**

(f) **H-corridor**

(g) **Crossing**

(h) **1-D Periodic Corridor**

(i) **Circle**

Source: the author.

RVO2 A broadly used algorithm (and therefore an excellent benchmark model), which is representative of velocity-based avoidance techniques (BERG et al., 2011).

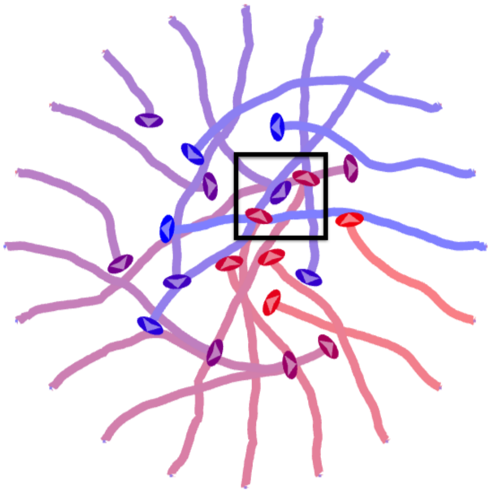
It was also compared with a particle-based model (HELBING et al., 2000). However, since its results were consistently worse than those of OSV and RVO2, the results were not reported. Ideally, the model could be compared to many other techniques, but only the most relevant techniques to compare with were retained here.

5.1 Qualitative evaluation

Synthetic vision enables processing static and dynamic obstacles of any shape in the same fashion: everything is abstracted as visible pixels moving relatively to each agent. In Figure 49, for example, it is shown a simulation in the *Circle* scenario with ellipse-shaped agents, which are more similar to the human shape. The highlight in Figure 49 shows how the space for maneuvers is affected by the shape of the agents by using this representation of cones with a ellipsoidal base. In that case, if the agents were circular-shaped, the blue agent would not fit between the red ones.

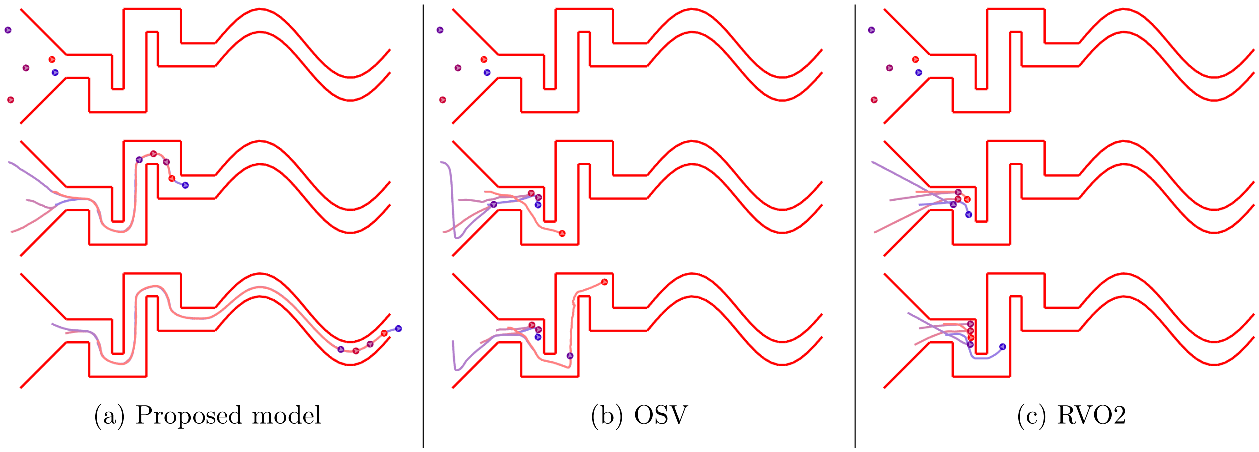
This process of abstracting the environment into pixels also implicitly filters invisible obstacles and gives priority to nearer obstacles (of equivalent size) as they are represented by more pixels. These properties are illustrated by the *Room*, *Columns*, *S-corridor*, *Corridor*, *H-corridor* and *Multi-obstacle* scenarios which mix static obstacles and moving agents. The *S-corridor* scenario (Figure 50) is particularly challenging because

Figure 49 – **Circle scenario** with ellipse-shaped agents. This example illustrates the ability of the proposed model of representing agents of any shape. The highlighted part shows how the space for maneuvers can be increased by using another kind of representation like this.



Source: the author.

Figure 50 – Comparison of the results for the **S-corridor scenario**. The agents must traverse the corridor so as to reach their goals. Trajectories were synthesized without resorting to a global path planner. Note that only for the proposed model (a) the agents could reach the end of the corridor. In the other models ((b) and (c)) most of the agents got stuck at the first turn.



(a) Proposed model

(b) OSV

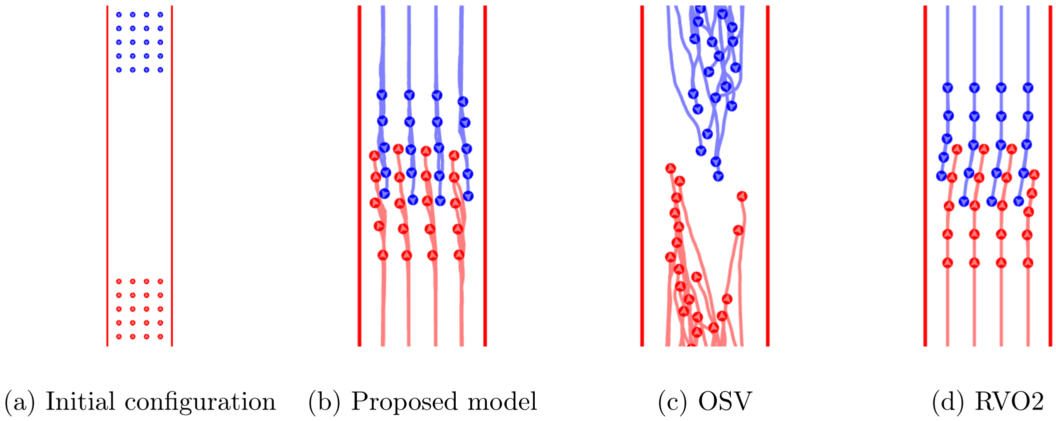
(c) RVO2

Source: the author.

no intermediary waypoint is defined for agents. It illustrates what the new proposed control loop brings in terms of motion quality: agents are perfectly able to find a locally optimal trajectory sketched by the shape of the corridor. None of the other techniques tested on this scenario were able to correctly steer agents through the environment without global motion planning.

The achieved improvements in terms of quality can also be visually assessed in the *Corridor* (Figure 51), *Opposite* (Figure 52) and *Columns* scenarios (Figure 53). In the *Columns* scenario, more specifically, the presence of obstacles forces the agents to reorganize themselves. The proposed model and OSV produce plausible emerging patterns: traffic is segregated in lanes according to the moving direction and the position

Figure 51 – Comparison of the results for the **Corridor scenario**. The groups of agents must traverse the corridor so as to reach their goals at the opposite side. In (b) and (d) lanes are kept and agents pass close to each other, while in (c) strong patterns are observed where the groups keep a large distance from each other.


(a) Initial configuration (b) Proposed model (c) OSV (d) RVO2

Source: the author.

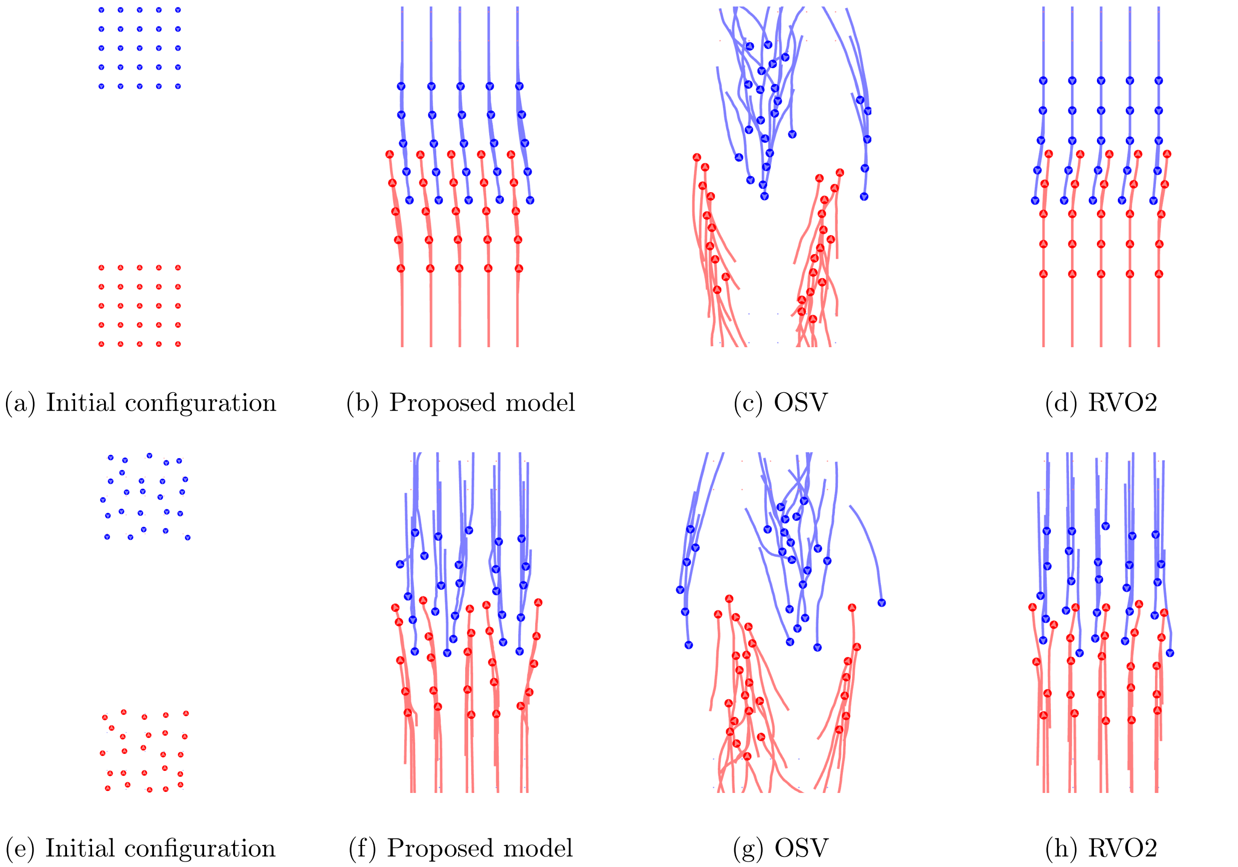
of the columns, yielding a pattern which is easily observed in the real world. However, the results of OSV model show that agents keep a very large distance to other agents moving in the opposite direction. This results in too strong emergent pattern signatures and unnatural interaction distances. The proposed model, on the other hand, searches for the locally safest moves, which prevents our animation from displaying too strong patterns and agents at too close or too large distances. It is able to perfectly adapt agents motion to the scenario by merging lanes of agents moving in the same sense. As for the RVO2 model, it makes agents perform the strictly required amount of maneuvers to avoid collision. Such a simplistic approach does not seem sufficient to solve this type of scenario: some agents agglomerate behind the columns and the expected lanes pattern does not emerge. The histogram in Figure 54 also suggests that the proposed model deals better with this scenario: most of the agents move at a speed close to their comfort speed ($1.5m/s$) and, unlike for the other models, there are no agents moving very slowly or almost stopped.

In the *H-corridor* scenario (Figure 55), a group of controlled agents (in red) has the goal of horizontally traversing a corridor. However, two groups of non-controlled agents (in green) vertically cross the corridor, temporarily obstructing the controlled agents' path. To reach their goal without collisions, some of the agents must stop and wait for the moving obstacles to pass. Figure 55 shows that the three models provide clearly different results. For OSV, the agents keep a very large distance to the walls, forming a single lane at the center of the corridor. The agents are able to wait for the moving obstacles to pass when needed, although some of them are dragged by the flow of non-controlled agents. As for RVO2, the agents get too close to the moving obstacles, which makes some of them be dragged. Finally, for the proposed model, most of the lanes are kept, the agents do not get too close to the moving obstacles and some agents accelerate to anticipate the obstacles' movement. Visually, both results (*Columns* and *H-Corridor*) are more satisfying as shown in the video[2], which corroborates the initial qualitative evaluation.

Finally, in the *Multi-obstacle* scenario (Figure 56) the differences between the models are also highlighted. For OSV, the agents avoid passing too close to each other, then the trajectories are sparser. Whereas for

[2] http://1drv.ms/1Bgd6JX

Figure 52 – Comparison of the results for the **Opposite scenario**. The groups of agents must reach their goals at the opposite side. In both aligned (top) and unaligned (bottom) situations lanes of agents can be observed for all the models. OSV produces strong patterns, the lanes are very far from each other. In RVO2, the agents do not respect personal space. In our model, the agents do not spread as OSV and respect personal space at same time.



(a) Initial configuration (b) Proposed model (c) OSV (d) RVO2

(e) Initial configuration (f) Proposed model (g) OSV (h) RVO2

Source: the author.

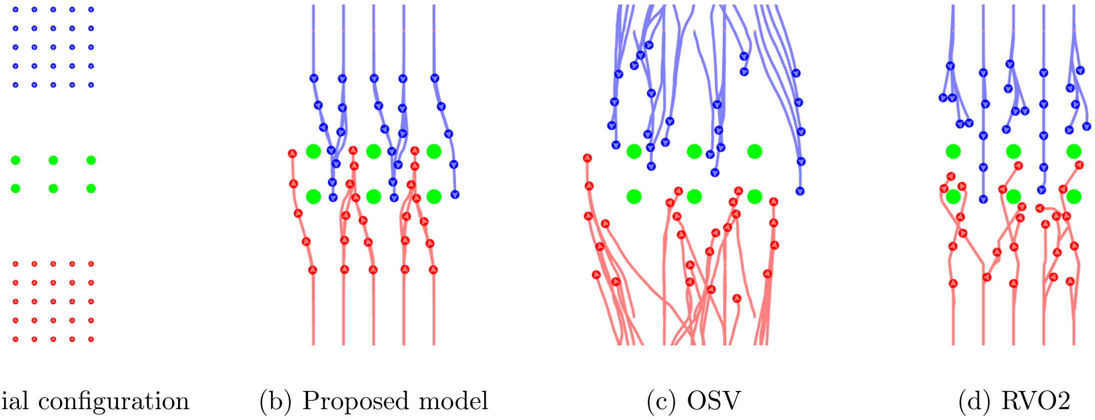
RVO2, two things can be observed: first, the trajectories are too rectilinear which means the agents do not adapt direction too much; and, second, the agents cannot anticipate and avoid a traffic jam of occurring at the center. The proposed model is a mid-term, where the agents do not keep unrealistic distances from each other neither fall into traffic jams.

5.2 Quantitative evaluation

5.2.1 Microscopic features

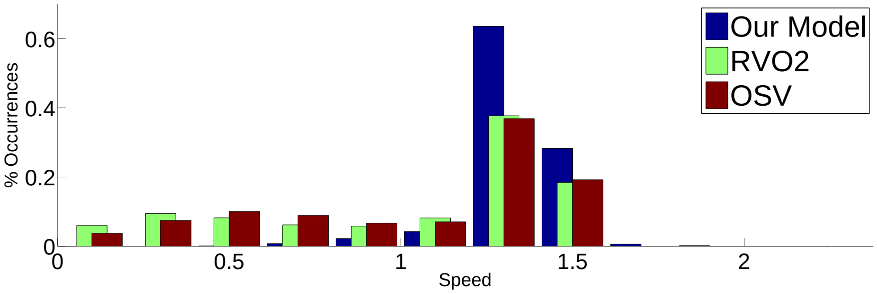
In this section, a microscopic evaluation of the tested methods is made, i.e., an evaluation based on the individual behavior of each of the agents. To reach this goal, it was used the Wolinski's framework introduced in (WOLINSKI et al., 2014) so as to fit the models to a circle scenario (with 12 people) available with the framework. To measure the distance between the simulations and the read data, the progressive difference metrics (a fast variant of the entropy metrics by (GUY et al., 2012)) was used, which is especially interesting since it captures the global motion characteristics instead of making agents follow exactly the real trajectories.

Figure 53 – Comparison of the results for the **Columns scenario**. Similar to the *Opposite scenario* but with columns. For OSV the strong patterns are present, but agents are still able to organize lanes and to avoid the columns. In RVO2, the agents are not able to produce lanes and some get stuck for a while when facing the columns. In the proposed model the lanes are still formed with no emergence of a strong separation as in OSV.


(a) Initial configuration (b) Proposed model (c) OSV (d) RVO2

Source: the author.

Figure 54 – Histograms showing the distribution of the speed in the **Columns scenario** for the three tested algorithms: the proposed model, OSV, RVO2.



Source: the author.

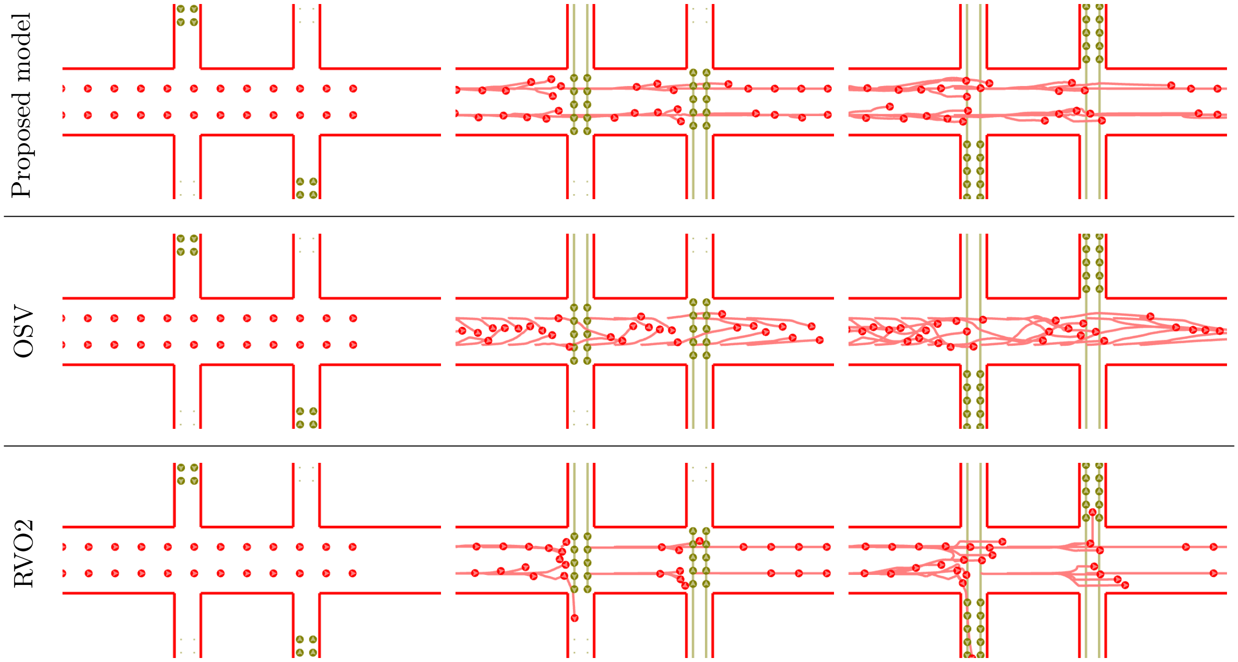
The following scores were obtained: 0.26 for RVO2, 0.35 for OSV and 0.06 for the proposed gradient-based model, where lower score means better performance.

5.2.2 Macroscopic features

The objective of the analysis made in this section is to quantify and compare global features of the trajectories generated by the three tested models. Let us start by analyzing the traffic segregation phenomenon observed for the *Opposite* scenario in the qualitative analysis. This phenomenon can be illustrated with the two histograms of Figure 57. The left histogram shows the distribution of distances between nearest agents moving in the same direction (red-red or blue-blue nearest agents). The right histogram shows the distribution of distances between nearest agents moving in opposite directions (red-blue nearest agents). The proposed approach is the one resulting into smallest differences between the two distributions. Visually, such result is more satisfying as shown in the video[3], which corroborates the initial quantitative evaluation.

[3] http://1drv.ms/1Bgd6JX

Figure 55 – Comparison of the results for the **H-corridor scenario**. The red agents want to reach their goals which makes them move to the right. However, two obstacles, composed of non-controlled agents (in yellow), traverse the flow of red agents. In the proposed model, agents anticipated the motion of obstacles and stopped while the obstacles cross the corridor, then they started moving again. In OSV, agents also waited but they formed a line to not get close to the walls. In RVO2, agents stopped too late and some got lost and others were dragged by the obstacles.

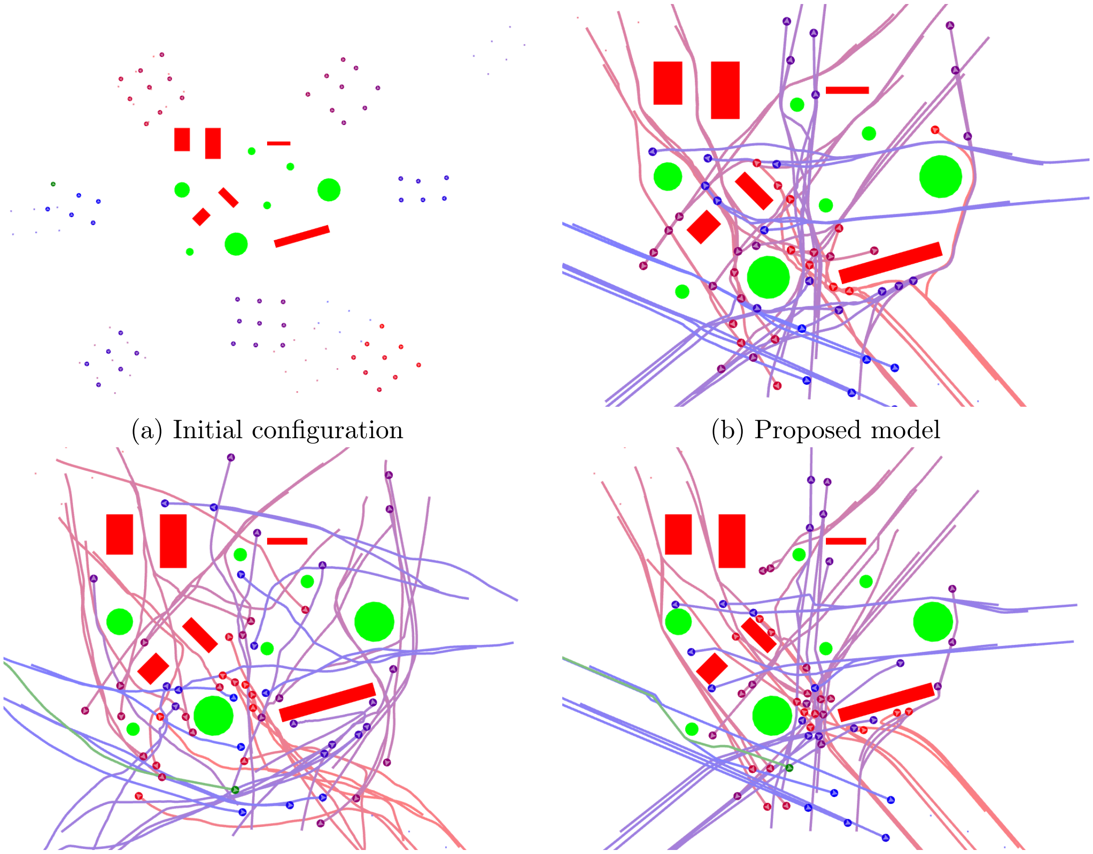


Source: the author.

Let us focus on the *Crossing* scenario shown in Figure 58, where two perpendicular flows of agents must cross. The top row depicts the resulting trajectories for the proposed model, OSV and RVO2. It can be clearly seen that the OSV model makes the agents cross at a very large distance, which makes them disperse unnecessarily. RVO2, on the other hand, makes minimal adaptation in direction and speed, which causes the agents to cross at very small and implausible distances. As for the proposed model, it provides an intermediate solution where the agents tend to keep a natural distance between themselves (not too large as OSV, not too close as RVO2). This feature is quantified by the distances histogram shown in Figure 59, where the occurrences of large distance is dominated by the OSV model and the occurrences of distances very close to zero are dominated by RVO2.

Still regarding Figure 58, it was performed a cluster analysis to quantify the emergence of 45° lanes patterns as observed in real crowds (CIVIDINI et al., 2013). To achieve this goal, it was implemented the clusters detection algorithm of (MOUSSAïD et al., 2012) which states that "two pedestrians belong to the same cluster at a given moment of time if one of them is following the other". The orientation of each detected cluster is determined by fitting a line to its agents. The bottom row of Figure 58 shows the detected clusters for a given step in the crossing scenario. The expected pattern is clearly well defined in the case of the proposed model, where the 11 clusters were detected with an average slope of 45° and a small standard deviation (±2°). For OSV, 26 clusters emerged with an average slope of 32° and a high standard deviation (±41°). As for RVO2, it does not produce the expected pattern: the initial structure of agents is almost unchanged after passing the crossing zone.

Figure 56 – Comparison of the results for the **Multi-obstacle scenario**. The same global features observed in the other scenarios are present in this scenario. In OSV, the agents keep a distance to moving obstacles which seems to be too large, whereas in RVO2, the agents pass too close, and at some point a traffic jam can be observed close to the center.



(a) Initial configuration

(b) Proposed model

(c) OSV

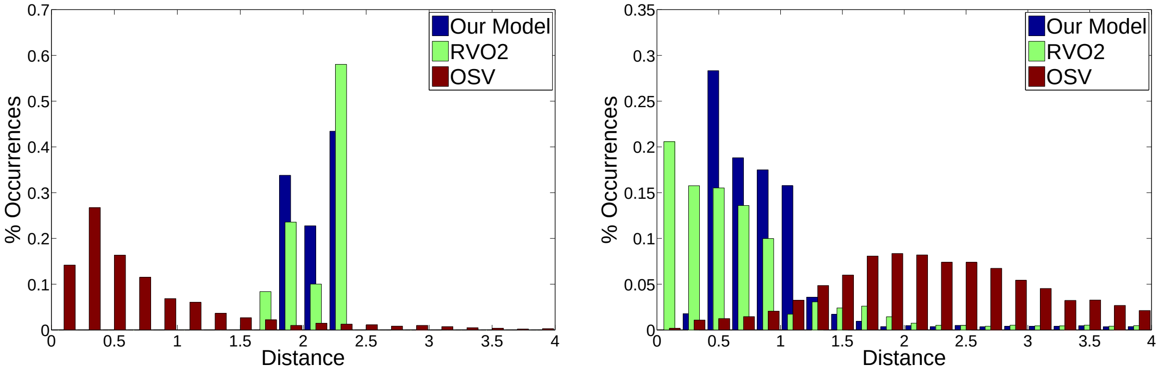
(d) RVO2

Source: the author.

Another way of quantifying global features of a model is to determine its Fundamental Diagram (FD) (SEYFRIED et al., 2005), which measures the speed of the agents as a function of the density of agents in a given area. The agents' speed as a function of the density was measured using a *1-D Periodic Corridor* scenario (as depicted in Figure 48 (h)) that works similarly to the circular corridor used for experiments with real people in (JELIĆ et al., 2012). In this 1-D case, the density is measured as $\rho = \frac{n}{p}$, where n is the number of agents and p is the circle's perimeter. Several experiments were performed with varying agents density for the three tested models using the same parameters setting. The resulting FDs are shown in Figure 60, as well as a FD issued from real data from (JELIĆ et al., 2012). It can be seen that, out of the three tested models, the proposed model presents the most similar curve to the real data. In OSV, the speed is not highly affected by the density, whereas in RVO2 sometimes the speed increases with the density and such behavior is not observed nor expected in fundamental diagrams.

Finally, the circle example (Figure 61) shows the ability of the proposed technique to explore the full range of reachable velocities to perform collision avoidance maneuvers. Reorientations are noticeable from the agents' trace, whereas speed variations are encoded in the trail color. The parts of the trails with a color closer to red correspond to speed accelerations to pass in front of some other agents. Blue corresponds to

Figure 57 – Histograms showing the distribution of the distance to the nearest neighbor (NN) of the same group and to the NN of the opposite group, in the unaligned **Opposite scenario** for the three tested algorithms: the proposed model, OSV, RVO2.



(a) NN distance - same group

(b) NN distance - other group

Source: the author.

parts of the agent's path where the agent moved slower than its comfort speed. OSV reactions are limited to turns to avoid future collisions and decelerations to avoid imminent impacts: this limited set of reactions is clearly visible in this scenario. RVO2 does not handle well these highly symmetric situations: reachable velocities are completely constrained and agents stop. Even when noise is added to such situations, the simulation results into unnatural gathering of agents at very close distance in the center of the circle (see the video[4]). Figure 62 shows the histogram of the agents' speed for the circle scenario. The result confirms that the proposed model is the single one which agents accelerate to avoid collisions. Moreover, the proposed model is the one with a larger number of agents moving at a speed close to their comfort speed ($1.5m/s$) and less slow agents (speed $< 0.5m/s$).

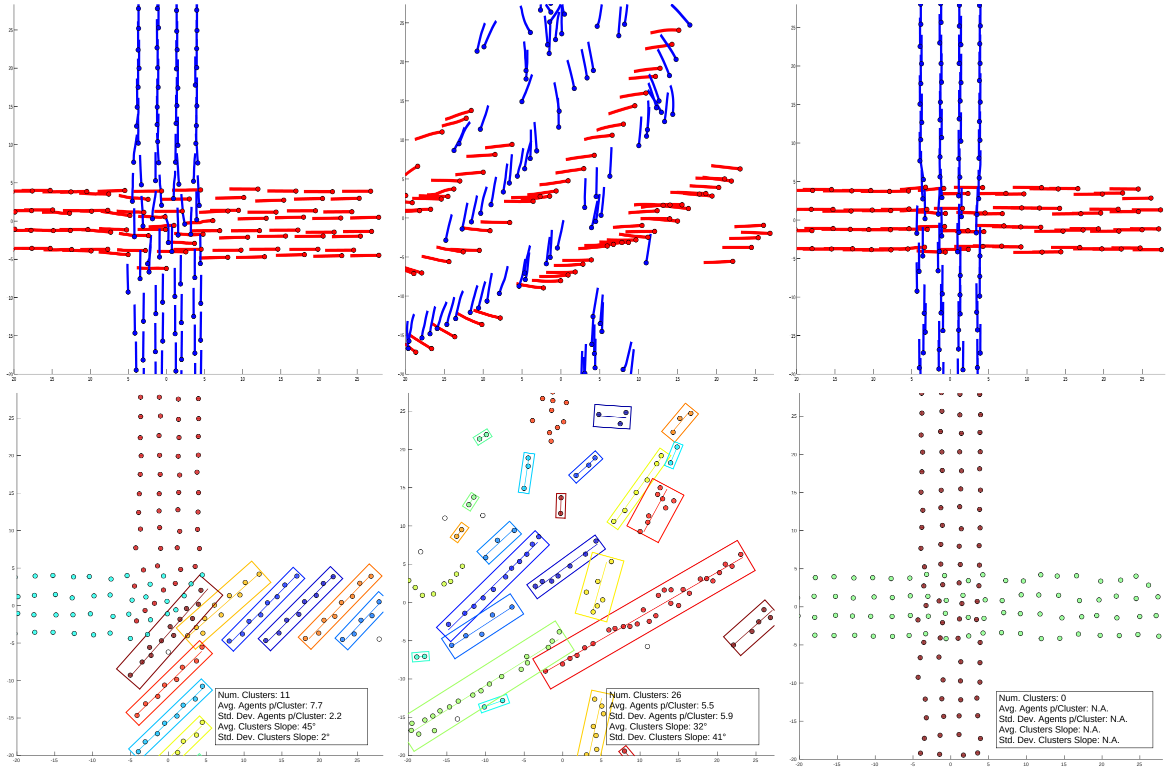
5.3 Performance

The simulations were performed on a MacBook with 2.13 GHz Intel Core 2 Duo processor, 4.0 GB of RAM, NVidia GeForce 9400M graphics card with 16 CUDA cores and 256 MB of dedicated RAM. The algorithm was written in C/C++ and OpenGL was used for graphics. Figure 63 (left), displays the average simulation loop runtime compared with OSV on the *Opposite* scenario. Synthetic vision techniques cannot compete with RVO2-like geometrical approaches, but still shows reasonable performances as it can run at 10 FPS for 300 agents in a low-profile computer. Compared with OSV, the proposed technique performs a larger amount of computations in GPU. However, it only needs a 2-channel texture to store the results (the two partial derivatives), whereas OSV requires a 4-channel texture. This makes the GPU/CPU data transfer time slightly faster for the proposed model. Moreover, it only process two texture channels in CPU whereas OSV must process four. This is why, overall, both methods roughly exhibit the same execution time.

Synthetic vision algorithms manipulate large amounts of data (12288 pixels per agent for a 256×48 camera resolution, for example). The main bottleneck is the texture download from GPU to CPU. This can be alleviated by downsizing the camera's resolution. Figure 63 (right) shows the performance of the proposed model for different camera resolutions in the *Opposite* scenario with 50 agents. Figure 64 shows the effect of

[4] http://1drv.ms/1Bgd6JX

Figure 58 – Results for the **Crossing scenario**. Two groups of unaligned agents moving orthogonally. The images in the following columns show the groups of agents separated by flow (red goes to the right and blue goes to the bottom) on the top and separated by clusters on the bottom for each of the three tested models.



(a) Proposed model

(b) OSV

(c) RVO2

Source: the author.

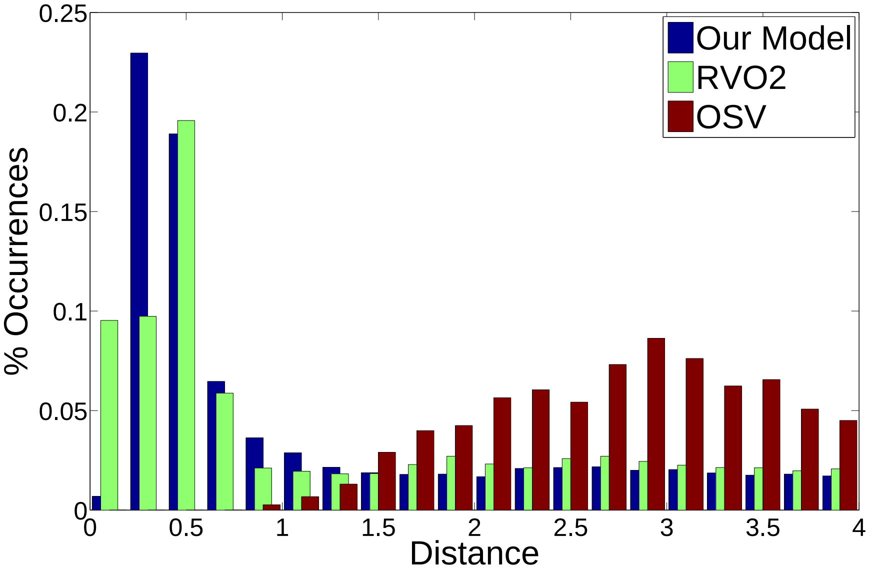
resolution decrease on simulation results for the proposed model and OSV. Results show that the proposed new technique is much less sensitive to this parameter. Visual differences in the trajectories can hardly be observed even when reducing the original resolution by 93.75%. A special case of camera resolution is 256×1, in which case the agent keeps a wide view (rightmost column of Figure 64). Finally, Figure 65 shows the performance of the proposed model when varying the number of agents and the camera resolution. It is noticeable that the more agents lesser is the influence of the camera resolution on performance.

5.4 Final considerations

The results presented in this chapter show the ability of the new technique proposed for local collision avoidance. Moreover, it was demonstrated several improvements on the quality of trajectories over previous representative works. Wolinski's framework showed that the proposed model is more prone to reproduce global features observed in real data. Also, the strong patterns observed in the other models are not present here, as the agents in the proposed model distribute the space among them similarly regardless the obstacles' flow.

Regarding only techniques based on synthetic vision some observations could be made. First, the proposed technique is less sensitive to the camera resolution. And second, despite the additional computation

Figure 59 – Histograms showing the distribution of the distance to the nearest neighbor (NN) of the opposite group, in the **Crossing scenario** for the three tested algorithms: the proposed model, OSV, RVO2.

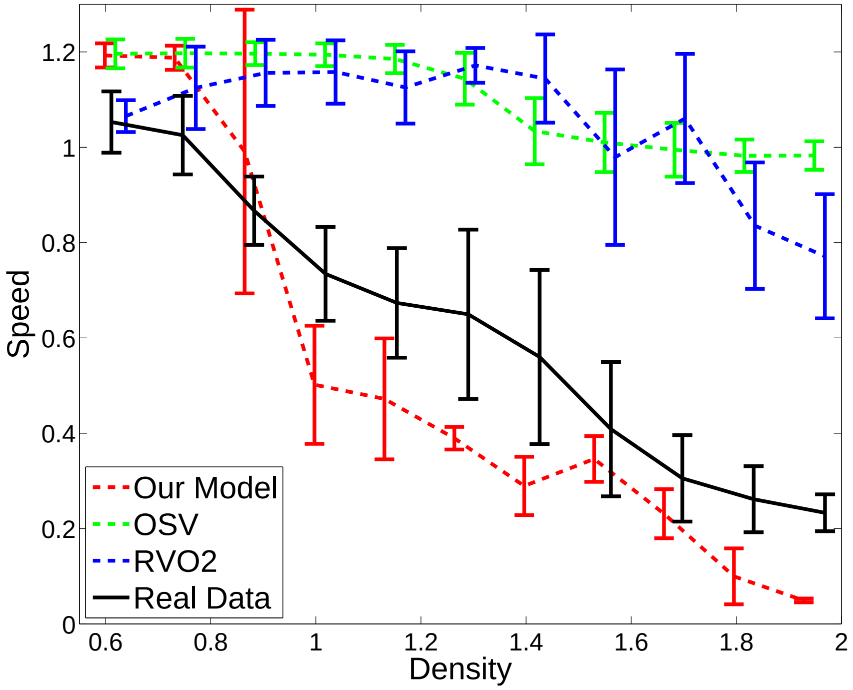


Source: the author.

required, the proposed technique has similar performance compared with OSV.

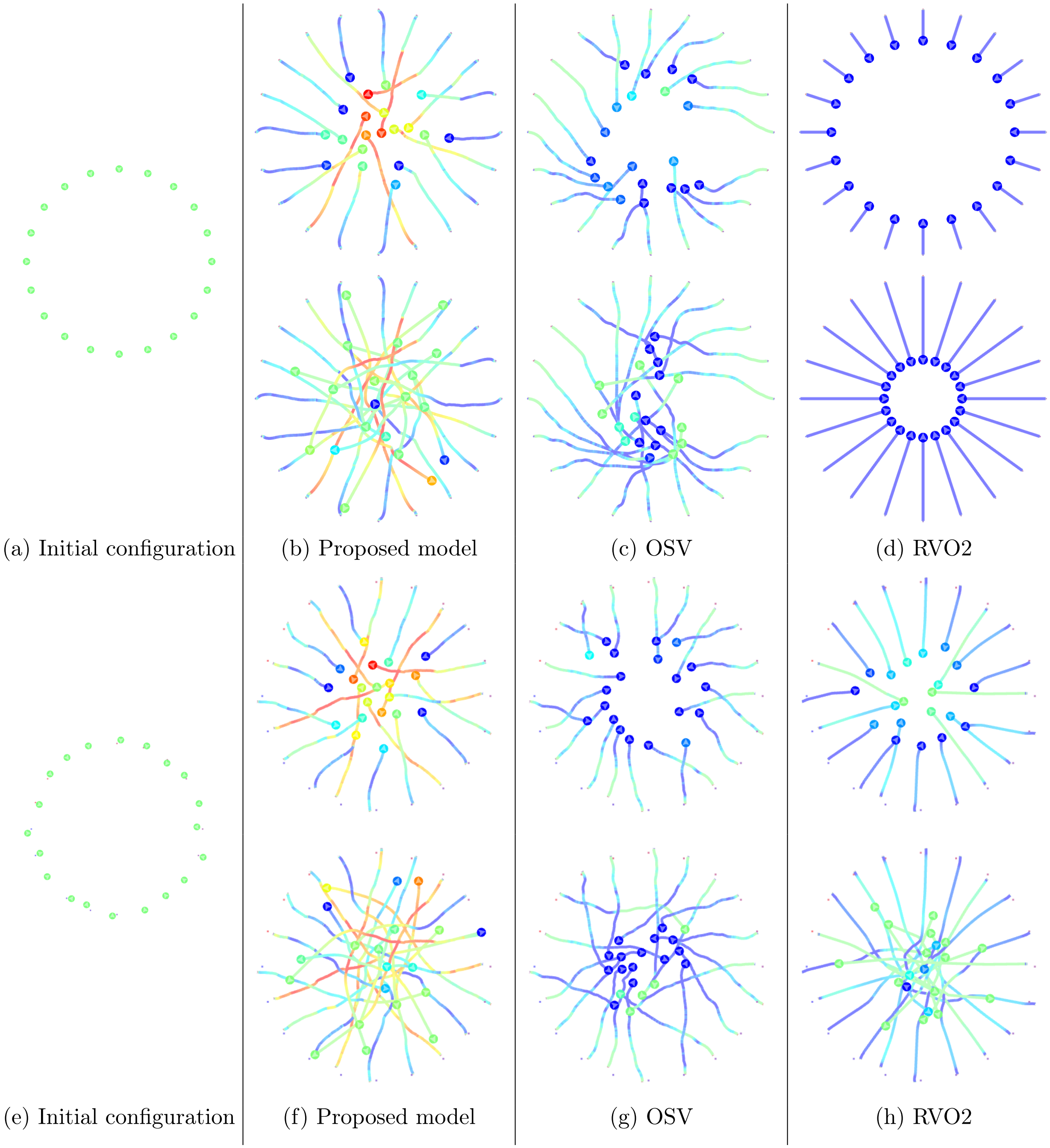
In the next chapter, an overall discussion of the contributions of this work is made. It is followed by a discussion about its current limitations whereas possible ways of solving these limitations are listed as future work.

Figure 60 – Results for the **1-D Periodic Corridor scenario**. The image compares the fundamental diagram (relation between density and speed) of the three tested models and real data.



Source: the author.

Figure 61 – Comparison of the results for the **Circle scenario** with symmetric (top) and noisy (bottom) initial positions. The goal of the agents is to reach the diametrically opposed position. The agents color encodes the speed: dark blue means the agent is stopped or moving slower than its comfort speed; light green means the agent is moving at its comfort speed; and red means the agent is moving faster than its comfort speed. Results are shown for the proposed model, OSV and RVO2.

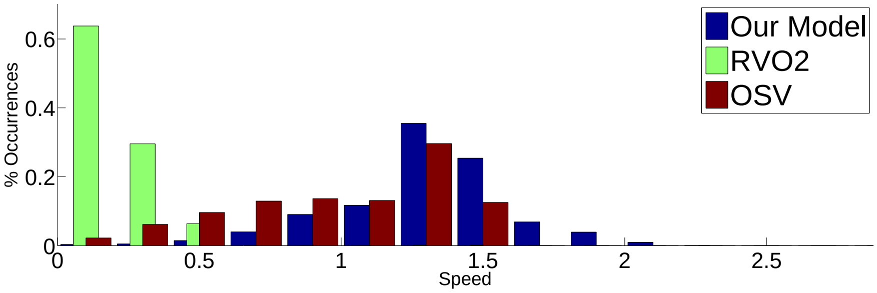


(a) Initial configuration (b) Proposed model (c) OSV (d) RVO2

(e) Initial configuration (f) Proposed model (g) OSV (h) RVO2

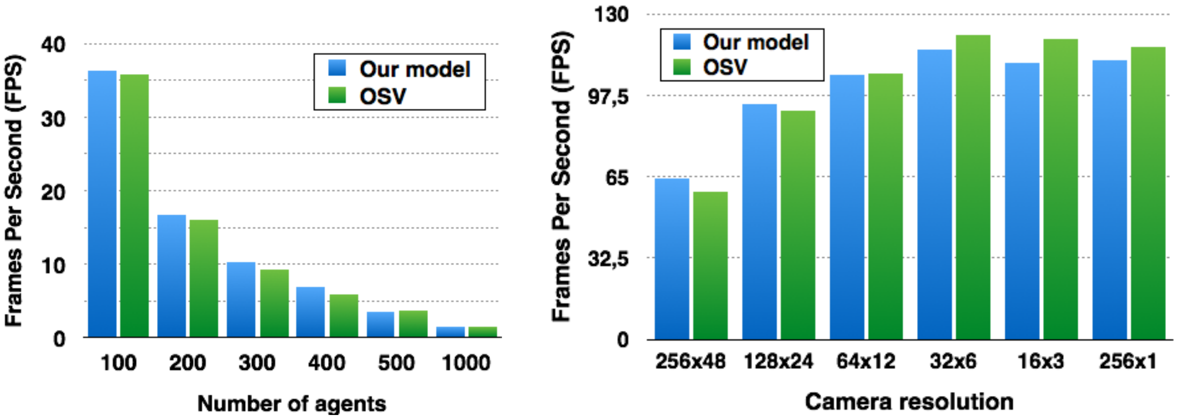
Source: the author.

Figure 62 – Histograms showing the distribution of the speed in the **Circle scenario** for the three tested algorithms: the proposed model, OSV, RVO2.



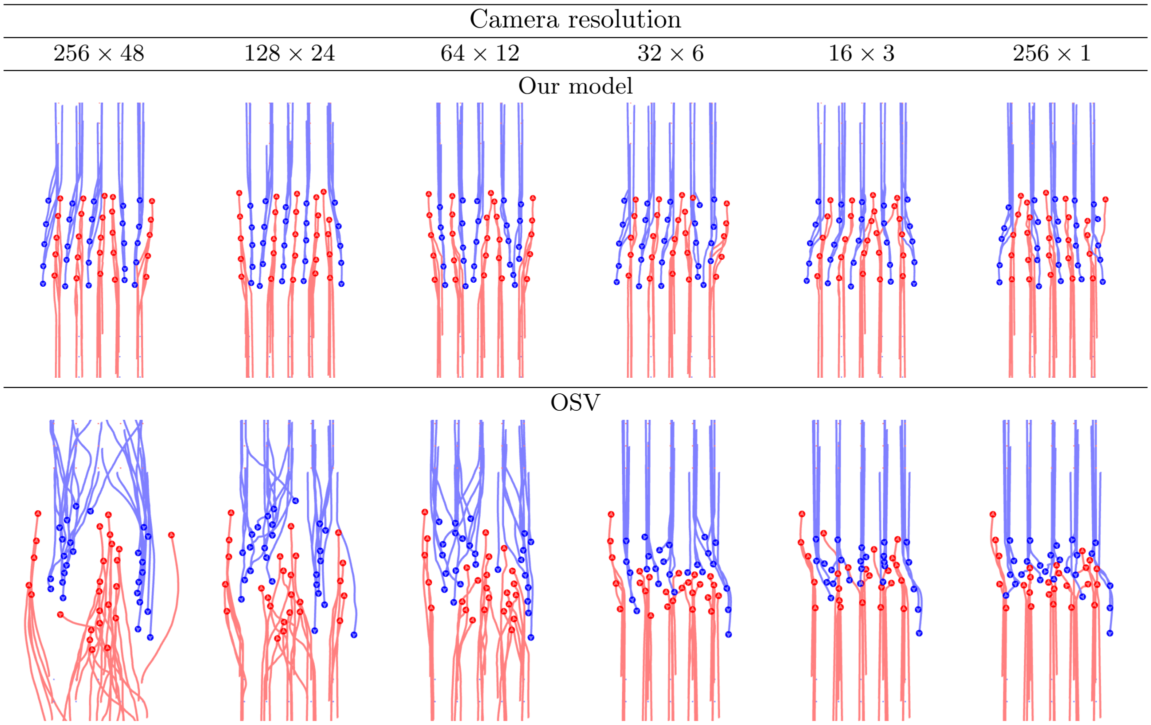
Source: the author.

Figure 63 – Performance comparison for the **Opposite scenario**. **Left**: different number of agents. **Right**: different camera resolutions (50 agents).



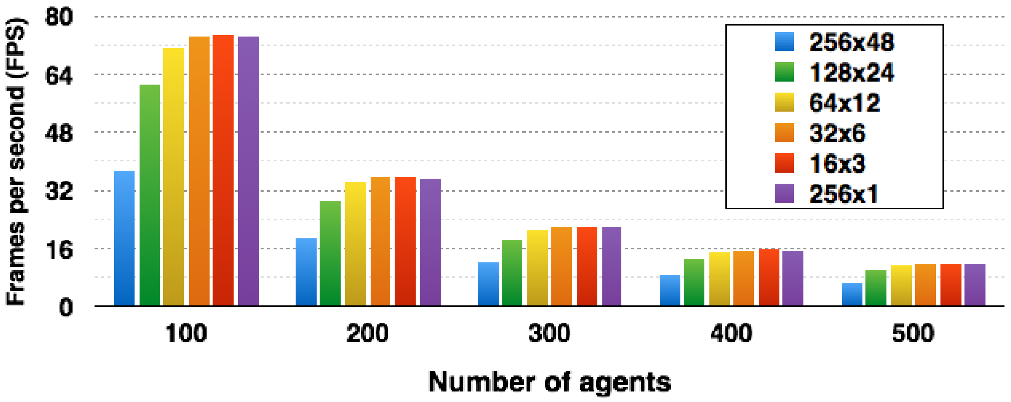
Source: the author.

Figure 64 – Impact of the camera resolution for our model and OSV. In our model the obstacles perception and anticipation are only slightly affected, even using a 1-D camera resolution.



Source: the author.

Figure 65 – Performance for the **Opposite scenario** varying the number of agents and the camera resolution.



Source: the author.

6 Conclusion

In this thesis, a new synthetic vision-based simulation algorithm is presented. The main contribution of this work is a new steering scheme based on:

a) A cost function which evaluates the agents' situation w.r.t. their target and risk of collision; and

b) A locally optimal gradient-descent scheme.

Vision-based approaches are interesting for many reasons and a particular motivation is to link with the Neuroscience community (e.g., to decipher visual guidance of the human locomotion in crowds). The results presented here show that high-quality trajectories can be generated and entertainment applications can be aimed too. In particular, this new method prevents some known artifacts generated by previous ones as demonstrated in the previous chapter. The contribution of this work is not a simple revisiting of vision-based algorithms, it makes this method more robust, especially with its new mathematical foundations. The new control scheme is more easily extendable to new behaviors and types of interactions between agents by defining some corresponding cost functions. In addition, the proposed model has moved from a purely reactive to a locally optimal steering of agents in complex situations mixing static and dynamic obstacles.

The simulations with the proposed model show a more human-like behavior than the other models. This opinion is based on the fact that this model is able to more closely follow the experimental data than the other models (see Section 5.3). Moreover, the model produces agent behaviors which are commonly observed in the day-to-day life, but which the other models cannot reproduce, such as accelerating to avoid collisions. Finally, the presented histograms indicate that, in the proposed model, agents keep a more natural distance between them, which reduces the strong and unrealistic patterns produced by other models.

The quality of the results can be partially explained by the use of perceptual variables to answer the two questions humans put to achieve a collision-free navigation (CUTTING et al., 1995):

1. Is a collision going to occur? (dca)
2. If so, when is it going to occur? ($ttca$)

Furthermore, the use of a control loop which tries to emulate the information flow in real humans locomotion also helps explaining these results.

Limitations

Despite the good results obtained, some limitations of this approach can be pointed out. Firstly, we need to look for a way of evaluating how closely is the synthetic perception/action loop to the real human perception/action loop. However, the definition of the real human perception/action loop in crowds is still an open problem and thus there are no available metrics to perform such an evaluation. Probably some cooperation between the fields of Crowd Simulation and Cognitive Science would be required to cope with this problem. Secondly, although the parameters' setting has been kept intuitive and the number of parameters has been kept low, a single parameter setting which fits all types of situations could not be found at this stage.

Some parameters had to be manually adjusted for different contexts. A typical example is the anticipation time which must be reduced when density increases. Nevertheless, several examples of the same kind could be generated with the same parameter set (e.g., *Opposite*, *Columns* and *Crossing*). Note that this statement is true for most advanced simulation algorithms. Thirdly, the proposed cost function is limited to collision avoidance and goal reaching behaviors. Finally, this method is not able to deal with complex navigation in large environments and should be integrated to a global motion planner.

Future work

Future work aims at directly tackling those limitations. Regarding the variation of parameters with different scenarios, the focus of future research should be put on identifying the current situation and selecting the appropriate parameters setting, rather than trying to find a single parameters setting which would work well in all situations (and which might not exist). To this end, one could resort to machine learning and image processing techniques, so as to classify the situations and learn their respective parameters based on features of the visual flow. Alternative frameworks for parameter optimization can also be considered, such as *SteerFit* (BERSETH et al., 2014).

To extend the proposed method with a richer set of behaviors, as well as to handle larger sets of contexts such as high density ones, different cost functions can be defined and combined. As an example, instead of *ttca* and *dca*, a cost function for ‘high-density’ contexts could be based on the pressure agents perceive from other agents in contact with them, or the simple distance they keep to the others. Another example is interactions with more dynamic obstacles like cars, for which a notion of collision energy could be added to avoid more carefully the most threatening obstacles. With such future extensions, it is expected that the idea of perceptual models for crowd simulation can be more generally developed and the relevance of the subtleties of human behaviors in such contexts can be provided.

APPENDIX A – Gradient of the cost function

Computing the gradient

$$\nabla C_t = \begin{bmatrix} \frac{\partial C_t}{\partial s_a} & \frac{\partial C_t}{\partial \theta_a} \end{bmatrix} \tag{A.1}$$

implies computing the partial derivatives given by:

$$\frac{\partial C_t}{\partial s_a} = \frac{\partial C_m}{\partial s_a} + \frac{\partial C_o}{\partial s_a} \tag{A.2}$$

and

$$\frac{\partial C_t}{\partial \theta_a} = \frac{\partial C_m}{\partial \theta_a} + \frac{\partial C_o}{\partial \theta_a}, \tag{A.3}$$

where

$$\frac{\partial C_o}{\partial s_a} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial C_{o_i,a}}{\partial s_a} \tag{A.4}$$

and

$$\frac{\partial C_o}{\partial \theta_a} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial C_{o_i,a}}{\partial \theta_a} \ . \tag{A.5}$$

The values of the partial derivatives of C_m (see Eqs. (A.2) and (A.3)) are given by:

$$\frac{\partial C_m}{\partial s_a} = \frac{\Delta_s}{2\sigma_s^2} \exp\left(-\frac{1}{2}\left(\frac{\Delta_s}{\sigma_s}\right)^2\right) \tag{A.6}$$

and

$$\frac{\partial C_m}{\partial \theta_a} = -\frac{\alpha_g}{2\sigma_{\alpha_g}^2} \exp\left(-\frac{1}{2}\left(\frac{\alpha_g}{\sigma_{\alpha_g}}\right)^2\right) \ . \tag{A.7}$$

To determine the values of the partial derivatives of C_o (see Eqs. (A.4) and (A.5)) the following quantities must be computed:

$$\frac{\partial C_{o_i,a}}{\partial s_a} = -C_{o_i,a}\left(\frac{\partial ttca_{o_i,a}}{\partial s_a}\frac{ttca_{o_i,a}}{\sigma_{ttca}^2} + \frac{\partial dca_{o_i,a}}{\partial s_a}\frac{dca_{o_i,a}}{\sigma_{dca}^2}\right) \tag{A.8}$$

and

$$\frac{\partial C_{o_i,a}}{\partial \theta_a} = -C_{o_i,a}\left(\frac{\partial ttca_{o_i,a}}{\partial \theta_a}\frac{ttca_{o_i,a}}{\sigma_{ttca}^2} + \frac{\partial dca_{o_i,a}}{\partial \theta_a}\frac{dca_{o_i,a}}{\sigma_{dca}^2}\right) \ . \tag{A.9}$$

In the next two sections of this appendix, we show how to compute the partial derivatives of $ttca_{o_i,a}$ and $dca_{o_i,a}$ required to evaluate Eqs. (A.8) and (A.9).

APPENDIX B – Partial derivatives of $\mathbf{ttca}_{o_i,a}$

Let us assume that $\mathbf{v}_{o_i|a} \neq (0,0)$ in which case the $ttca_{o_i,a}$ is given by:

$$ttca_{o_i,a} = -\frac{f}{g} \tag{B.1}$$

where

$$f = \mathbf{p}_{o_i|a} \cdot \mathbf{v}_{o_i|a} \tag{B.2}$$

$$g = \mathbf{v}_{o_i|a} \cdot \mathbf{v}_{o_i|a} \ . \tag{B.3}$$

The partial derivative of $ttca$ with respect to a hypothetical argument x is thus:

$$\frac{\partial ttca_{o_i,a}}{\partial x} = -\frac{\frac{\partial f}{\partial x}}{g} + \frac{\frac{\partial g}{\partial x} f}{g^2} \ . \tag{B.4}$$

Let us recall that:

$$\mathbf{v}_{o_i|a} = (v_{xo_i} - s_a \cos\theta_a,\, v_{yo_i} - s_a \sin\theta_a) \ . \tag{B.5}$$

Let us also notice the following equalities:

$$\frac{\partial\, \mathbf{v}_{o_i|a}}{\partial \theta_a} = s_a\,(\sin\theta_a, -\cos\theta_a) = (v_{ya}, -v_{xa}) \tag{B.6}$$

and

$$\frac{\partial\, \mathbf{v}_{o_i|a}}{\partial s_a} = -(\cos\theta_a, \sin\theta_a) = -\hat{\mathbf{v}}_a \ . \tag{B.7}$$

To compute $d\frac{\partial ttca_{o_i,a}}{\partial \theta_a}$ and $d\frac{\partial ttca_{o_i,a}}{\partial s_a}$, we need thus to derive $\frac{\partial f}{\partial \theta_a}$, $\frac{\partial f}{\partial s_a}$, $\frac{\partial g}{\partial \theta_a}$ and $\frac{\partial g}{\partial s_a}$. Follow the two partial derivatives of f:

$$\begin{aligned}\frac{\partial f}{\partial \theta_a} = \frac{\partial\left(\mathbf{p}_{o_i|a} \cdot \mathbf{v}_{o_i|a}\right)}{\partial \theta_a} &= \mathbf{p}_{o_i|a} \cdot \frac{\partial\, \mathbf{v}_{o_i|a}}{\partial \theta_a} \\ &= \mathbf{p}_{o_i|a} \cdot (v_{ya}, -v_{xa})\end{aligned} \tag{B.8}$$

and

$$\begin{aligned}\frac{\partial f}{\partial s_a} = \frac{\partial\left(\mathbf{p}_{o_i|a} \cdot \mathbf{v}_{o_i|a}\right)}{\partial s_a} &= \mathbf{p}_{o_i|a} \cdot \frac{\partial\, \mathbf{v}_{o_i|a}}{\partial s_a} \\ &= -\mathbf{p}_{o_i|a} \cdot \hat{\mathbf{v}}_a \ .\end{aligned} \tag{B.9}$$

The partial derivatives of g are given by:

$$\begin{aligned}\frac{\partial g}{\partial \theta_a} &= \frac{\partial\left(\mathbf{v}_{o_i|a} \cdot \mathbf{v}_{o_i|a}\right)}{\partial \theta_a} \\ &= \frac{\partial \mathbf{v}_{o_i|a}}{\partial \theta_a} \cdot \mathbf{v}_{o_i|a} + \mathbf{v}_{o_i|a} \cdot \frac{\partial \mathbf{v}_{o_i|a}}{\partial \theta_a} \\ &= (v_{ya}, -v_{xa}) \cdot \mathbf{v}_{o_i|a} + \mathbf{v}_{o_i|a} \cdot (v_{ya}, -v_{xa}) \\ &= 2\,(v_{ya}, -v_{xa}) \cdot \mathbf{v}_{o_i|a}\end{aligned} \tag{B.10}$$

and

$$\begin{aligned}\frac{\partial g}{\partial s_a} &= \frac{\partial\left(\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}\right)}{\partial s_a}\\ &= \frac{\partial\mathbf{v}_{o_i|a}}{\partial s_a}\cdot\mathbf{v}_{o_i|a}+\mathbf{v}_{o_i|a}\cdot\frac{\partial\mathbf{v}_{o_i|a}}{\partial s_a}\\ &= -\hat{\mathbf{v}}_a\cdot\mathbf{v}_{o_i|a}-\mathbf{v}_{o_i|a}\cdot\hat{\mathbf{v}}_a\\ &= -2\,\hat{\mathbf{v}}_a\cdot\mathbf{v}_{o_i|a}\ . \end{aligned} \tag{B.11}$$

Using Eqs. (B.4), (B.8) and (B.10) we can compute:

$$\begin{aligned}\frac{\partial ttca_{o_i,a}}{\theta_a} &= -\frac{\mathbf{p}_{o_i|a}\cdot(v_{ya},-v_{xa})}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}+\frac{\left(2\,(v_{ya},-v_{xa})\cdot\mathbf{v}_{o_i|a}\right)\left(\mathbf{p}_{o_i|a}\cdot\mathbf{v}_{o_i|a}\right)}{\left(\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}\right)^2}\\ &= -\frac{\mathbf{p}_{o_i|a}\cdot(v_{ya},-v_{xa})}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}+\frac{2\,ttca_{o_i,a}\,\mathbf{v}_{o_i|a}\cdot(v_{ya},-v_{xa})}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}\\ &= -\frac{\left(\mathbf{p}_{o_i|a}+2\,ttca_{o_i,a}\,\mathbf{v}_{o_i|a}\right)\cdot(v_{ya},-v_{xa})}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}\ . \end{aligned} \tag{B.12}$$

Using Eqs. (B.4), (B.9) and (B.11) we can compute:

$$\begin{aligned}\frac{\partial ttca_{o_i,a}}{s_a} &= \frac{\mathbf{p}_{o_i|a}\cdot\hat{\mathbf{v}}_a}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}-\frac{2\left(\hat{\mathbf{v}}_a\cdot\mathbf{v}_{o_i|a}\right)\left(\mathbf{p}_{o_i|a}\cdot\mathbf{v}_{o_i|a}\right)}{\left(\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}\right)^2}\\ &= \frac{\left(\mathbf{p}_{o_i|a}\cdot\hat{\mathbf{v}}_a\right)+\left(2\,ttca_{o_i,a}\,\mathbf{v}_{o_i|a}\cdot\hat{\mathbf{v}}_a\right)}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}\\ &= \frac{\left(\mathbf{p}_{o_i|a}+2\,ttca_{o_i,a}\,\mathbf{v}_{o_i|a}\right)\cdot\hat{\mathbf{v}}_a}{\mathbf{v}_{o_i|a}\cdot\mathbf{v}_{o_i|a}}\ . \end{aligned} \tag{B.13}$$

APPENDIX C – Partial derivatives of $\mathbf{dca}_{o_i,a}$

Let us recall the expression of $dca_{o_i,a}$:

$$\begin{aligned} dca_{o_i,a} &= \|\mathbf{dca}_{o_i,a}\| = \|\mathbf{p}_{o_i|a} + ttca_{o_i,a}\mathbf{v}_{o_i|a}\| \\ &= \left(\mathbf{dca}_{o_i,a} \cdot \mathbf{dca}_{o_i,a}\right)^{1/2} \ . \end{aligned} \tag{C.1}$$

The partial derivative of $dca_{o_i,a}$ with respect to a hypothetical argument x is given by:

$$\begin{aligned} \frac{\partial dca_{o_i,a}}{\partial x} &= \frac{1}{2}\left(\mathbf{dca}_{o_i,a} \cdot \mathbf{dca}_{o_i,a}\right)^{-1/2} \frac{\partial\left(\mathbf{dca}_{o_i,a} \cdot \mathbf{dca}_{o_i,a}\right)}{\partial x} \\ &= \frac{1}{2\, dca_{o_i,a}} \frac{\partial\left(\mathbf{dca}_{o_i,a} \cdot \mathbf{dca}_{o_i,a}\right)}{\partial x} \\ &= \frac{1}{2\, dca_{o_i,a}} \frac{\partial\left(\|\mathbf{dca}_{o_i,a}\|^2\right)}{\partial x} \ . \end{aligned} \tag{C.2}$$

The rightmost term of Eq. (C.2) can be developed as:

$$\begin{aligned} \frac{\partial\left(\|\mathbf{dca}_{o_i,a}\|^2\right)}{\partial x} &= \frac{\partial \mathbf{dca}_{o_i,a}}{\partial x} \cdot \mathbf{dca}_{o_i,a} + \frac{\partial \mathbf{dca}_{o_i,a}}{\partial x} \cdot \mathbf{dca}_{o_i,a} \\ &= 2\,\frac{\partial \mathbf{dca}_{o_i,a}}{\partial x} \cdot \mathbf{dca}_{o_i,a} \\ &= 2\,\frac{\partial\left(\mathbf{p}_{o_i|a} + ttca_{o_i,a}\mathbf{v}_{o_i|a}\right)}{\partial x} \cdot \mathbf{dca}_{o_i,a} \ . \end{aligned} \tag{C.3}$$

given that $\mathbf{p}_{o_i|a}$ is a constant, this equation can be simplified as:

$$\frac{\partial\left(\|\mathbf{dca}_{o_i,a}\|^2\right)}{\partial x} = 2\left(\frac{\partial(ttca_{o_i,a}\mathbf{v}_{o_i|a})}{\partial x}\right) \cdot \mathbf{dca}_{o_i,a} \tag{C.4}$$

where

$$\frac{\partial(ttca_{o_i,a}\mathbf{v}_{o_i|a})}{\partial x} = \frac{\partial ttca_{o_i,a}}{\partial x}\mathbf{v}_{o_i|a} + ttca_{o_i,a}\frac{\partial \mathbf{v}_{o_i|a}}{\partial x} \ . \tag{C.5}$$

To compute the partial derivative of $dca_{o_i,a}$ with respect to θ_a, we can use the results from Eqs. (B.6), (B.12), (C.2) and (C.4), yielding:

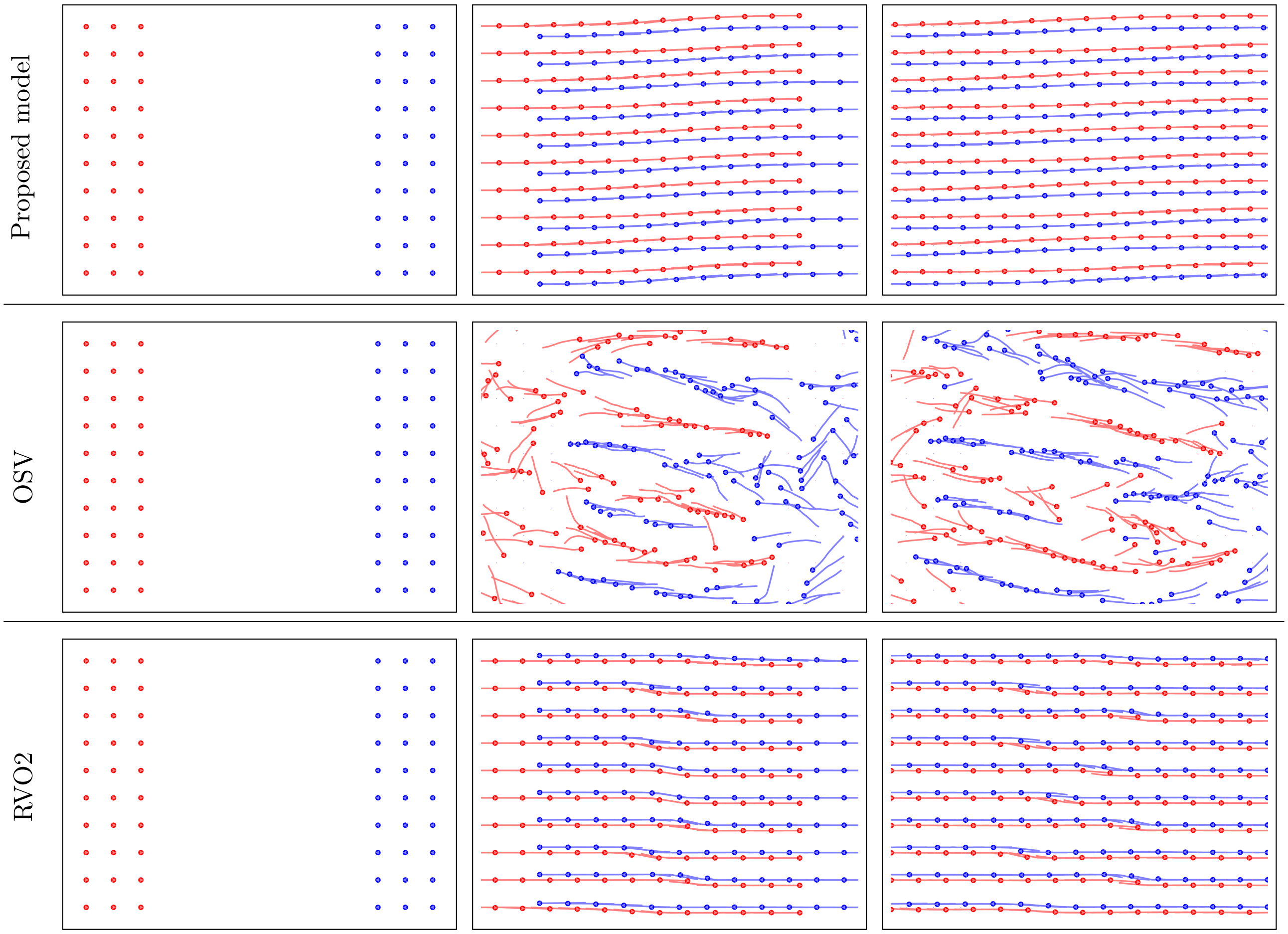
$$\begin{aligned} \frac{\partial dca_{o_i,a}}{\partial \theta_a} &= \left(-\frac{\left(\mathbf{p}_{o_i|a} + 2\, ttca_{o_i,a}\, \mathbf{v}_{o_i|a}\right) \cdot (v_{ya}, -v_{xa})}{\mathbf{v}_{o_i|a} \cdot \mathbf{v}_{o_i|a}}\mathbf{v}_{o_i|a} + ttca_{o_i,a}(v_{ya}, -v_{xa})\right) \cdot \mathbf{dca}_{o_i,a}\frac{1}{dca_{o_i,a}} \qquad \text{(C.6)} \\ &= \frac{\mathbf{dca}_{o_i,a} \cdot \left(\frac{\partial ttca_{o_i,a}}{\partial \theta_a}\mathbf{v}_{o_i|a} + ttca_{o_i,a}\,(v_{ya}, -v_{xa})\right)}{dca_{o_i,a}} \ . \end{aligned}$$

Similarly, to compute the partial derivative of $dca_{o_i,a}$ with respect to s_a, we can use the results from Eqs. (B.7), (B.13), (C.2) and (C.4).

$$\begin{aligned} \frac{\partial dca_{o_i,a}}{\partial s_a} &= \left(\frac{\left(\mathbf{p}_{o_i|a} + 2\, ttca_{o_i,a}\, \mathbf{v}_{o_i|a}\right) \cdot (v_{ya}, -v_{xa})}{\mathbf{v}_{o_i|a} \cdot \mathbf{v}_{o_i|a}}\mathbf{v}_{o_i|a} - ttca_{o_i,a}\hat{\mathbf{v}}_a\right) \cdot \mathbf{dca}_{o_i,a}\frac{1}{dca_{o_i,a}} \\ &= \frac{\mathbf{dca}_{o_i,a} \cdot \left(\frac{\partial ttca_{o_i,a}}{\partial s}\mathbf{v}_{o_i|a} - ttca_{o_i,a}\hat{\mathbf{v}}_a\right)}{dca_{o_i,a}} \ . \end{aligned} \tag{C.7}$$

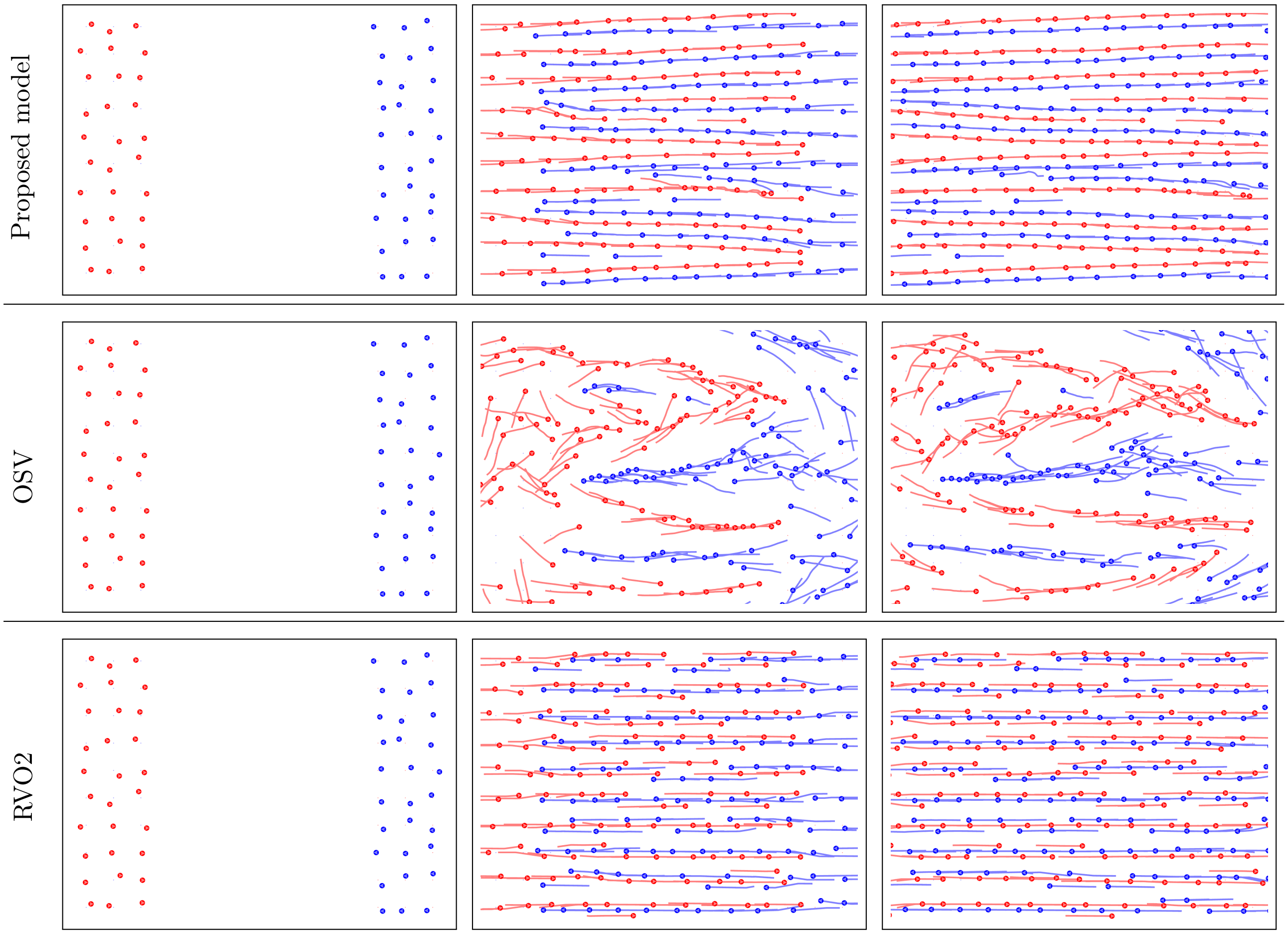
APPENDIX D – Additional examples

Figure 66 – Comparison of the results for the **Opposite scenario** with **many agents** and **structured initial positions**. The two groups of agents (red and blue) have as goal to switch positions. Results are shown for the proposed model, OSV and RVO2.



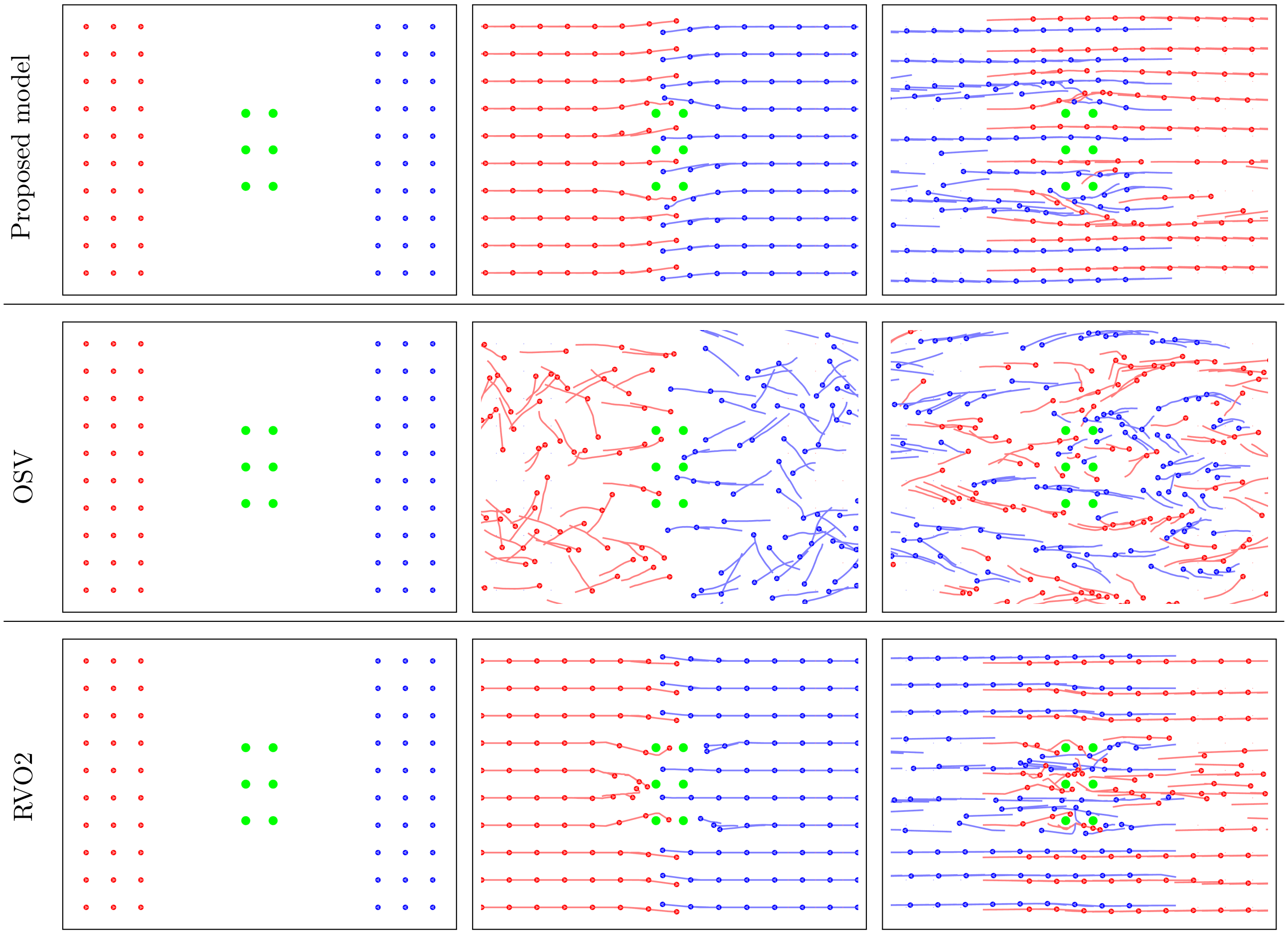
Source: the author.

Figure 67 – Comparison of the results for the **Opposite scenario** with **many agents** and **noisy initial positions**. The two groups of agents (red and blue) have as goal to switch positions. Results are shown for the proposed model, OSV and RVO2.



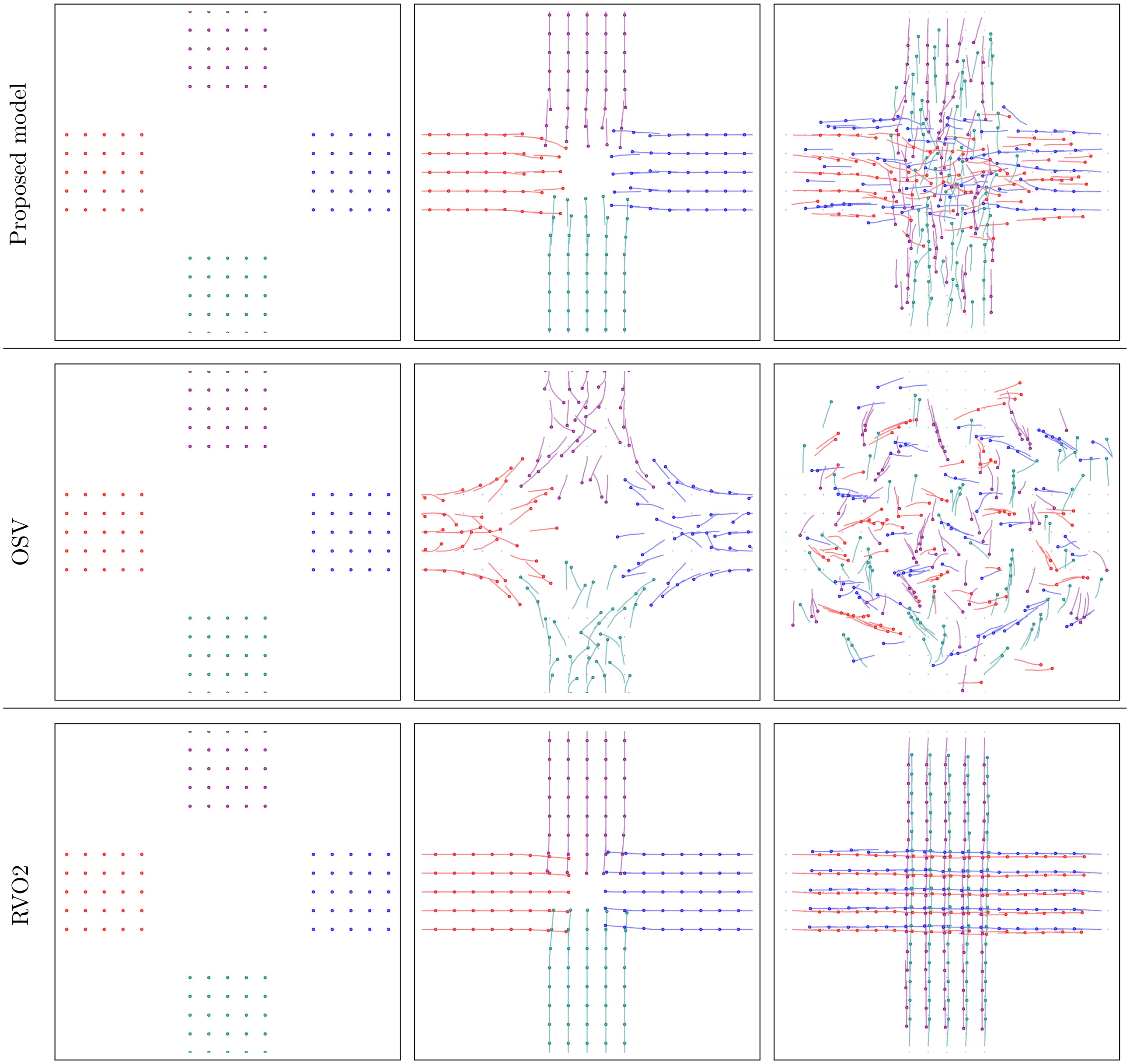
Source: the author.

Figure 68 – Comparison of the results for the **Columns scenario** with **many agents**. The two groups of agents (red and blue) have as goal to switch positions. Results are shown for the proposed model, OSV and RVO2.



Source: the author.

Figure 69 – Comparison of the results for the **Crossing scenario** with **many agents** in four groups. The four groups of agents have as goal to reach the opposite positions. Results are shown for the proposed model, OSV and RVO2.



Source: the author.

Bibliography

ALI, S.; NISHINO, K.; MANOCHA, D.; SHAH, M. In: *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective.* [S.l.]: Springer New York, 2013. ISBN 978-1-4614-8482-0.

AUTODESK. *"Autodesk 3ds Max"*. 2015. Http://www.autodesk.com.br/products/3ds-max . Last visited on 26/01/2015.

AUTODESK. *"Autodesk Maya"*. 2015. Http://www.autodesk.com.br/products/maya . Last visited on 26/01/2015.

BANDI, S.; THALMANN, D. Space discretization for efficient human navigation. *Computer Graphics Forum*, Blackwell Publishers Ltd, v. 17, n. 3, p. 195–206, 1998. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/1467-8659.00267>.

BARZILAI, J.; BORWEIN, J. M. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, v. 8, n. 1, p. 141–148, 1988. Disponível em: <http://imajna.oxfordjournals.org/content/8/1/141.abstract>.

BAYAZIT, O.; LIEN, J.-M.; AMATO, N. Roadmap-based flocking for complex environments. In: *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on.* [S.l.: s.n.], 2002. p. 104–113.

BAYAZIT, O. B.; LIEN, J.-M.; AMATO, N. M. Better group behaviors in complex environments using global roadmaps. In: *Proceedings of the eighth international conference on Artificial life.* Cambridge, MA, USA: MIT Press, 2003. (ICAL 2003), p. 362–370. ISBN 0-262-69281-3. Disponível em: <http://dl.acm.org/citation.cfm?id=860295.860352>.

BERG, J. van den; GUY, S.; LIN, M.; MANOCHA, D. Reciprocal n-body collision avoidance. In: *Robotics Research.* [S.l.]: Springer, 2011. v. 70, p. 3–19.

BERG, J. van den; LIN, M.; MANOCHA, D. Reciprocal velocity obstacles for real-time multi-agent navigation. In: *IEEE International Conference on Robotics and Automation.* [S.l.: s.n.], 2008. p. 1928–1935. ISSN 1050-4729.

BERG, J. van den; PATIL, S.; SEWALL, J.; MANOCHA, D.; LIN, M. Interactive navigation of multiple agents in crowded environments. In: *Proceedings of the 2008 symposium on Interactive 3D graphics and games.* New York, NY, USA: ACM, 2008. (I3D '08), p. 139–147. ISBN 978-1-59593-983-8.

BERSETH, G.; KAPADIA, M.; HAWORTH, B.; FALOUTSOS, P. SteerFit: Automated Parameter Fitting for Steering Algorithms. In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* New York, NY, USA: ACM, 2014. (SCA '14).

BICHO, A. de L.; RODRIGUES, R. A.; MUSSE, S. R.; JUNG, C. R.; PARAVISI, M.; MAGALHãES, L. P. Simulating crowds based on a space colonization algorithm. *Computers & Graphics*, v. 36, n. 2, p. 70 – 79, 2012. ISSN 0097-8493.

BLENDER. *"Blender"*. 2015. Http://www.blender.org/ . Last visited on 26/01/2015.

BRAUN, A.; MUSSE, S.; OLIVEIRA, L. de; BODMANN, B. Modeling individual behaviors in crowd simulation. In: *Computer Animation and Social Agents, 2003. 16th International Conference on*. [S.l.: s.n.], 2003. p. 143–148. ISSN 1087-4844.

BRUNEAU, J.; DUTRA, T. B.; PETTRÉ, J. Following behaviors: A model for computing following distances. *Transportation Research Procedia*, v. 2, n. 0, p. 424 – 429, 2014. ISSN 2352-1465. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2352146514000854>.

BRUNEAU, J.; DUTRA, T. B.; PETTRÉ, J. Following behaviors: A model for computing following distances based on prediction. In: *Proceedings of the Seventh International Conference on Motion in Games*. New York, NY, USA: ACM, 2014. (MIG '14), p. 17–24. ISBN 978-1-4503-2623-0. Disponível em: <http://doi.acm.org/10.1145/2668064.2668085>.

BURSTEDDE, C.; KLAUCK, K.; SCHADSCHNEIDER, A.; ZITTARTZ, J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A Statistical Mechanics and its Applications*, v. 295, p. 507–525, jun. 2001.

CAUCHY, A. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, v. 25, n. 1847, p. 536–538, 1847.

CHARALAMBOUS, P.; CHRYSANTHOU, Y. The pag crowd: A graph based approach for efficient data-driven crowd simulation. *Computer Graphics Forum*, v. 33, n. 8, p. 95–108, 2014. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/cgf.12403>.

CHARALAMBOUS, P.; KARAMOUZAS, I.; GUY, S. J.; CHRYSANTHOU, Y. A data-driven framework for visual crowd analysis. *Computer Graphics Forum*, v. 33, n. 7, p. 41–50, 2014. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/cgf.12472>.

CHENNEY, S. Flow tiles. In: *Proc. of the 2004 ACM SIGGRAPH/Eurographics symp. on Computer animation*. [S.l.: s.n.], 2004. p. 233–242. ISBN 3-905673-14-2.

CIVIDINI, J.; APPERT-ROLLAND, C.; HILHORST, H. J. Diagonal patterns and chevron effect in intersecting traffic flows. *EPL (Europhysics Letters)*, v. 102, n. 2, p. 20002, 2013. Disponível em: <http://stacks.iop.org/0295-5075/102/i=2/a=20002>.

COURTY, N.; MUSSE, S. "Simulation of large crowds in emergency situations including gaseous phenomena". In: *Computer Graphics International 2005*. [S.l.: s.n.], 2005. p. 206 – 212. ISSN 1530-1052.

CUTTING, J.; VISHTON, P.; BRAREN, P. How we avoid collisions with stationary and moving objects. *Psychological Review*, US: American Psychological Association, v. 102, n. 4, p. 627–651, 1995. ISSN 1939-1471.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, p. 269–271, 1959.

DIJKSTRA, J.; TIMMERMANS, H.; JESSURUN, A. A multi-agent cellular automata system for visualising simulated pedestrian activity. In: BANDINI, S.; WORSCH, T. (Ed.). *Theory and Practical Issues on Cellular Automata*. [S.l.]: Springer London, 2001. p. 29–36. ISBN 978-1-85233-388-1.

DUTRA, T.; CAVALCANTE-NETO, J.; VIDAL, C.; MUSSE, S. A multipotential field model for crowds with scalable behaviors. In: *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on*. [S.l.: s.n.], 2013. p. 31–38. ISSN 1530-1834.

FIORINI, P.; SHILLER, Z. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, SAGE Publications, v. 17, n. 7, p. 760–772, 1998.

GAYLE, R.; SUD, A.; ANDERSEN, E.; GUY, S.; LIN, M.; MANOCHA, D. Interactive navigation of heterogeneous agents using adaptive roadmaps. *Visualization and Computer Graphics, IEEE Transactions on*, v. 15, n. 1, p. 34–48, 2009. ISSN 1077-2626.

GERAERTS, R.; OVERMARS, M. H. The corridor map method: a general framework for real-time high-quality path planning. *Computer Animation and Virtual Worlds*, John Wiley & Sons, Ltd., v. 18, n. 2, p. 107–119, 2007. ISSN 1546-427X. Disponível em: <http://dx.doi.org/10.1002/cav.166>.

GIBSON, J. J. Visually controlled locomotion and visual orientation in animals*. *British Journal of Psychology*, Blackwell Publishing Ltd, v. 49, n. 3, p. 182–194, 1958. ISSN 2044-8295. Disponível em: <http://dx.doi.org/10.1111/j.2044-8295.1958.tb00656.x>.

GOLAEM. *"Golaem Crowd"*. 2015. Http://www.golaem.com/ . Last visited on 26/01/2015.

GOLAS, A.; NARAIN, R.; CURTIS, S.; LIN, M. Hybrid long-range collision avoidance for crowd simulation. *Visualization and Computer Graphics, IEEE Transactions on*, v. 20, n. 7, p. 1022–1034, July 2014. ISSN 1077-2626.

GRÖSCHEL, A. *Towards Believable Crowd Simulation for Interactive Real-Time Applications*. Tese (Doutorado) — HTW Berlin, jun. 2011.

GUY, S. J.; BERG, J. van den; LIU, W.; LAU, R.; LIN, M. C.; MANOCHA, D. A statistical similarity measure for aggregate crowd dynamics. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 31, n. 6, p. 190:1–190:11, nov. 2012. ISSN 0730-0301. Disponível em: <http://doi.acm.org/10.1145/2366145.2366209>.

GUY, S. J.; CHHUGANI, J.; KIM, C.; SATISH, N.; LIN, M.; MANOCHA, D.; DUBEY, P. Clearpath: Highly parallel collision avoidance for multi-agent simulation. In: *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. New York, NY, USA: ACM, 2009. (SCA '09), p. 177–187. ISBN 978-1-60558-610-6. Disponível em: <http://doi.acm.org/10.1145/1599470.1599494>.

GUY, S. J.; KIM, S.; LIN, M. C.; MANOCHA, D. Simulating heterogeneous crowd behaviors using personality trait theory. In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: ACM, 2011. (SCA '11), p. 43–52. ISBN 978-1-4503-0923-3.

HART, P.; NILSSON, N.; RAPHAEL, B. "A formal basis for the heuristic determination of minimum cost paths". *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 2, p. 100–107, 1968. ISSN 0536-1567.

HAUMONT, D.; DEBEIR, O.; SILLION, F. Volumetric cell-and-portal generation. *Computer Graphics Forum*, Blackwell Publishing, Inc, v. 22, n. 3, p. 303–312, 2003. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/1467-8659.00677>.

HEIGEAS, L.; LUCIANI, A.; THOLLOT, J.; CASTAGNÉ, N. A Physically-Based Particle Model of Emergent Crowd Behaviors. In: *Graphicon*. [S.l.: s.n.], 2003.

HELBING, D.; FARKAS, I.; VICSEK, T. Simulating dynamical features of escape panic. *Nature*, [London: Macmillan Journals], 1869-, v. 407, n. 6803, p. 487–490, 2000.

HELBING, D.; MOLNÁR, P. Social force model for pedestrian dynamics. *Physical Review E*, American Physical Society, v. 51, n. 5, p. 4282–4286, 1995.

HOFF III, K. E.; KEYSER, J.; LIN, M.; MANOCHA, D.; CULVER, T. Fast computation of generalized voronoi diagrams using graphics hardware. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999. (SIGGRAPH '99), p. 277–286. ISBN 0-201-48560-5. Disponível em: <http://dx.doi.org/10.1145-/311535.311567>.

HUGHES, R. L. "A continuum theory for the flow of pedestrians". *Transportation Research Part B: Methodological*, v. 36, n. 6, p. 507 – 535, 2002. ISSN 0191-2615.

HUGHES, R. L. "The flow of human crowds". *Annual Review of Fluid Mechanics*, v. 35, n. 1, p. 169–182, 2003.

JELIĆ, A.; APPERT-ROLLAND, C.; LEMERCIER, S.; PETTRÉ, J. Properties of pedestrians walking in line: Fundamental diagrams. *Phys. Rev. E*, American Physical Society, v. 85, p. 036111, Mar 2012.

JIANG, H.; XU, W.; MAO, T.; LI, C.; XIA, S.; WANG, Z. "Continuum crowd simulation in complex environments". *Computers & Graphics*, v. 34, n. 5, p. 537 – 544, 2010. ISSN 0097-8493.

JORDAO, K.; PETTRÉ, J.; CHRISTIE, M.; CANI, M.-P. Crowd sculpting: A space-time sculpting method for populating virtual environments. *Computer Graphics Forum*, v. 33, n. 2, p. 351–360, 2014. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/cgf.12316>.

JU, E.; CHOI, M.; PARK, M.; LEE, J.; LEE, K.; TAKAHASHI, S. Morphable crowds. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 29, p. 140:1–140:10, 2010. ISSN 0730-0301.

KALLMANN, M. Shortest paths with arbitrary clearance from navigation meshes. In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2010. (SCA '10), p. 159–168. Disponível em: <http://dl.acm.org/citation.cfm?id=1921427.1921451>.

KAPADIA, M.; BADLER, N. I. Navigation and steering for autonomous virtual humans. *Wiley Interdisciplinary Reviews: Cognitive Science*, John Wiley & Sons, Inc., v. 4, n. 3, p. 263–272, 2013. ISSN 1939-5086. Disponível em: <http://dx.doi.org/10.1002/wcs.1223>.

KAPADIA, M.; SINGH, S.; ALLEN, B.; REINMAN, G.; FALOUTSOS, P. Steerbug: An interactive framework for specifying and detecting steering behaviors. In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: ACM, 2009. (SCA '09), p. 209–216. ISBN 978-1-60558-610-6. Disponível em: <http://doi.acm.org/10.1145/1599470.1599497>.

KAPADIA, M.; SINGH, S.; HEWLETT, W.; FALOUTSOS, P. Egocentric affordance fields in pedestrian steering. In: *Proceedings of the 2009 symposium on Interactive 3D graphics and games*. New York, NY, USA: ACM, 2009. (I3D '09), p. 215–223. ISBN 978-1-60558-429-4.

KAPADIA, M.; WANG, M.; SINGH, S.; REINMAN, G.; FALOUTSOS, P. Scenario space: Characterizing coverage, quality, and failure of steering algorithms. In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: ACM, 2011. (SCA '11), p. 53–62. ISBN 978-1-4503-0923-3. Disponível em: <http://doi.acm.org/10.1145/2019406.2019414>.

KARAMOUZAS, I.; HEIL, P.; BEEK, P.; OVERMARS, M. H. A predictive collision avoidance model for pedestrian simulation. In: EGGES, A.; GERAERTS, R.; OVERMARS, M. (Ed.). *Motion in Games*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5884). p. 41–52. ISBN 978-3-642-10346-9.

KARAMOUZAS, I.; OVERMARS, M. Simulating the local behaviour of small pedestrian groups. In: *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. New York, NY, USA: ACM, 2010. (VRST '10), p. 183–190. ISBN 978-1-4503-0441-2.

KARAMOUZAS, I.; OVERMARS, M. Simulating and evaluating the local behavior of small pedestrian groups. *Visualization and Computer Graphics, IEEE Transactions on*, v. 18, n. 3, p. 394–406, 2012. ISSN 1077-2626.

KAVRAKI, L.; SVESTKA, P.; LATOMBE, J.-C.; OVERMARS, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, v. 12, n. 4, p. 566–580, 1996. ISSN 1042-296X.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. [S.l.: s.n.], 1985. v. 2, p. 500–505.

KIRCHNER, A.; NAMAZI, A.; NISHINARI, K.; SCHADSCHNEIDER, A. Role of conflicts in the floor field cellular automaton model for pedestrian dynamics. In: *2nd International Conference on Pedestrians and Evacuation Dynamics*. [S.l.: s.n.], 2003. p. 51–62.

KIRCHNER, A.; SCHADSCHNEIDER, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, v. 312, n. 1–2, p. 260 – 276, 2002. ISSN 0378-4371.

KUFFNER, J. Effective sampling and distance metrics for 3d rigid body path planning. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. [S.l.: s.n.], 2004. v. 4, p. 3993–3998 Vol.4. ISSN 1050-4729.

KUFFNER J.J., J.; LATOMBE, J.-C. Fast synthetic vision, memory, and learning models for virtual humans. In: *Computer Animation, 1999. Proceedings*. [S.l.: s.n.], 1999. p. 118–127. ISSN 1087-4844.

LAMARCHE, F.; DONIKIAN, S. Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum*, Blackwell Publishing, Inc, v. 23, n. 3, p. 509–518, 2004. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/j.1467-8659.2004.00782.x>.

LEE, K. H.; CHOI, M. G.; HONG, Q.; LEE, J. Group behavior from video: a data-driven approach to crowd simulation. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007. (SCA '07), p. 109–118. ISBN 978-1-59593-624-0.

LEE, K. H.; CHOI, M. G.; LEE, J. Motion patches: Building blocks for virtual environments annotated with motion data. In: *ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006. (SIGGRAPH '06), p. 898–906. ISBN 1-59593-364-6. Disponível em: <http://doi.acm.org/10.1145/1179352.1141972>.

LEGION. *"Legion"*. 2015. Http://www.legion.com/ . Last visited on 26/01/2015.

LEMERCIER, S.; AUBERLET, J.-M. Towards more behaviours in crowd simulation. *Computer Animation and Virtual Worlds*, p. n/a–n/a, 2015. ISSN 1546-427X. Disponível em: <http://dx.doi.org/10.1002/cav-.1629>.

LEMERCIER, S.; JELIC, A.; KULPA, R.; HUA, J.; FEHRENBACH, J.; DEGOND, P.; APPERT-ROLLAND, C.; DONIKIAN, S.; PETTRÉ, J. Realistic following behaviors for crowd simulation. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 31, n. 2pt2, p. 489–498, 2012. ISSN 1467-8659.

LERNER, A.; CHRYSANTHOU, Y.; COHEN-OR, D. Efficient cells-and-portals partitioning. *Computer Animation and Virtual Worlds*, John Wiley & Sons, Ltd., v. 17, n. 1, p. 21–40, 2006. ISSN 1546-427X. Disponível em: <http://dx.doi.org/10.1002/cav.70>.

LERNER, A.; CHRYSANTHOU, Y.; LISCHINSKI, D. Crowds by example. *Computer Graphics Forum*, Wiley Online Library, v. 26, n. 3, p. 655–664, 2007.

LERNER, A.; FITUSI, E.; CHRYSANTHOU, Y.; COHEN-OR, D. Fitting behaviors to pedestrian simulations. In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: ACM, 2009. (SCA '09), p. 199–208. ISBN 978-1-60558-610-6.

LIEN, J.-M.; RODRIGUEZ, S.; MALRIC, J.; AMATO, N. Shepherding behaviors with multiple shepherds. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. [S.l.: s.n.], 2005. p. 3402–3407.

LOSCOS, C.; MARCHAL, D.; MEYER, A. Intuitive crowd behavior in dense urban environments using local laws. In: *Theory and Practice of Computer Graphics, 2003. Proceedings*. [S.l.: s.n.], 2003. p. 122–129.

MASSIVE. *"Massive Software"*. 2015. Http://www.massivesoftware.com/ . Last visited on 26/01/2015.

METOYER, R. A.; HODGINS, J. K. Reactive pedestrian path following from examples. *The Visual Computer*, Springer-Verlag, v. 20, n. 10, p. 635–649, 2004. ISSN 0178-2789.

MOUSSAÏD, M.; HELBING, D.; THERAULAZ, G. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, v. 108, n. 17, p. 6884–6888, 2011. Disponível em: <http://www.pnas.org/content/108/17/6884.abstract>.

MOUSSAÏD, M.; PEROZO, N.; GARNIER, S.; HELBING, D.; THERAULAZ, G. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, Public Library of Science, v. 5, n. 4, p. e10047, April 2010.

MOUSSAïD, M.; GUILLOT, E. G.; MOREAU, M.; FEHRENBACH, J.; CHABIRON, O.; LEMERCIER, S.; PETTRé, J.; APPERT-ROLLAND, C.; DEGOND, P.; THERAULAZ, G. Traffic instabilities in self-organized pedestrian crowds. *PLoS Comput Biol*, Public Library of Science, v. 8, n. 3, p. e1002442, 03 2012. Disponível em: <http://dx.doi.org/10.1371%2Fjournal.pcbi.1002442>.

MUSSE, S.; THALMANN, D. A model of human crowd behavior : Group inter-relationship and collision detection analysis. In: THALMANN, D.; PANNE, M. (Ed.). *Computer Animation and Simulation '97*. [S.l.]: Springer Vienna, 1997, (Eurographics). p. 39–51. ISBN 978-3-211-83048-2.

MUSSE, S. R. *Human crowd modelling with various levels of behaviour control*. Tese (Doutorado), Lausanne, 2000. Disponível em: <http://library.epfl.ch/theses/?nr=2128>.

MUSSE, S. R.; CASSOL, V. J.; JUNG, C. R. Towards a quantitative approach for comparing crowds. *Computer Animation and Virtual Worlds*, John Wiley & Sons, Ltd, v. 23, n. 1, p. 49–57, 2012. ISSN 1546-427X. Disponível em: <http://dx.doi.org/10.1002/cav.1423>.

MUSSE, S. R.; THALMANN, D. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 7, p. 152–164, April 2001. ISSN 1077-2626.

NARAIN, R.; GOLAS, A.; CURTIS, S.; LIN, M. C. Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics*, ACM, New York, NY, USA, v. 28, p. 122:1–122:8, 2009. ISSN 0730-0301.

NOSER, H.; RENAULT, O.; THALMANN, D.; THALMANN, N. M. Navigation for digital actors based on synthetic vision, memory, and learning. *Computers & Graphics*, v. 19, n. 1, p. 7 – 19, 1995. ISSN 0097-8493.

ONDŘEJ, J.; PETTRÉ, J.; OLIVIER, A.-H.; DONIKIAN, S. A synthetic-vision based steering approach for crowd simulation. *ACM Trans. Graph.*, ACM, New York, NY, USA, v. 29, n. 4, p. 123:1–123:9, jul. 2010. ISSN 0730-0301.

PARAMOUNT PICTURES, M. *"Hercules"*. 2014. Http://www.mightyhercules.com/ . Last visited on 26/01/2015.

PARAVISI, M.; WERHLI, A.; JUNIOR, J. J.; RODRIGUES, R.; JUNG, C. R.; MUSSE, S. R. Continuum crowds with local control. *Computer Graphics International*, p. 108–115, 2008.

PARIS, S.; PETTRÉ, J.; DONIKIAN, S. Pedestrian reactive navigation for crowd simulation: a predictive approach. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 26, n. 3, p. 665–674, 2007. ISSN 1467-8659. Disponível em: <http://dx.doi.org/10.1111/j.1467-8659.2007.01090.x>.

PATIL, S.; BERG, J. van den; CURTIS, S.; LIN, M. C.; MANOCHA, D. Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 17, p. 244–254, February 2011. ISSN 1077-2626.

PELECHANO, N.; ALLBECK, J. M.; BADLER, N. I. Controlling individual agents in high-density crowd simulation. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007. (SCA '07), p. 99–108. ISBN 978-1-59593-624-0. Disponível em: <http://dl.acm.org/citation.cfm?id=1272690.1272705>.

PELECHANO, N.; ALLBECK, J. M.; BADLER, N. I. Virtual crowds: Methods, simulation, and control. *Synthesis Lectures on Computer Graphics and Animation*, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–176, 2008.

PELECHANO, N.; BADLER, N. Modeling crowd and trained leader behavior during building evacuation. *Computer Graphics and Applications, IEEE*, v. 26, n. 6, p. 80–86, 2006. ISSN 0272-1716.

PETERS, C.; O'SULLIVAN, C. Bottom-up visual attention for virtual human animation. In: *Computer Animation and Social Agents, 2003. 16th International Conference on.* [S.l.: s.n.], 2003. p. 111–117. ISSN 1087-4844.

PETTRÉ, J.; LAUMOND, J.-P.; THALMANN, D. A navigation graph for real-time crowd animation on multilayered and uneven terrain. In: *First International Workshop on Crowd Simulation.* [S.l.: s.n.], 2005. v. 43, n. 44, p. 194.

PETTRÉ, J.; ONDŘEJ, J.; OLIVIER, A.-H.; CRETUAL, A.; DONIKIAN, S. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In: *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* New York, NY, USA: ACM, 2009. (SCA '09), p. 189–198. ISBN 978-1-60558-610-6. Disponível em: <http://doi.acm.org/10.1145/1599470.1599495>.

QIU, F.; HU, X. Modeling group structures in pedestrian crowd simulation. *Simulation Modelling Practice and Theory*, v. 18, n. 2, p. 190 – 205, 2010. ISSN 1569-190X. Disponível em: <http://www.sciencedirect-.com/science/article/pii/S1569190X09001555>.

RENAULT, O.; MAGNENAT-THALMANN, N.; THALMANN, D. A vision-based approach to behavioural animation. *Journal of Visualization and Computer Animation*, John Wiley & Sons, Ltd, v. 1, n. 1, p. 18–21, 1990. ISSN 1099-1778.

REYNOLDS, C. Flocks, herds and schools: a distributed behavioral model. *SIGGRAPH Comp. Graphics*, ACM, New York, NY, USA, v. 21, n. 4, p. 25–34, 1987. ISSN 0097-8930.

REYNOLDS, C. Steering behaviors for autonomous characters. In: *Game Developers Conference 1999.* [S.l.: s.n.], 1999. p. 763–782.

REYNOLDS, C. W. "Interaction with groups of autonomous characters". In: *Game Developers Conference 2000.* [S.l.: s.n.], 2000. p. 449–460.

REYNOLDS, C. W. "Big fast crowds on ps3". In: *Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames.* New York, NY, USA: ACM, 2006. p. 113–121. ISBN 1-59593-386-7.

RICKS, B.; EGBERT, P. More realistic, flexible, and expressive social crowds using transactional analysis. *The Visual Computer*, Springer-Verlag, v. 28, n. 6-8, p. 889–898, 2012. ISSN 0178-2789.

RIO, K. W.; RHEA, C. K.; WARREN, W. H. Follow the leader: Visual control of speed in pedestrian following. *Journal of Vision*, v. 14, n. 2, 2014.

SCHADSCHNEIDER, A. Cellular automaton approach to pedestrian dynamics-theory. *Pedestrian and evacuation dynamics*, Springer Verlag, p. 75–85, 2001.

SCHUERMAN, M.; SINGH, S.; KAPADIA, M.; FALOUTSOS, P. Situation agents: agent-based externalized steering logic. *Computer Animation and Virtual Worlds*, John Wiley & Sons, Ltd., v. 21, n. 3-4, p. 267–276, 2010. ISSN 1546-427X.

SEYFRIED, A.; STEFFEN, B.; KLINGSCH, W.; BOLTES, M. The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics: Theory and Experiment*, v. 2005, n. 10, p. P10002, 2005.

SHAO, W.; TERZOPOULOS, D. Autonomous pedestrians. *Graphical Models*, v. 69, n. 5-6, p. 246–274, 2007. ISSN 1524-0703.

SHIMODA, S.; KURODA, Y.; IAGNEMMA, K. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. [S.l.: s.n.], 2005. p. 2828–2833.

SILVA, A. R. D.; LAGES, W. S.; CHAIMOWICZ, L. Boids that see: Using self-occlusion for simulating large groups on gpus. *Comput. Entertain.*, ACM, New York, NY, USA, v. 7, n. 4, p. 51:1–51:20, jan. 2010. ISSN 1544-3574. Disponível em: <http://doi.acm.org/10.1145/1658866.1658870>.

SIMÉON, T.; LAUMOND, J.-P.; NISSOUX, C. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, v. 14, n. 6, p. 477–493, 2000.

SINGH, S.; KAPADIA, M.; FALOUTSOS, P.; REINMAN, G. Steerbench: a benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds*, John Wiley & Sons, Ltd., v. 20, n. 5-6, p. 533–548, 2009. ISSN 1546-427X. Disponível em: <http://dx.doi.org/10.1002/cav.277>.

SINGH, S.; KAPADIA, M.; HEWLETT, B.; REINMAN, G.; FALOUTSOS, P. A modular framework for adaptive agent-based steering. In: *Symp. Int. 3D Graphics and Games*. [S.l.: s.n.], 2011. p. 141–150. ISBN 978-1-4503-0565-5.

SUD, A.; ANDERSEN, E.; CURTIS, S.; LIN, M.; MANOCHA, D. Real-time path planning for virtual agents in dynamic environments. In: *Virtual Reality Conference, 2007. VR '07. IEEE*. [S.l.: s.n.], 2007. p. 91–98.

SUD, A.; GAYLE, R.; ANDERSEN, E.; GUY, S.; LIN, M.; MANOCHA, D. Real-time navigation of independent agents using adaptive roadmaps. In: *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*. New York, NY, USA: ACM, 2007. (VRST '07), p. 99–106. ISBN 978-1-59593-863-3. Disponível em: <http://doi.acm.org/10.1145/1315184.1315201>.

TECCHIA, F.; LOSCOS, C.; CONROY, R.; CHRYSANTHOU, Y. Agent behaviour simulator (abs): A platform for urban behaviour development. In: *In GTEC'2001*. [S.l.: s.n.], 2001.

TELLER, S. J. *Visibility Computations in Densely Occluded Polyhedral Environments*. Berkeley, CA, USA, 1992.

THALMANN, D.; MUSSE, S. In: *Crowd Simulation*. [S.l.]: Springer London, 2013. ISBN 978-1-4471-4449-6.

TREUILLE, A.; COOPER, S.; POPOVIć, Z. Continuum crowds. In: *SIGGRAPH '06*. [S.l.]: ACM, 2006. p. 1160–1168. ISBN 1-59593-364-6.

TU, X.; TERZOPOULOS, D. Artificial fishes: physics, locomotion, perception, behavior. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1994. (SIGGRAPH '94), p. 43–50. ISBN 0-89791-667-0.

UBISOFT. *"Assassin's Creed Unity"*. 2014. Http://assassinscreed.ubi.com/ . Last visited on 26/01/2015.

WARREN, C. Global path planning using artificial potential fields. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. [S.l.: s.n.], 1989. p. 316–321 vol.1.

WARREN, C. Multiple robot path coordination using artificial potential fields. In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.* [S.l.: s.n.], 1990. p. 500–505 vol.1.

WARREN, W.; FAJEN, B. Optic flow and beyond. In: VAINA, L. M.; BEARDSLEY, S. A.; RUSHTON, S. K. (Ed.). Norwell, MA, USA: Kluwer Academic Publishers, 2004. cap. From optic flow to laws of control, p. 307–337. ISBN 1-4020-2091-0.

WOLFRAM, S. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, American Physical Society, v. 55, p. 601–644, Jul 1983.

WOLINSKI, D.; GUY, S.; OLIVIER, A.-H.; LIN, M.; MANOCHA, D.; PETTRÉ, J. Parameter Estimation and Comparative Evaluation of Crowd Simulations. *Computer Graphics Forum*, v. 33, n. 2, p. 303–312, 2014. Disponível em: <http://diglib.eg.org/EG/CGF/volume33/issue2/v33i2pp303-312.pdf>.

WU, Q.; JI, Q.; DU, J.; LI, X. Simulating the local behavior of small pedestrian groups using synthetic-vision based steering approach. In: *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry.* New York, NY, USA: ACM, 2013. (VRCAI '13), p. 41–50. ISBN 978-1-4503-2590-5. Disponível em: <http://doi.acm.org/10.1145/2534329.2534337>.

XIONG, M.; LEES, M.; CAI, W.; ZHOU, S.; LOW, M. Y. H. “Hybrid modelling of crowd simulation”. *Procedia Computer Science*, v. 1, n. 1, p. 57 – 65, 2010. ISSN 1877-0509.

XU, S.; DUH, H. B.-L. A simulation of bonding effects and their impacts on pedestrian dynamics. *Intelligent Transportation Systems, IEEE Transactions on*, v. 11, n. 1, p. 153–161, 2010.

YERSIN, B.; MAÏM, J.; MORINI, F.; THALMANN, D. “Real-time crowd motion planning: Scalable avoidance and group behavior”. *Visual Computer*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 24, n. 10, p. 859–870, 2008. ISSN 0178-2789.

YERSIN, B.; MAÏM, J.; PETTRÉ, J.; THALMANN, D. Crowd patches: Populating large-scale virtual environments for real-time applications. In: *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games.* New York, NY, USA: ACM, 2009. (I3D '09), p. 207–214. ISBN 978-1-60558-429-4. Disponível em: <http://doi.acm.org/10.1145/1507149.1507184>.

ZHENG, X.; ZHONG, T.; LIU, M. “Modeling crowd evacuation of a building based on seven methodological approaches”. *Building and Environment*, v. 44, n. 3, p. 437 – 445, 2009. ISSN 0360-1323.