

SOCIAL PATH FOLLOWING

June 2011

Carmine Oliva

Master of Science in Computer Science



SOCIAL PATH FOLLOWING

Carmine Oliva

Master of Science

Computer Science

June 2011

School of Computer Science

Reykjavík University

M.Sc. PROJECT REPORT



Social Path Following

by

Carmine Oliva

Project report submitted to the School of Computer Science
at Reykjavík University in partial fulfillment of
the requirements for the degree of
Master of Science in Computer Science

June 2011

Project Report Committee:

Hannes Högni Vilhjálmsson, Supervisor
Associate Professor, Reykjavík University

Emanuela Merelli, Supervisor
Professor, University of Camerino

Yngvi Björnsson
Associate Professor, Reykjavík University

Luca Tesei
Assistant Professor, University of Camerino

Copyright
Carmine Oliva
June 2011

The undersigned hereby certify that they recommend to the School of Computer Science at Reykjavík University for acceptance this project report entitled **Social Path Following** submitted by **Carmine Oliva** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Science**.

Date

Hannes Högni Vilhjálmsson, Supervisor
Associate Professor, Reykjavík University

Emanuela Merelli, Supervisor
Professor, University of Camerino

Yngvi Björnsson
Associate Professor, Reykjavík University

Luca Tesei
Assistant Professor, University of Camerino

The undersigned hereby grants permission to the Reykjavík University Library to reproduce single copies of this project report entitled **Social Path Following** and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the project report, and except as herein before provided, neither the project report nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

Date

Carmine Oliva
Master of Science

Social Path Following

Carmine Oliva

June 2011

Abstract

Path planning consists of finding a route from one location point to another. It is also known as motion planning or navigation problem and it is common in several fields in Computer Science and commercial games since it has to be solved in real time, under constraints of memory and CPU resources. Abstracting the problem, path planning is divided in two main sub-problems: path finding, used to find a simplified route composed of connected segments and path following to make the locomotion along the route as realistic as possible. The path following should avoid contact between objects through the collision avoidance; together, they represent the main topic of our studies. We work on a virtual environment populated by agents and avatars that are engaged in social situations and show autonomous believable social behaviors. In this context, natural and fluid paths are the keys to providing a high level of presence and realism. Building on several theories about human territories, social forces and body language, we are extending the state of the art of path following; each agent is now able to avoid static and dynamic obstacles along its path, to predict future interaction patterns of others and accordingly to apply corrections to its movement. We model spontaneous form of non-verbal negotiation that humans engage in every day during their locomotion. Their behavior is also slightly affected by stochastic factors that appear under the form of little changes on velocity and/or orientation, called distortions. Our approach is able to solve path following and collision avoidance in many types of situations, finding a good balance between realism and performance. To test this, we implemented several scenarios of completely different simulations. Moreover, creating a specific profile for each agent, our system can also show how different stereotypes of people act in those situations and compare them with expected results.

Social Path Following

Carmine Oliva

Júní 2011

Útdráttur

abstract

Dedicated to my little sister Claudia

Acknowledgements

First of all, I would like to thank my supervisor Hannes Vilhjálmsson. During these months, he has been available for any problem or open question; I have received many advises from him, remained always fascinated of his experience and huge knowledge. He is a great teacher but especially an excellent person. Thanks also to my teachers in Italy that had encouraged me to come in Iceland, one of the most amazing adventure of my life. Thank you very much for your support.

Thanks to Lorenzo and Francesco. I have shared with them studies, accommodations and one whole year of my life. I am grateful of their helps and friendship. The same for all other people that I have met in Iceland; I will take care of your memory in my heart, forever.

The most special thanks is to my family and my best friends in Italy. I could not achieve this result without their support exactly as I cannot live without them. It is surprising how a smile, a message or a simple conversation with people you love most, can recharge your batteries when you are away from home and things are not going on perfectly...

Thank you so much!!!

Contents

| | |
|-------------------------------------|-----------|
| List of Figures | xi |
| 1 Introduction | 1 |
| 1.1 Social Environments | 2 |
| 1.2 Social Path Following | 4 |
| 1.3 Thesis Overview | 5 |
| 2 Background | 7 |
| 2.1 Path Finding | 8 |
| 2.1.1 A* Algorithm | 9 |
| 2.1.2 Recast and Detour | 13 |
| 2.2 Path Following | 13 |
| 2.2.1 Steering Behaviors | 14 |
| 2.2.2 Recent Works | 18 |
| 3 Human Territories | 21 |
| 3.1 Territorial Behaviors | 21 |
| 3.2 Sources of Signals | 22 |
| 3.3 Formations | 24 |
| 3.4 Spaces | 26 |
| 4 Eye Contact | 29 |
| 4.1 Functions | 30 |
| 4.2 Duration | 30 |
| 4.3 Types | 31 |
| 4.4 Behaviors | 32 |
| 4.5 During a Conversation | 34 |
| 4.6 Before a Conversation | 35 |
| 5 Approach | 37 |

| | | |
|----------|--|-----------|
| 5.1 | Prior Work | 37 |
| 5.1.1 | Colliding Agents | 38 |
| 5.1.2 | Range of Speeds and Orientations | 40 |
| 5.1.3 | Optimal Velocity | 41 |
| 5.2 | Social Awareness | 43 |
| 5.2.1 | Memory | 44 |
| 5.2.2 | Territory avoidance | 45 |
| 5.2.3 | Negotiation of Space | 50 |
| 5.2.4 | Other improvements | 52 |
| 5.2.5 | Profiles | 54 |
| 6 | Implementation and Demos | 57 |
| 6.1 | CADIA Populus | 57 |
| 6.1.1 | Path Planning | 59 |
| 6.1.2 | Gaze Behavior | 59 |
| 6.2 | Structure and Features | 59 |
| 6.3 | Demo | 63 |
| 6.4 | GUI | 67 |
| 7 | Conclusion and Future Work | 69 |
| 7.1 | Future Work | 70 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Screenshots of social interactions that occur in Second Life and The Sims | 2 |
| 2.1 | Example of path finding solution using the A* algorithm. Gray cells represent an impassible zone and red cells describe a path from the starting position (green) to the goal (blue) [17] | 10 |
| 2.2 | Different paths computed by A* and Theta* algorithms [20] | 11 |
| 2.3 | Example of a navigation mesh used in a game; the path between A and B is represented as a sequence of yellow links | 12 |
| 2.4 | Difference between path finding (black line) and following (red line) | 14 |
| 2.5 | Architecture of different layers to represent behaviors | 16 |
| 2.6 | Visual example of seek and pursuit behaviors [26] | 17 |
| 2.7 | Difference between the Velocity Obstacle (left) and the Reciprocal Velocity Obstacle (right) [30] | 19 |
| 3.1 | Cubit spaces configurations in two different postures | 23 |
| 3.2 | Territorial behaviors, postures and gestures | 24 |
| 3.3 | Conversations with different shapes approximated by circles | 25 |
| 3.4 | Graphical representation of concentric zones | 26 |
| 4.1 | Mutual gaze between lovers and fighters | 31 |
| 4.2 | Business, social and intimate gaze | 32 |
| 4.3 | Types of eyes patterns | 34 |
| 5.1 | Different perceptions already in CADIA Populus [21] | 38 |
| 5.2 | Progressive movement of agents represented by discs, highlighting two times of contact; on the right there are paths and future contacts | 39 |
| 5.3 | Particular situations to manage: same velocity vector, "false" collision, contact in the past | 40 |
| 5.4 | Heuristic functions to compute the range of speed and orientation [13] | 40 |

| | | |
|------|--|----|
| 5.5 | Width of ranges related to the first time to contact. Tables contain parameters used in the algorithm and geometric details of the picture | 42 |
| 5.6 | Continuous and discrete range of velocities | 42 |
| 5.7 | Formula to compute the cost of each velocity [13] | 43 |
| 5.8 | Correct social behavior perceiving an approaching space | 45 |
| 5.9 | The first picture shows the path followed by the agent, avoiding the obstacle; in the second one we show the difference between the desired movement of the agent, in the absence of obstacles (black line) and the linear regression result (green line) derived from the path in the image on the left | 47 |
| 5.10 | Representation of the approaching space considering future positions . . . | 49 |
| 5.11 | New formula to compute the cost of each velocity, taking into account the territory avoidance | 50 |
| 5.12 | Scenario related to the negotiation of space | 51 |
| 5.13 | Different velocity vectors produced without and with distortions | 53 |
| 5.14 | Several profiles of agents | 54 |
| 5.15 | Example of a profile stored in a XML file | 55 |
| 6.1 | Screenshots of Populus | 58 |
| 6.2 | Intervals of values saved in our lists | 62 |
| 6.3 | First scenario: basic overtaking scenario | 63 |
| 6.4 | Second scenario: robust collision avoidance moving a user controlled avatar | 63 |
| 6.5 | Third scenario: crowded environment | 64 |
| 6.6 | Fourth scenario: territory avoidance | 65 |
| 6.7 | Fifth scenario: negotiation of space | 65 |
| 6.8 | Sixth scenario: complex environment | 66 |
| 6.9 | GUI to create profiles | 67 |

Chapter 1

Introduction

In several domains like robot navigation, biological systems or simulations in general, developers need sub-systems that are able to compute accurate, correct and realistic motion. For instance, they might need to find the best route for airplanes to connect any possible pairs of cities in the world or the shortest way to reach a destination for a submarine.

Considering these examples, the problem seems pretty easy; but in more complicated situations, there are static and/or dynamic obstacles with stochastic and unpredictable movements to manage, like cars that are moving in the traffic or people in a crowded environment.

The problem is called **path planning**. Actually it is also known as motion planning, navigation problem or piano mover's problem since the problem is found in many fields of Artificial Intelligence.

The definition of the problem is quite simple: it consists of producing a continuous path (output) from a starting position S to goal point G (input). "Continuous" means a line without breaks between the two points. In most of the cases it should be computed to avoid collisions with obstacles in order to make the system more believable and to not generate spatial inconsistencies. A path that satisfies all these properties is considered valid and it represents a possible solution of the problem.

Sometimes a valid path is not enough: video-games should be as immersive as possible and many simulations of the world must be faithful to reality, following physical laws and satisfying specific properties. In these situations, the path should also be natural, realistic, believable and fluid.

The more parameters and constraints we take into account, the harder the problem becomes. This makes path planning an extremely interesting problem, especially if it should be addressed in real time with a limited amount of memory and hardware capabilities. For these reasons it requires efficient algorithms and data-structures to model the space that

represents the environment such as uniform grids, hierarchical structures, octree and so on. They can accelerate queries and reduce the time consumed.

Usually developers prefer to divide the path planning in two specific sub-problems:

- **path finding:** it takes as input the configuration of the environment, the current position of the subject and the goal to find a sequence of connected segments between departure and destination;
- **path following:** it takes as input the approximated path from the path finding and its purpose is to realistically steer the subject along the path, adding slight bends or ripples and preventing strong changes of direction.

Actually there is another intrinsic problem behind the path planning, known as **collision avoidance**. It consists of the ability to dodge around static and/or dynamic obstacles avoiding overlaps of objects. Having a strong correlation, path following and collision avoidance are merged together in several approaches that provide a unique solution for both.

The path finding is not the main topic of our study but we want to give a brief overview about the current state of art, highlighting some of the best techniques, in terms of efficiency and realism. But mainly we focus on path following addressing the problem in the specific context of human environments ruled by social behaviors. We will give a better explanation about this domain in the next section.

1.1 Social Environments

Nowadays our live is widely affected by other parallel realities provided by social networks, engaging games and virtual worlds that, in some cases, are immersive enough for users to become the dominant reality.



Figure 1.1: Screenshots of social interactions that occur in Second Life and The Sims

By definition a **social environment** consists of a specific type or portion of the virtual world in which agents or avatars behave, interact and have interpersonal relationships. Popular examples of this specific domain might be Second Life and The Sims (Figure 1.1) that create a persistent reality to socialize and to start a new life-style. Especially Second Life has received adverse criticisms about their way of representing social behaviors and animations.

The mind of agents is typically implemented using a continuous *perception-reaction* mechanism: through sensors they are able to perceive the environment and using these new data they decide to react, performing an action or activating a specific behavior.

The aim of social environments is to emphasize social behaviors. They are continuous (no break between adjacent behaviors), spontaneous (natural and absolutely involuntary) and sophisticated (precise and specific); even a little incoherence could destroy the credibility of the virtual reality especially if it contradicts with the user's common sense.

Social environments should support conversations and other face-to-face interactions between agents, coordinating their body automatically. To achieve this goal, systems should control gaze, gesture, positioning, orientation, social dispositions, and locomotion. For example, simulating a conversation: positions create a circular formation, orientations converge to the center of the formation, agents alternate mutual gaze and gaze avoidance, there are typical human gestures and so on. The whole conversation should be extremely dynamic; it implies a continuous rearrangement of agents during the speech that ensures a non-rigid formation, exactly as usual interactions in the real world. These dynamics are known as *territorial behaviors*; they are visible especially while virtual humans join/leave a conversation.

Addressing these situations, we need to talk about **social negotiation**. It is the way to produce a silent agreement between people trying to express/understand intentions and needs. Social negotiation appears under the form of non-verbal behaviors and it is ruled by shared social protocols. For instance, we do not violate the private space of a conversation, without taking part in it or usually we prefer to avoid an obstacle going on the right of it. Prior works highlight the strong relation between social negotiation, human territories and body language. They affect each other and at the same time they gain an extremely relevant role in human behaviors. Several theories explain how people interact, how they create several layers of private and non-physical space that temporarily "belong" to them and what are meanings of actions and gestures.

Naturally, there are other possible factors that should be taken into account in social negotiation. Emotion, mood, personal state, degree of attention and concentration, social role and personality affect human decisions and behaviors. Modeling these aspects of human beings is a real challenge for social environments.

1.2 Social Path Following

As we said in the previous section, social environments can achieve a high level of realism and immersion by modeling behaviors and highlighting visible features of social negotiations. Particular negotiations occur while people are moving around in crowd or structured environments.

We expect that, equipping agents with an intelligence that addresses path planning, each of them is able to reach a destination avoiding collisions with obstacles along the path. That is the basic requirement of our system. Naturally, agents also have the capability to perceive conversations and avoid them even though interlocutors represent single units.

Moreover, using a probabilistic model, our virtual humans can detect preliminary phases of future conversations by observing behaviors, such as mutual gaze, approaching and salutation and accordingly they can correct paths adopting relevant changes in their velocity. Avoiding paths has less priority than avoiding real obstacles; using an heuristic approach, for each interaction every agent is able to find the best combination of speed and orientation, going closer to the destination and dodging objects at the same time. More details about the heuristic function and parameters are in the next chapters.

During movements, our agents can also request more space to pass others if the space is not enough, using non-verbal behaviors. In this case, understanding intentions of motion and creating projections of future positions, they can predict and prevent stuck situations for themselves or other perceived agents. It is one of the most common and frequent form of negotiation that involves people every day.

Our system should be general and flexible enough to manage any kind of simulated environment, such as chaotic streets, crowded with pedestrians. It should ensure believable behaviors given information about surrounding agents to reach higher levels of realism than typical path planners that lack any social awareness.

Of course we should take into account also performance. Our solution is an extension of a prior work that emphasizes credible behavior. Adjusting some parameters, our approach can reach a good balance between performance and realism; in general the precision of movements depends on the number of agents and the complexity of the scene.

Finally, it is also possible to create several stereotypes of people, saving profiles. For instance, we can simulate behaviors of aggressive, lazy, shy or drunk agent, just modifying some attributes. Different simulations and scenarios can be compared with expected results of the real world to evaluate the system.

1.3 Thesis Overview

After a brief introduction about main features of the problem, goals and domain of our approach, the rest of the thesis is organized as follow.

Chapter 2 provides exhaustive descriptions of other techniques and methods that researchers have proposed recently, highlighting their strong and weak points and comparing them with our approach.

Chapter 3 is about human territories and social behaviors. Several theories and examples of the existing research literature will be presented to understand mechanisms and singular aspects of human interactions.

In Chapter 4 we discuss the importance of gaze and other types of gestures, initializing a conversation. Timing and behavior form are extremely important to give a positive impact, to send feedback or simply to create a good feeling. It is not a surprise if we often heard or read about "the art of conversation" or "the power of eye contact".

Chapter 5 shows our approach for the social path following. We explain the main features of the prior work that we extended, differences and improvements.

Chapter 6 is more technical. We start describing CADIA Populus, the system that includes our social path following implementation. We go on presenting our code and focusing on particular solutions and choices. We also show several scenarios in which the social path following provides realistic results.

Finally, Chapter 7 is about conclusions, and possible future works.

Chapter 2

Background

As we said in the first chapter, path planning is a common problem of several fields such as video-games, robotics and various types of simulations. These domains have in common subjects that need to move from an initial position to a destination, finding and following a pre-computed path. Most of the main techniques prefer to divide the path planning into two separated sub-problems: **path finding** and **path following**. The subject should also be able to avoid collisions with dynamic and static obstacles.

Time and memory constraints make the path planning a hard problem to solve. Especially in crowded simulations that run in real time, systems have to plan paths for a huge number of subjects and each path is difficult to compute because of many potential collisions. From this follows that a good path planning approach should absolutely take into account good **performance**. Another required feature is **realism**: natural and credible paths are essential to simulate human motion in social environments. Since the realism is in contrast with performance, almost all algorithms try to find the best combination of resources consumption and results. Finally they should be **general** and consistent enough to generate solutions for any type of situation.

Henceforth, several researches and approaches will be presented and analyzed considering these relevant properties, highlighting strong and weak points for each of them.

Starting with approaches that address the path planning without separating the two sub-problems, there is an interesting research made by Wei Shao and Demetri Terzopoulos [29]. They have worked on a simulation of pedestrians in urban environment. In particular their purpose is to reproduce dynamics of the old (and now demolished) central station of New York, creating hundreds of travelers. The project is not simply as a "crowd animation", characterized by a central brain able to manage all virtual humans because each pedestrian is powered with a cognitive and reactive mind.

This approach has several positive features. First of all the space is modeled using three

data-structures; there is a topological map that contains the physical composition of the environment, a perceptual map to facilitate perception queries and a path map useful to solve the path planning. It is a really excellent point because in this way they achieve good performance taking advantage of efficient data-structures and, at the same time, they can simulate perception and actions using a valid model. Perhaps our brain works exactly in this way, abstracting the space in several layers of precision/function. On the other hand, we are not completely sure if this multiple-layers perception could be built automatically (without additional work for developers) in other types of environments.

Secondly, their rational agents are equipped with locomotion, perception, behavior and cognition models. Each pedestrian has a mental state that selects actions and behaviors based on a stack of goals to achieve. In particular, for the path planning, they act following six rules, called *reactive behavior routines*; a special sequence of these routines is meant to solve every complicated situation. We are fascinated but at the same time doubtful because these rules are not supported by the literature of human behaviors; so nobody can assert that they guarantee completeness and rational/coherent behaviors in any situation. Moreover, this approach cannot be extended with new behaviors adding social attitudes to agents. In that sense, it seems too restrictive considering our purposes.

A similar project is "Group Behavior from Video: A Data-Driven Approach to Crowd Simulation" [16]; exactly as the previous research, the simulation evolves in a crowded environment but actually their work is completely different.

The path planning is addressed using a data driven approach: recording and storing movements of pedestrians from a video, a central brain is able to extract trajectories and to create a set of simple rules. Clustering virtual humans in groups (using the k-means algorithm) and learning rules, each group can follow a specific motion pattern derived from the video. A particular heuristic is used when more patterns are equally adequate. Moreover, rules that come from real movements of people should already include social and psychological factors, ensuring a good level of realism.

A similar solution does not fit to our context. We would like that each agent is able to compute its path, without any request to a sub-system, specialized in path planning. In this way we guarantee a decentralized application. Furthermore this approach does not allow to create customized profiles of agents and to add/activate/deactivate social behaviors.

2.1 Path Finding

The problem consists of finding a path from any possible coordinate of the virtual world to another. A path means an approximated way, composed by connected segments, that

satisfies a set of constraints. Possible constraints might be to generate the shortest path or to avoid static obstacles. Being a time consuming problem, usually dynamic and relatively small obstacles are not taken into account.

Using this definition, it is possible to give the formulation of the problem as follows:

| Inputs | Output |
|---|--|
| - starting point (S) - destination point (G) | - sequence of connected segments (from S to G) |

To solve the problem, a system requires a preprocessed discrete representation of the virtual world. For example it is possible to divide the world in cells, all with the same size, creating a *uniform grid* or a *hierarchical grid* is useful if we need several layers of sizes; we can create recursive tree structures such as *quadtree* or *octree* or show the space as a network of waypoints using a *graph*. The most important thing is that the chosen structure must be appropriate to the path finding algorithm and vice versa; furthermore any implementable technique should highlight walls, holes, doors or any other types of fixed obstacles, storing passable or impassable areas. In general space models are categorized into **bounding volumes** and **space partitioning**; details and features about them are in [10, 6].

Since path finding relates to our work, we want to summarize the state of art, giving an overview of the most interesting approaches.

2.1.1 A* Algorithm

A* (pronounced as A-star) is one of the key algorithms of Artificial Intelligence [27]. It is a particular case of "best-first search" because it explores the state space (a discrete representation of the environment) following a heuristic, without searching blindly. It is so popular because it ensures **completeness** and **optimality**: completeness means that it always finds a path between two points (given an admissible heuristic), if one exists; A* is also optimal since its solution represents always the shortest path. Moreover it works relatively quickly and it is flexible for any environment.

To run the A* algorithm we need to represent the space as a uniform grid or a weighted graph because we need to find a cost of distance between any pair of nodes/cells. Costs are computed using the Manhattan distances, if we are using the uniform grid, or as sum of weights, if we are using the graph. So, the cost of each current node is the sum of the distance-cost from it to the initial one (that contains the starting point) and an heuristic value for the distance-cost from the current node/cell to the goal (that contains the

destination point) [27]. Heuristic values are an optimistic estimate and never greater than the actual cost. We can get the optimal path organizing the list of accessible nodes/cells in a *priority queue* and finding the sequence of nodes/cells that minimizes the total cost.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 7 | 6 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | 19 | 20 | 21 | 22 |
| 6 | 5 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 18 | 19 | 20 | 21 |
| 5 | 4 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | 17 | 18 | 19 | 20 |
| 4 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 16 | 17 | 18 | 19 |
| 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 15 | 16 | 17 | 18 |
| 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | | 14 | 15 | 16 | 17 |
| 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 13 | 14 | 15 | 16 |
| 4 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 12 | 13 | 14 | 15 |
| 5 | 4 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 6 | 5 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

Figure 2.1: Example of path finding solution using the A* algorithm. Gray cells represent an impassible zone and red cells describe a path from the starting position (green) to the goal (blue) [17]

The picture above (Figure 2.1) explains how the A* works. In this example, the space is a uniform grid, the green cell is the initial cell and the blue one represents the goal. Almost in the middle of the space there is a gray obstacle. From the initial node, finding the best successor at each iteration, we can build the optimal path.

A* is an excellent algorithm for path finding but there are some open questions to address. First of all, using the uniform grid, we should set the size of each cell. Usually it depends on the dimension of the whole environment and of impassable zones. Another problem is how to give weights if perception is limited or if the agent does not know the exact destination; in this case we need to use approximated values. Finally, huge and complicated environments require more memory and time and obviously the efficiency of the A* decreases. To solve this problem, many researchers have worked on A* in order to improve the algorithm, proposing new powerful versions of it. For example **IDA*** (Iterative-Deepening A*) is still complete and optimal but it reduces the memory requirements applying the *cut-off* technique [27].

Other really interesting versions are discussed in next sections.

Theta* Algorithm

If IDA* is a good version since it requires less memory than the basic algorithm, **Theta***[20] is a good alternative to improve the realism of the path. In fact, reducing the number of

angles, intermediated points and changes of direction, the path becomes believable and more natural.

Exactly as A*, Theta* can work with a graph or a uniform grid that represent the environment. The idea behind this algorithm is based on the following assertion: the shortest path of a graph (or grid), is not always equivalent to the shortest path in a continuous environment [20]. To reduce the gap, it deletes intermediated nodes creating a direct connection between not-adjacent points, if it is possible, in other words if there is no obstacle along the path.

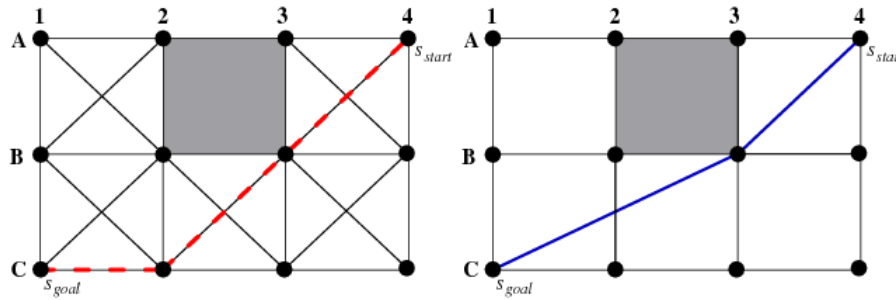


Figure 2.2: Different paths computed by A* and Theta* algorithms [20]

The picture (Figure 2.2) shows that Theta* does not include the point [C,2] to have a more realistic solution. On the other hand it is not possible to create a direct line from [A,4] to [C,1] because that path is in collision with the gray obstacle.

Actually there is a possible improvement for the Theta* proposed by researchers in recent papers. It consists of smoothing the optimized path to make it as realistic as possible. A possible disadvantage is that being a post technique, it affects the performance negatively. Theta* is the best solution for systems that want to achieve the highest level of realism. Moreover, its contribution facilitates the work of the path following starting with an excellent approximated path.

HPA* Algorithm

HPA*[2] is synonymous of hierarchical path finding. Dividing the whole grid/graph in several small areas, finding connected areas that approximately contains a solution-path and running the A* for each graph representation of them, we can obtain a sequence of sub-paths that should represent the solution. Parallelizing executions of A*, the performance increases significantly.

This technique is useful especially if we are working on huge environments or with limited resources. To apply this solution we should be careful about continuous changes of the environment and end-points of the path which are not nodes of sub-graphs. A recent

research is trying to address these open questions studying different solutions for static and dynamic environments [2].

Navigation Mesh

Another way to improve the A* in terms of performance consists of working on and optimizing the data-structure that represents the environment. It is basically the *state space* of our search problem.

One of the most used technique in computer games to achieve this purpose is called **navigation mesh**. It is a particular type of graph that allows to model the space as a set of convex polygons, defining "walkable" areas. Vertices of each area represent nodes and each edge is a link of the graph; a path is a particular sequence of adjacent links (Figure 2.3).



Figure 2.3: Example of a navigation mesh used in a game; the path between A and B is represented as a sequence of yellow links

Using a navigation mesh it is possible to approximate the environment collapsing completely empty zones (saving memory) and to have more details in more complicated areas. In this way there is a perfect balance between performance and accuracy in computing the path.

An essential prerequisite to use this technique is that the space should not change significantly during the execution of the system. Another requirement is to represent areas using *convex* polygons; it is important to have a unique straight line between each couple of points/nodes of the mesh. A typical shape is the triangle because using it, it is extremely easy to merge or divide areas.

[22] shows the main positive features of a navigation mesh: it ensures simplicity, com-

pleteness and efficiency. In particular we want to highlight two specific points: building a navigation mesh could take few minutes but at runtime, performance of the path finding is extremely good. Furthermore it ensures full automation: knowing geometric features of the environment it is possible to automatize the navigation mesh construction process.

2.1.2 Recast and Detour

We have just seen how to improve the A* in terms of memory usage, realism and performance. There exists an impressive work of Mikko Mononen that tries to achieve the best result in all these tasks, known as **Recast and Detour**. It is open-source and it is currently implemented in C++.

Recast consists of a tool to create a navigation mesh for every kind of environment, using rasterization and voxelization techniques. It is general, completely automatic and efficient. Detour is the complementary part, used to find the path from the mesh. Of course it can work with any kind of navigation mesh but Detour fits perfectly with the output of Recast. Details about the project are available in [1].

2.2 Path Following

Having an approximated path available from a path finding procedure, the task of the path following is to apply little changes of locomotion (speed and/or orientation) in order to make it as realistic as possible. In particular we want to obtain a unique continuous movement without strong changes of direction adjusting the current path. Usually these corrections are driven by forces that rule motions in each specific domain.

From this description we can summarize the problem as follows:

| Inputs | Output |
|---|------------------------|
| - approximated path - forces and constraints | - final realistic path |

An example can be useful to understand the real, visible difference between finding and following. Image a car that is running along a racetrack (Figure 2.4). In the picture, the black sequence of segments that follows the center of the racetrack represents the approximated path of the path finding. The real trajectory of the car is completely different; the red line tends to go close to the corner of each curve and close to the opposite board after and before it. The goal of the path following is exactly to transform the black path

into the red one.

In this example all forces to consider are physical: friction coefficient of ground, wind, centrifugal force and so on. Other domains imply other types of forces. For instance humans are ruled by social and territorial needs that affect their behaviors and motions.

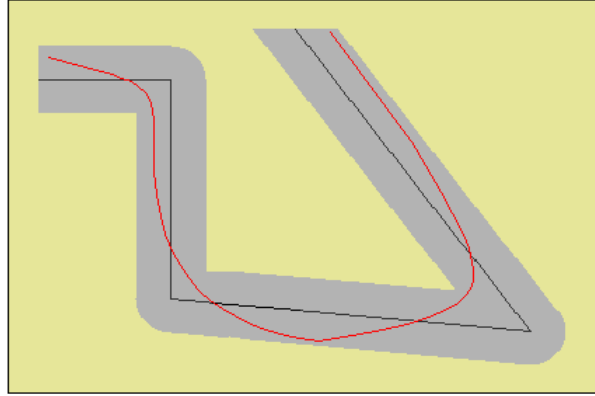


Figure 2.4: Difference between path finding (black line) and following (red line)

The path finding computes a path only once, without managing dynamic obstacles. Consequently, the **collision avoidance** becomes a sub-problem of the path following that updates the trajectory for each iteration until the subject reaches the destination.

Every subject should be able to perceive potential obstacles along the path and to dodge around them in a realistic way; this is done with both dynamic and static obstacles, achieving the same degree of realism. Actually, the collision avoidance can be seen as a particular force that prevents inconsistencies, common in almost all the domains; so all is congruent with our formulation.

Path following does not have a general procedure and it does not require specific data-structures. Its approach depends on the subject and the domain. For example humans and animals are affected by psychological, unconscious and instinctive forces, difficult features to implement. Because of these reasons the path following is a hard problem, sometimes snubbed by developers, even though it is fundamental in social environments to achieve a high level of realism.

2.2.1 Steering Behaviors

Talking about path following it is almost impossible not to cite Craig W. Reynolds and his contribution in Artificial Intelligence [23, 24, 25, 26]. He has proposed mental models composed of distinct forces that drive behaviors and movements. From this principle, the work of Reynolds is also known as **steering behaviors**. It represents the essential starting point of many other researches in this field; even today, after several years, path following

is almost synonymous of steering behaviors.

Initially, he has worked on interesting dynamics that we can find in the natural world; in particular he was interested in movements of flocks, herds and schools. In fact behaviors of birds, fish and other kind of animals are affected by a set of shared rules and stimuli that come from their interactions. Similar theoretical assertions are applicable in other types of contexts, for example traffic and pedestrian simulations.

Observing birds in flight, they have an incredible aggregation motion: it seems driven by a common brain that creates a fluid, harmonious and unified flow of moving entities. The same happens for fish and insects; especially small animals, moving in a group, look like a single living being. Their representation is similar to a *particle system*: collection of a huge amount of individual particles, each having its own simple behavior; usually it is the best way to represent particular effects like fire and clouds. Exactly like in a particle system, to simulate a flock we need to clearly define the behavior of each bird.

Simulating each "**boi**d" (the technical term to identify any creature such as a schooling fish) is complicated. Its perception is limited: it is aware of itself, two or three nearest neighbors and only by approximation the rest of the flock. From this information, it generates a behavior that fits perfectly with the others, establishing orientation, speed and intention. Reynolds highlights that their decisions are the result of two balanced and opposing behaviors: *a desire to stay close to the flock* and *to avoid collisions with others*. In other words each of them would like to preserve its space but at the same time it needs the flock which ensures protection from predators, supply of food and a better chance of survival.

In particular he points out three rules to simulate their behavior [25]:

- *collision avoidance*: maintaining separation from other flockmates it should avoid collisions;
- *velocity matching*: similar velocities (speeds and orientations) of boids, makes the flock "polarized", creating a simple moving entity;
- *flock centering*: they attempt to stay close enough to neighbors.

These are the same rules that in [24] are better summarized in **separation**, **alignment** and **cohesion**. The behavior is modeled as a particular combination of these rules. A good way to manage opposing forces is to give them a weight; for example, to prevent critical situations, the separation (collision avoidance) should have a higher priority than the others.

A simulation can start assigning random positions to all boids, fixing the minimum and maximum separate distance values. Immediately, they start to rearrange themselves affected by behavioral forces, similarly to people entering a conversation. In few moments,

solitary boids form new small flocks or join to already existing groups.

This should create a robust and believable animation; of course to manage a huge amount of agents and their behaviors, the system should consider the level of performance. In this regards, [23] reports possible efficient solutions to implement perceptions and reactions taking advantage of hierarchical representations of the space.

In [24] Reynolds extend his previous work to the world of humans. Exactly as birds and fish, it is possible to create a behavioral model also for people, simulating credible actions; obviously the degree of complexity is completely different.

First of all the animation of the body is more sophisticated; secondly people interact more with the environment, extracting more information, elaborating them and activating a behavioral response. For this reason we need a mind model that converts a perceptual input to visual updates of position, orientation, speed and appearance in general. Sometimes behaviors are modeled using *finite state machines* (FSM); actually it is not the best way to represent human beings because most of the time there is not a unique state to identify a behavior and having discrete states, FSM is not able to simulate continuous reactions, like steering behaviors.

”Steering Behaviors For Autonomous Characters” [26] is the most important research of Reynolds. The work is focused on creating realistic movements for autonomous characters. They belong to a particular class of agents that are *situated*, *embodied*, *reactive* and *virtual*. Situated means that the character shares the world with other entities; having a physical representation it is embodied; each character is reactive since its behavior is instinctive and consequence of perceptual stimuli; it represents a virtual agents or in other words a real agent in a virtual environment [26].

The goal of each character is actually a set of simple sub-goals to achieve. From this point of view, Reynolds models also behaviors as a hierarchy of three layers, as showed in the picture (Figure 2.5).

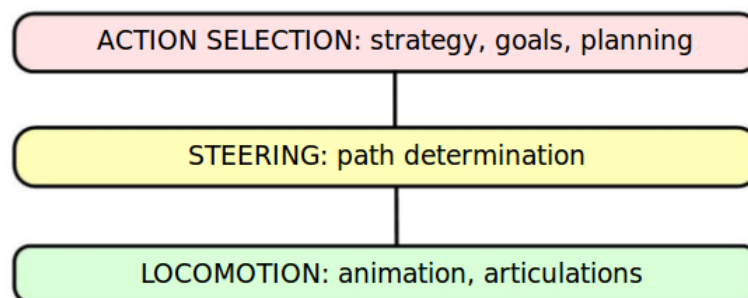


Figure 2.5: Architecture of different layers to represent behaviors

Each agent is driven by a set of steering behaviors (executed in sequence and/or in parallel) to achieve a specific goal and mapping them into signals, the locomotion layer creates movements and animation. It is extremely important to leave the three layers completely independent.

For example, a character wants to follow a target (goal). First of all, the steering layer generates an approximated path whence to extract desired velocities; actually these vectors of velocity will be slightly modified considering forces related to subject, intentions and environment properties. Each final vector determines a sequence of graphical changes produced by the locomotion layer. From this example we can image a steering behavior as a precise geometric computation; in general it finds the right balance between desired and environmental forces. As result we obtain the expected, fluid and realistic motion.

Craig Reynolds explains how we can create a wide set of behaviors. Most of them are simply a simultaneous combination of basic steering behaviors; it is one of the most powerful features of this approach. Of course, describing all of them takes too long, we want just to show most common human behaviors (Figure 2.6):

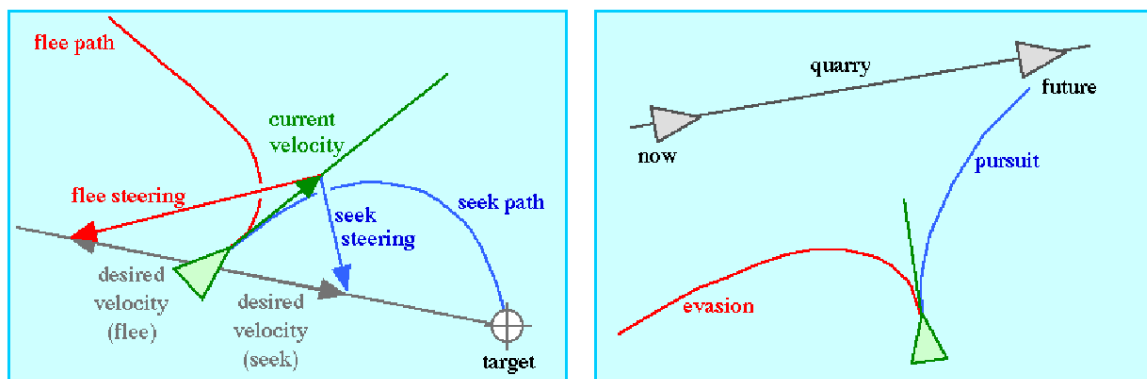


Figure 2.6: Visual example of seek and pursuit behaviors [26]

- **seek**: it steers the character towards a specific destination (target). It merges the "desired movement" with the "attractive movement" adjusting the trajectory. **Flee** is the inverse of the seek (opposite direction).
- **pursuit**: it is similar to seek but the target is moving. The real pursuit requires a prediction that can be estimated step by step considering the behavior of the target and previous moves. A possible improvement is to realize large movements at the beginning and small movements when they are nearby. Its counterpart is the **evasion**.
- **offset pursuit**: it is equal to the pursuit but the path should pass near the target but not exactly on it.

- **arrival:** it is equal to the seek but at the beginning the speed of the movement is high and when the subject is near to the target it decreases.

A combination of steering behaviors can also simulate a realistic **collision avoidance**, but only for static obstacles. All entities are approximated using discs; computing the distance between positions (Euclidean distance) and comparing it with the sum of radii, it is possible to check collisions and eventually to correct the trajectory. Unfortunately this approach does not give any guarantee about dynamic obstacles avoidance.

Other possible composed behaviors that we can generate are: **wander, explore, containment, alignment** and so on. Details and geometric computations of these behaviors are described in [26].

2.2.2 Recent Works

From the seminal work of Reynolds other research has been done in recent years; in particular [19] follows exactly his idea. It is a project that consists of creating a believable scene simulating pedestrians and their motions. Assigning a repulsive potentials to obstacles and attractive potentials to goals/targets, their combination produces a realistic movement. Each pedestrian can select a plan of navigation that derived from five tasks (monitoring, yielding, checker-boarding, streamlining and avoiding of obstacles) and relevant features of the current environment such as speeds and directions of other pedestrians, distances and blocked configurations.

It seems a good approach but exactly as steering behaviors, it simplifies the problem of collision avoidance with dynamic obstacles and does not take into account social behaviors. A completely different work is [11]. This research gives another meaning and purpose to path following: instead of improving the output of the path finding, it generates a path directly from the motion tracking of humans. Basically they need to give to an assistant-robot, able to accompanying people, socially acceptable behaviors. They have studied two different approaches: direction-following and path-following. Surprisingly the simple direction-following ensures more realism but of course a hybrid method, as often happens, is the best alternative.

A very interesting work is called "Reciprocal Velocity Obstacle" [30]. It offers a new method to solve the collision avoidance in real-time and multi-agents systems, fitting perfectly with our context.

This approach provides several advantages:

- it can work in any kind of domain (robotics, crowd simulations and games);
- continuous cycle of perception-reaction;

- agents navigate independently without explicit communication with others;
- each subject can solve collisions on its own;
- since agents are independent and autonomous, the approach is suitable for parallelization;
- the collision avoidance works well with static and dynamic obstacles;
- ensuring good performance, it is possible to simulate environments with hundreds of agents.

”Reciprocal Velocity Obstacle” is of an extension of a prior research called ”Velocity Obstacle” proposed by Fiorini and Shiller. By definition, the velocity obstacle VO_B^A is a set of vectors of velocity for the agent A that will result in collision with the obstacle B, taking into account its movement. Graphically VO_B^A represents a convex area. In other words, any possible desired velocity of A that belong to the set, generates a future collision. To avoid overlaps, every subject should select a velocity that does not lie in that area, but still close to the desired vector. The problem of this prior work is that it might generate undesirable *oscillatory motions*: it is a continuous alternation of choices between the desired vector and its correction without solving the collision [30].

The Reciprocal Velocity Obstacle provides a good correction to the initial approach. For each interaction, the subject selects the intermediate vector between the desired and the corrected velocity. As we can see in the image (Figure 2.7), the idea is similar to the Theta* algorithm that improves the A* by combining vectors and smoothing the trajectory.

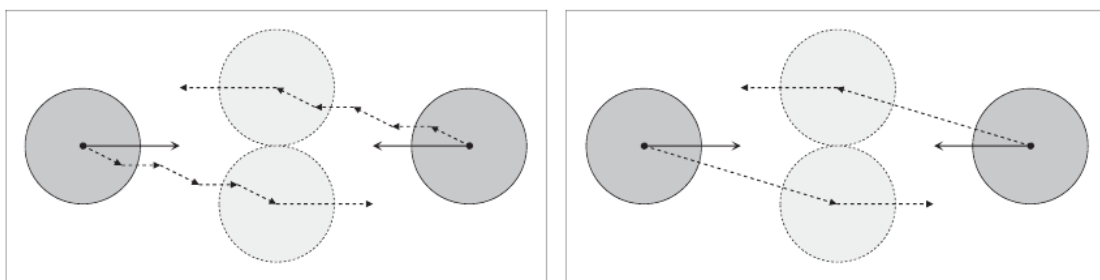


Figure 2.7: Difference between the Velocity Obstacle (left) and the Reciprocal Velocity Obstacle (right) [30]

The Reciprocal Velocity Obstacle is more robust than its ancestor; it also provides simpler locomotion with a good level of realism, as we can see in the picture. It should be able to solve the collision avoidance in any kind of simulation achieving always optimal results. Several videos demonstrate its goodness.

Unfortunately we need to highlight some weak points of this approach. It is a particular

typology of path following, completely oriented towards collision avoidance; absolutely avoiding collisions is one of the most important requirement but sometimes we may need to assign a high priority also to other steering forces, as shown by Reynolds. The Reciprocal Velocity Obstacle seems too rigid and mechanical to create other behaviors, so it seems also unsuitable for social environments.

A really recent research fits perfectly with our problem [13]. Like the previous one, it is a path following approach for human simulations, able to solve the collision avoidance with dynamic obstacles. Each agent, perceiving other entities, can predict times of contact. In case of future collisions, they create a range of possible orientations and speeds from the current desired velocity. Its size depends on the time of contact: short time implies a large range, to avoid any overlaps. Discretizing the range, the agent can select the best combination of orientation and speed through an heuristic function that minimizes energy consumption, deviation angle with the desired velocity and the risk of collision. This procedure should be similar to real mental processes of humans during their motion.

It is the first approach that provides a collision avoidance based on changes of orientation and speed as well; this generates a real human-like behavior. It is extensible and flexible because we can easily modify the heuristic function adding/removing factors or giving them a new priority; similarly it is possible to include social behaviors. Furthermore it can work with any type of path finding already implemented.

A drawback of this approach is the performance spent on processing candidates of orientation and speed. It depends on how much we have discretized the range; in fact reducing the number of candidates it is possible to have good performance and results at the same time. Finally this path following technique shows impressive results of simulation, much higher than all other algorithms.

In conclusion, we build on this previous work but go beyond the state of the art by adding a level of social awareness that should achieve better results.

Chapter 3

Human Territories

Usually when we hear the word "territory" we think about a physical and well-defined geographic area that belongs to animals, people or institutions. This is a general definition, but when we talk about **human territories**, things are a little bit different.

As explained in "Human Territories: how we behave in space and time" [28] (Schefflen, 1976), a territory is not always a physical, tangible and lasting thing. It is the space between, around and among people in which they act and behave. We can think of human territory as a dynamic unit of space defined for a certain amount of time and continuously rearranged by people and their behaviors. Any kind of action maintains, changes or affects territorial spaces; from this point of view they represent **territorial behaviors**.

3.1 Territorial Behaviors

In everyday life, people don't pay attention to their orientations, positions and gestures; all is spontaneous, involuntary and coherent with the situation. Every movement seems natural and in harmony with the others, especially during social interactions with other individuals.

This is just an illusion. Everything has a specific meaning and expresses a function: we act in a specific way because we want to communicate something or we want to claim our territory or we want to send feedback to other people, even if unconsciously. In most of the cases, our actions are expression of presence in the territory.

The origin of territorial behaviors are related to the evolution of human beings. Almost all animals claim their geographical areas marking it through scent or visible signals. This ensures power, food and possibility of breeding. Humans are descendents of animals and their needs and rationality are different although some behaviors come from similar

reasons of survival.

The ability to relate to others is exercised daily, acquiring new knowledge everyday and social unwritten rules. In particular, a wide variety of skills are dedicated to just understanding behaviors of others and replicating them when we need.

First of all, humans tend to protect their physical space keeping a minimum *distance* from others. It depends on the situation, degree of involvement, type of relationship and cultures. Crowded or empty places, partners or strangers, kinship or employment relationships, Mediterranean or Nordic origins determine a greater or a shorter space between people. Just changing or adjusting the distance, you are communicating something to the other person. [18] shows that in standing interactions people are more flexible and keep less distance with other. Conversely, seated positions implies greater distances; sometimes interactions are also constrained by the arrangement of furniture or rooms. Possible examples are a queue of people or two people sitting on a bench. During a conversation participants continuously rearrange their distance especially during changing of topic or while other people join/leave.

Secondly, there are situations in which we claim a bigger area than our personal space. For example, in public places if we want to leave an empty chair near our place, we usually place a bag on it avoiding eye contact with newcomers. In this case, objects are used as territorial markers to protect temporary places. The same happens during particular activities; having a private conversation, we act keeping people away, protecting our area.

3.2 Sources of Signals

Head, hands, shoulders, arms, elbow, legs and feet are interesting sources of non-verbal signals. They are extremely important considering that the seventy percent of communication between human beings is transmitted using just the body [18]. Through them, a person can defend his/her territory and other individuals, understanding signals, will act with a polite non-verbal behavior. It is a sort of silent agreement that suggests, for example, to avoid a private conversation maintaining a certain distance or (if it is impossible, in crowded situations) at least dipping head and eyes.

Schefflen explains that it is possible to identify four main regions of our body that can transmit communication. They are called **cubit spaces** or simply cubits (Figure 3.1). Vertically they cover the whole body and horizontally their width is more or less the distance from elbow to elbow, in other words eighteen inches (forty five centimeters). Obviously the real size depends on sex, age and features of each person. All together compose the *k-space*

(minimal space allocation for an individual).

Each cubit is independent from the other; they could have the same or different orientations, especially if people are performing several activities. A parent that is cooking with the head turned to the family and the rest of the body directed to the food, is a typical example.



Figure 3.1: Cubit spaces configurations in two different postures

Orientations and postures as well have a relevant role in interpersonal relationships. They can encourage/repulse interactions through their hidden meanings. Shared orientation/territory and mutual involvement can generate understandable **links** between people; on the other hand, opposite configurations impede any form of communication. For instance, two people that are facing each other have a good probability to interact but if one of them lowers his/her face down, covers the mouth with his/her hands or changes orientation sending negative feedback, the link weakens.

Gestures are even more powerful in communication than postures because they are more expressive, immediate and easier to understand. They include movements of any part of the body but the most interesting signals come from face and hands. They are used to emphasize a concept during a speech, point a specific object, draw attention and hundreds of other purposes. Despite their expressive power, paradoxically, the impact of gestures on human territories is minimal.

Another property to take into account is the location, defined by the position of a subject. It is represented as an approximated circle with ill defined boundaries; usually it is bigger than the space required for the body (k-space) but the size depends on culture and situation. In crowded situations, people tend to claim a restricted location but in general, small locations suggest involvement or friendly relationships. Strangers should not violate this space without consent.

Figure 3.2 shows two interesting situations for analysis, taking into account all the things considered so far. In the first one, adults are acting with typical gestures that express attention and interest; the upper part of their body (not the lower part) addresses the girl; she is refusing any kind of interaction through her posture and gaze; the huge space between subjects means that the girl wants to protect her territory. The second photo is completely different: almost all people are claiming small locations; the posture of the old woman denotes her territorial power.



Figure 3.2: Territorial behaviors, postures and gestures

Going deeper in the field of gestures, orientations and postures, we need to talk about the **body language**. This is a really complex and wide subject, studied by social psychologists for decades, out of the aim of the thesis; it is analyzed in [12]. For our purpose, we need to know that behaviors require the right interpretation, evolve dynamically applying coherent changes and many of them are expression of territorial needs.

3.3 Formations

One of the most common form of interaction between people is the conversation. In particular when people cluster together facing each other, we use the term **formation**.

It consists of a specific human territory defined by people and organized with several degrees of complexity. Each participant follows and silently accepts social norms that regulate the formation; they are expressed through verbal and non-verbal behaviors and related to experience, situation and relationship.

The simplest type of interaction is called **face formation**, basically a face-to-face conversation. Supposing that individuals share equal status and affiliation, three people define a

triangle and a square is the common shape of a conversation between four people. Actually, all these different shapes can be generalized using a circle and assigning equal circular sector for each person (Figure 3.3). Circles are also the typical way to describe a conversation between more than four participants.

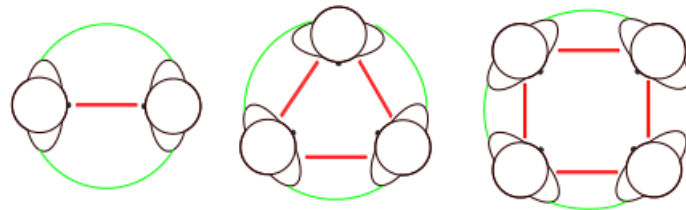


Figure 3.3: Conversations with different shapes approximated by circles

When more people join the formation, it becomes difficult to maintain face-to-face interaction. For this reason, as explained in [28], the circle tends to be unstable and all breaks down into separate clusters of few interlocutors.

As we said before talking about locations, during interactions people give part of their space up, especially if there is a high involvement or a good feeling towards the others. For instance, usually there is no space between partners or longtime friends.

Typical arrangements of people are an arc, a row/column, a set of rows/columns or a larger area if the number of participants is high. Its structure is quite complicated; it is a hierarchical organization and main layers are:

- **locations:** areas needed by each participant of the whole formation. It extends the k-space, requiring more space.
- **modules:** a formation is organized into a module when conversants keep deliberately and empty location between them or if one side/arc of the formation is occupied by affiliated people.
- **nuclei:** locations and modules are organized around a central space called nucleus. It is a larger space than a single module and usually, during a conversation splitting, it is affected by some changes such as space divided in adjacent modules or cross-channels of face-to-face interactions.
- **regions:** it surrounds the nucleus and covers a bigger area. It is used by potential newcomers to approach an established formation or as a passageway for passersby.

This general structure is combined with another type of organization that divides the human territory in zones, also called **spaces**. For example a nucleus is actually a composition of two zones with different functions and meanings: *o-space* and *p-space*. Spaces are further

explained in the next section.

Human territories are also defined by combined formations; actually, they are very common in everyday life but at the same time complicated and difficult to comprehend. There are two main types of combined formations:

- **gatherings:** people are in a same space without sharing orientation. They can occupy a single or a set of areas, depending on their size; sometimes they consists of a composition of small subgroups; in this case there are multiple sub-formations, especially if people have different status or functions.
- **hubs:** they consist of a specialization of gatherings in which there is a remarkable difference between status and role of individuals placed in the inner and outer zone. The nucleus is occupied by "performers" and in the region there are "spectators"; in fact a theater is a typical hub formation. Sometimes the region is subdivided in two parts: one for spectators and the other used by newcomers and passersby.

3.4 Spaces

Dividing a human territory in spaces is an abstraction to understand how the complete area is used, in other words how people organize a territorial space. It consists of considering locations, modules, nuclei and regions as a unique complex area divided into six concentric zones (Figure 3.4).

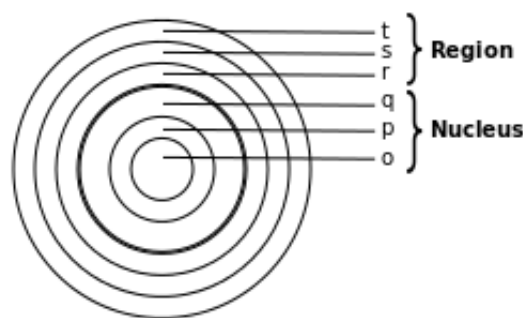


Figure 3.4: Graphical representation of concentric zones

As explained by Schefflen in [28] and confirmed in [21], each of them has a function providing a specific purpose:

- **o-space:** this is a small zone that lies at the center of the cluster and contains the "spot" of the conversation (fulcrum of orientations). It is usually occupied by children

or dogs that wander there; it is assumed that any other people (juveniles or adults) do not walk through it or use it as a stable position.

- **p-space**: it is the zone of main interlocutors of the conversation; for this reason it is also known as "zone of participants" and holds all their locations.
- **q-space**: this zone is about salutations of people that want to join/leave the conversation. It represents the area between newcomers and real participants; it is also the last level of the nucleus.
- **r-space, s-space and t-space**: they belong to the region and are typically used as passageway for passerby or buffer area for potential new participants. Levels differ in degree of affiliation and relationship with current conversants.

During the motion, territories are less defined: we need to consider the *k-space* and the *personal space* (approximately two times the k-space) of people. Actually, approaching each other, people are moving in a particular territory called **approaching space**. It might be represented by an oval that progressively becomes a circle when people are close enough to start the conversation; in other words the p-space derives from the approaching space.

Chapter 4

Eye Contact

Often we assume that conversations require no special skill. Actually many people feel foolish and inadequate interacting with others. How many times are we unable to find the right words to use or we do not understand timing and turns or we do not feel comfortable in general. These are just few examples. Just starting a conversation sometimes is a problem.

Conversing is an art that requires experience and attitude. It is not just an exchange of information because each interchange has a high level of complexity derived from automatic and unconscious rules applied by people. A good conversant should be a interested listener, able to take turns and direct attention, all by using appropriate non-verbal behaviors.

Eyes have a critical role [12]. Usually we use them as receivers of information but during social interactions they are also transmitters of signals. Their language is understood everywhere and any minimum variation or movement is taken and translated into intentional messages. Just using eyes you can looks wise, anxious, disinterested and so on. It is their power but their weakness as well because other people can perceive your mood, personality and current emotions. Eyes are an observable reflection of the cognitive state that the person is in. People's soul becomes a open book.

| Signal | Interpretation |
|--------------------------------|------------------------|
| maintaining direct eye contact | confidence and honesty |
| looking down | shame |
| looking away | dishonesty |

Studies assert that eyes are the most expressive part of the body concerning non-verbal behaviors; their intensity, direction and duration belong to a second level of communication. The table shows some typical interpretations of eye movement.

In general eye contact provides different functions and is extremely useful starting or during a conversation, taking part in the continuous process of *action-understanding-reaction* that evolves in human interactions.

4.1 Functions

Walking down the street without engaging in conversations, people interact just using their eyes, even if the contact takes only few moments. Research demonstrates that when looking at a photo or playing with an infant, we usually focus on the eyes; *gazing* is the right term to represent an intentional and protract eye contact toward a point of attraction. Gaze is used as non-verbal behavior to achieve several purposes [15]:

- **provide information:** statistics assert that during a vocal communication approximately eighty per cent of information is received through the eyes and only the ten per cent through ears. Using eyes, we store different types of data about the interaction and other participants.
- **regulate interaction:** eyes are used to understand turns and timing of a conversation. Usually there is a brief glance before your intervention; it represents the expectation of having your answers/clarifications, coming from other participants.
- **prediction of intentions:** often eyes are a powerful tool to express/understand decisions and purposes. Of course it requires good interpretations of signals, in order to avoid misunderstandings.
- **express intimacy:** gaze is expression of feeling, relationship and degree of involvement between two or more people.
- **exercise social control:** sending and receiving feedback through eyes and facial expressions, people increase their skill in communication and social relationships. For example humans look more at those they like and the duration of the gaze increases in pleasant activities.

4.2 Duration

The duration of the gaze is another interesting point because it depends on several parameters. For example Arabs and Latin people maintain the eye contact more than others. Timid, nervous or dishonest people may have a shorter or intermittent gaze. Women usually

look at the eyes of partner or conversant more often than men. Positive moods produce high level of gaze. Strangers exchange only few glances, nothing compared to lovers; paradoxically, the length of gaze of lovers and rivals is almost the same (Figure 4.1).



Figure 4.1: Mutual gaze between lovers and fighters

From these examples, we can say that the duration depends on individuals, cultures and current emotion/mood. In general, using statistics, the average person looks at the other between thirty and sixty per cent of the whole time of their interaction [12]. Sometimes this data is not really useful or too general especially analyzing specific situations. More interesting results can be obtained studying the duration of the gaze before and during a conversation. Details about this research is in the next sections of the chapter.

4.3 Types

H. Lewis in [12] explains that people of different cultures do not interact using the same gaze; in particular he highlights three forms of it that we summarize in the table below.

| Name | Focus on | People |
|-------------------|---------------|---|
| sharp | eyes | Arabs, Latin Americans and south European |
| clear | head and face | |
| peripheral | no real focus | |
| | | north European, most of the Asians |

Moreover, diverse situations can also imply diverse gazes (Figure 4.2). Our behavior is completely different during friendly conversations, important work meetings or unpleasant encounters. We want to point out some of the main typologies [12]:

- **business gaze:** gazing an imaginary triangle between eyes and forehead of the interlocutor we look serious, with clear intentions and able to keep the situation under control.
- **social gaze:** in social situations, looking interested in the speech to create a friendly atmosphere, our gaze is directed to a triangle between eyes and mouth.
- **intimate gaze:** in this case the gaze can cover all the upper part of the person's body. It is expression of interest and attraction. Intimate gaze is extremely important to encourage and predict face-to-face encounters. We can distinguish intimacy, intimidation and involvement considering the length of the gaze.
- **eye-dart:** eyes that dart in all directions means that the person is nervous and under pressure.

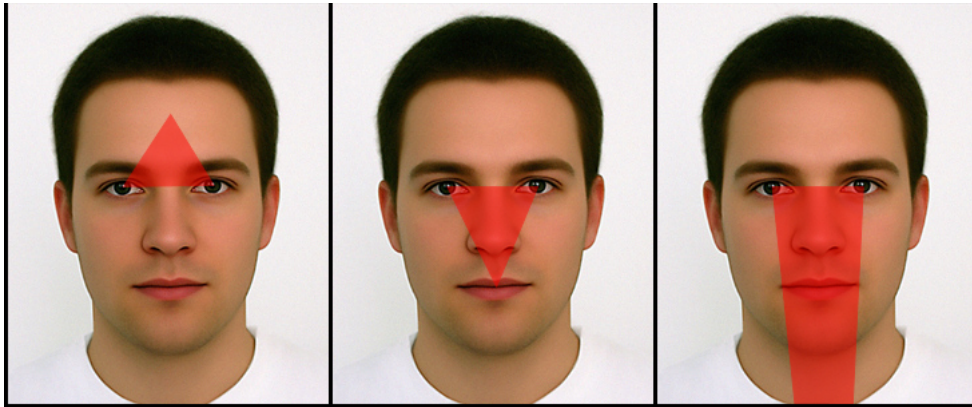


Figure 4.2: Business, social and intimate gaze

Actually there is another typology (unrelated to situations and cultures) to take into account, perhaps the most important; it is called **mutual gaze**. It represents the real and pure form of eye contact that begins with brief glance and progresses in repeated and longer contacts. Generally, mutual gaze tends to be rather short (about two seconds). Individuals that look simultaneously into each other's eyes for an extended period exhibit reciprocal attraction while brief mutual gaze indicates mutual recognition or interest.

4.4 Behaviors

In greetings, conversations or general interactions there is a continuous exchange of glances between people. The duration and number of gazes explain immediately what type of relationship exists between them. Before a conversation, while people are approaching each other, they follow specific social norms that facilitate future encounters and demonstrate

respect of other's privacy at the same time.

Something similar happens in other situations. People know that staring or gazing could be annoying or out of place, especially if they are in public places interacting with strangers. Exactly as being too close to an individual is a violation of the territory, staring at him/her is a violation of his/her privacy.

For these reasons, Lewis suggests appropriated behaviors that humans activate in various contexts [12]:

- **polite inattention:** in uncrowded places, people tend to avoid staring at another individual (especially if they are not moving). There is just a brief exchange of glances to make it obvious that they have seen each other. It is a polite way to say: "I aware your presence but I'm not interested in conversing with you". [3] describes this avoidance as a specific force, called *inhibition of gaze*, that affect our behavior.
- **exchanging glances:** it is the typical behavior of people that are walking on the street: it consists of giving a quick look in the direction of the other individual at a certain distance (more or less eight feet) and then keeping looking away till they pass each other.
- **look-and-away:** it is used when we are facing anyone that we find a little bit odd, for example someone with unconventional hairdo. In order to respect their privacy, we avoid staring and limit the eye contact as much as possible.
- **awkward glances:** it represents the behavior of people in a bus, train or elevator. In these situations the eye contact is almost inevitable; a good way to apologize for the gaze is a simple, slight smile. Politely, people avoid staring at others.
- **bedroom eyes:** eyelids are almost three-quarters closed; sleepy or sick people behave in this way. The opposite behavior is called **flashbulbs eyes:** eyelids are open as much as possible expressing anger, surprise or fear.

Some simple positions and directions of eyes are immediately understandable by everyone (Figure 4.3). Most of these *eye-patterns* are related to real, physical mind processes: considering our brain divided in two hemispheres, one for logic and the other for instinct, during the process of thinking our eyes seem to point out the specific part of the brain that is currently used. For example looking up and to the right is a typical behavior of a person that is trying to recall something from his/her memory; contrariwise looking up but to the left is often used when people are rebuilding images of their experience or dreams; when a person responds to a remembered sound they usually look on the right, defocused eyes are an expression of constructed visual imagery access and closed eyes denote concentration in smells and tastes.



Figure 4.3: Types of eyes patterns

4.5 During a Conversation

The role of gaze in face-to-face interactions is a really interesting topic. There is a wide literature about it since eye contact is used as a mean of synchronization with the interlocutor and a second channel of communication. Accordingly, research is mainly interested in regulation of turn taking and differences of eye-behavior between speaker and listener.

Human interactions seem realistic and natural because each participant is able to manage and understand the control of the flow; during conversations, it is provided by gaze and other non-verbal behaviors. As with human behaviors in general, eyes are affected by the mechanism of action-reaction that drives people to search/avoid eye contact depending on the context of the situation and behaviors of other people involved. In particular it is interesting to see what happens during pause or clause junctures, in other words points when there are a temporary breaks of the flow.

Analyzing how people interact, several studies agree on some rules applicable during conversation [9]. Listeners, showing that they are interested and paying attention, look at the speaker more than vice versa; in fact the speaker looks away for most of the time. Obviously, changing their roles, people change also their behavior. A speaker tends to gaze at the other person at the end of long utterances. Listeners always look at the speaker when he or she is about to start talking. The mean time a participant looks away is about four seconds. In pauses there is a brief mutual gaze that indicates the change of roles.

These considerations are useful to become a good conversant or to build credible virtual conversations. Naturally, times and behaviors are affected also by personalities, changes of topic relationships and naturally there are stochastic factors to take into account. [4] suggests that we should consider also psychological subcomponent. For example the speaker can have different reactions if the question is embarrassing or personal. *Gaze aversion* (or inhibition of gaze, as we said before) is a typical way to send messages about level of emotionality and feeling.

4.6 Before a Conversation

If sometimes conversing represents a difficulty, starting a conversation could be even worse. Especially with acquaintances or people that we have met just one time, we are not completely sure about their intention to have a conversation or not. Eye communication can be fundamental in these situations.

Before approaching someone, usually people wait to have at least one eye contact because of the fear of being ignored or rebuffed. Sometimes the waiting state could be infinite and after or before they need to decide if it is better to attempt or give up. In this case people should be able to guess if the other individual is not interested or simply if he or she has not perceived them. In most of the cases they decide to give up unless the relationship is strong enough to guarantee no possibility of refusal. After the approach and the mutual gaze there is the salutation, the phase of the initial greetings before the real conversation. Social norms suggest that if the distance between people is great, there is a brief initial mutual gaze but then people do not look at each other until they are close enough for the greeting [18]. Before the greeting there are other two or three seconds of mutual gaze. In particular the person that covers more ground tends to avoid as much as possible other eye contacts. A possible reason about this particular behavior is that a great distance involves obstacles that can impede a continuous mutual gaze; actually a more probable explanation is that a continuous staring produces an invasion of the personal space before the interaction.

[8] is an interesting work about observation of behaviors of two people placed in a room. Dividing their actions in four segments and analyzing gazes in each of them it is possible to see how eye-behavior affects a potential future conversation. In particular they found a specific segment in which inappropriate glances invade the privacy of others and prevent any interactions. On the other hand, eye contact in precise moments encourage the conversation. Their data are based on a statistical analysis. Our eyes are also able to transmit information to the brain about positions, actions and reactions of other people. Unconsciously the brain always stores data received by visual sensors and it uses them to model our behavior. For example, if we "perceive" that two people are approaching each other, we avoid to hamper their encounter or if someone is running toward us, we react avoiding the contact as well. It is a sort of **prediction** of future facts provided by the visual perception; our imagination builds a probabilistic model based on motion paths and interpretation of signals. As explained in [14] and [5] temporary occlusions or breaks of perception do not impede our estimation; processing approximated information, there is no way to be sure that our interpretation is correct.

Chapter 5

Approach

Having acquired numerous notions about path planning, human territories and behaviors, we tried to transfer our new knowledges into a virtual world simulating believable interactions and locomotion of virtual humans. In this chapter we want to provide an exhaustive description of our approach highlighting the most interesting features.

Before going into details of our work, we need to explain again the initial situation of the problem: there is a virtual environment where several agents navigate, perceive and interact. The time is discrete; for each iteration positions of agents are updated considering a velocity vector, derived from their decisions. Potential obstacles that hamper movements are walls (static) and other virtual humans (static or dynamic). They can generalize any type of situation so we do not need other types of obstacles to develop a general path following sub-system.

For simplicity, we assume that movements are always on a two dimensional plane; agents and walls are represented by discs and rectangles. It is like watching the scene from the top.

Building a uniform grid, taking into account positions and sizes of walls, the path finding is solved through the Theta* algorithm, described in the second chapter. This represents the starting point of our social path following.

5.1 Prior Work

As we said in the second chapter, our project represents an extension of a prior work proposed in [13] by Ioannis Karamouzas and Mark Overmars of the University of Utrecht. It consists of a velocity-based model to solve the path following simulating human mental processes, activated during locomotion. It is a general, flexible and extremely realistic

approach that we have improved in the context of social environments.

It is possible to summarize this path following algorithm in three main steps:

- retrieve colliding agents;
- define the range of admissible speeds and orientations;
- select the optimal combination.

In next sections we will explain how to address each sub-problem and the variations that we introduced in our project.

5.1.1 Colliding Agents

The first step requires that each agent is able to perceive potential colliding obstacles along its path. Fortunately CADIA Populus, the framework that we are using to develop the social path following, is already powered with a sophisticated perception system. As described in [21], each agent collects data about the environment through a sense of *vision* and *proximity*. From social theories, the proximity is structured in four concentric areas considering intimate, personal, social or public zone (Figure 5.1). Agents are also powered with visual sensors (central and peripheral) with different levels of detail, simulating a human-like perception.

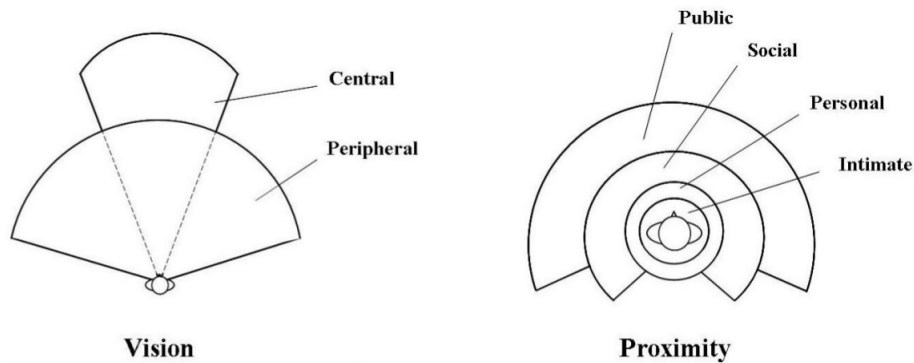


Figure 5.1: Different perceptions already in CADIA Populus [21]

People perceive obstacles through their eyes, that represent human visual sensors. At the same time, we need to ensure always no unperceived colliding obstacle. Considering these reasons the **peripheral vision** is the best solution for our purpose.

From perceived virtual humans we extract a sub-set of colliding agents. Knowing velocities, current positions and radii (derived from its k-spaces), it is possible to predict future positions and check collisions. In particular for each couple of agents A_i and A_j , we can

use the inequation of Euclidean distance and sum of radii, proposed also by Reynolds, as follows:

$$\|(x_j + v_j t) - (x_i + v_i t)\| \leq (r_j + r_i) * incr$$

where x , v , t , r represent respectively positions, desired velocities, time and radii. The radius of a k -space is too small to avoid a collision before the contact; at the same time, the *personal space* of each agent is disproportionately large to represent its location. For these reasons we increment each radius using a variable *incr*. The default value is 1.5 but it depends on the personality of each agent that might want to preserve more/less distance from others.

The formula contains an unknown variable, the time. Actually, for each couple we compute the first future time of contact evaluating if it belongs to the "near" future. This means that its value should be positive and less than an certain amount of time, called tc_{max} . We use about 8 seconds. Note that extracting the time from the formula, it becomes continuous and not discrete.

Computing the time consists of solving a quadratic equation. Accordingly, we can obtain two different solutions. We want to point out again that we need the **first** (minimal) **future** (positive) **time of contact**. The picture (Figure 5.2) shows two different contacts between discs. Predicting future positions, we can see that the first collision will happen after two iterations and the second one after three. Imaging movements, it is easy to see an overlap of shapes exactly between the two times of contact. In conclusion, we are interested in the first collision, because avoiding it, we can also avoid overlaps. From the figure on the right, we can also deduce that not always the nearest neighbor represents the first colliding agent.

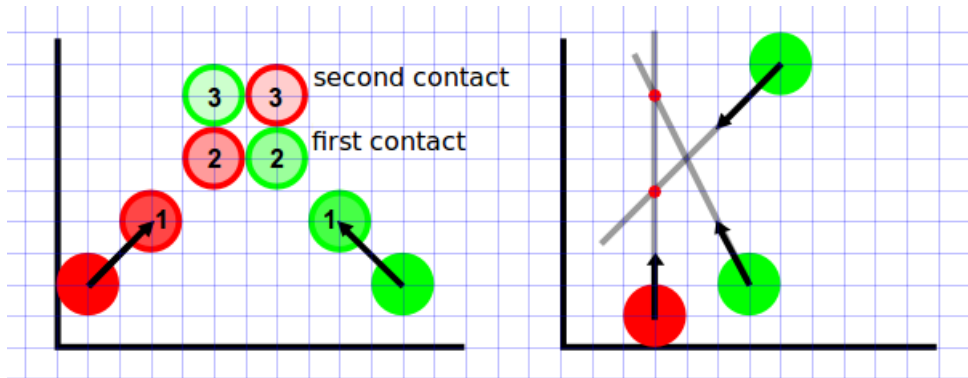


Figure 5.2: Progressive movement of agents represented by discs, highlighting two times of contact; on the right there are paths and future contacts

Someone could object by saying that, to compute the first contact, we could just solve the linear system of equations that describe the paths of the two agents. After finding its

solution, it should be possible to get the time as well. Sometimes this technique provides the same result as ours but not always. Looking at the second case of the picture below (Figure 5.3), the system of linear equations should find a contact in $[6,2]$. Actually this collision point does not exist; in fact while the red agent is reaching that position, the green one has already avoided the contact going upward.

Two agents with the same or opposing velocity vectors can produce infinite/negative values of time or no solution. Of course, our algorithm should be robust enough to manage any kind of situation, considering possible determinant less than zero or denominator equal to zero of the quadratic equation (Figure 5.3).

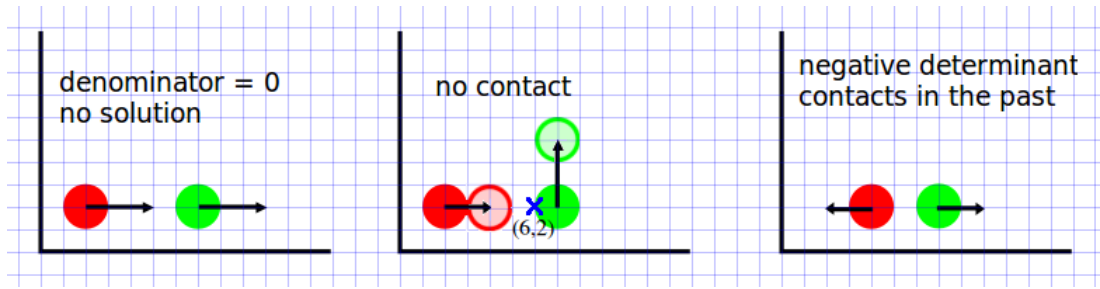


Figure 5.3: Particular situations to manage: same velocity vector, "false" collision, contact in the past

Having found the set of perceived-colliding agents, it must be sorted in order of increasing collision time. Furthermore the number of items should be kept under control, especially in crowded simulations. [13] suggests to save five agents reflecting human behavior and improving the performance at the same time. Our approach is able to decide the size of the set dynamically: it depends on the total number of agents inside the environment.

5.1.2 Range of Speeds and Orientations

The goal of the second step of the algorithm is to determine the set of speeds U_i and orientations O_i , for each agent A_i that needs to avoid future contacts.

| | | |
|--|--|--|
| $\Delta\theta_i^{\max}(tc) = \begin{cases} \frac{\delta_{\max} - \delta_{mid}}{e^{tc}} + \delta_{mid}, \\ \delta_{mid}, \\ \delta_{mid} \frac{tc_{mid} - tc}{tc_{\max} - tc_{mid}} + \delta_{mid}, \\ 0, \end{cases}$ $O_i = \{\mathbf{n}_\theta \mid \theta \in [\theta^{des} - \Delta\theta_i^{\max}, \theta^{des} + \Delta\theta_i^{\max}]\}$ | $U_i = \begin{cases} u \mid u \in [0, u^{\max}], \\ u \mid u \in [u^{pref}, u^{pref} \pm \Delta u_i^{\max}], \\ u \mid u \in [u^{pref}, u^{pref} \pm \Delta u_i^{\max}], \\ u^{pref}, \end{cases}$ | $\begin{cases} \text{if } 0 \leq tc < tc_{\min} \\ \text{if } tc_{\min} \leq tc < tc_{mid} \\ \text{if } tc_{mid} \leq tc \leq tc_{\max} \\ \text{if } tc_{\max} < tc \end{cases}$ |
|--|--|--|

Figure 5.4: Heuristic functions to compute the range of speed and orientation [13]

This set will be used to build velocity vectors to resolve or at least postpone collisions.

Their wideness depends on the time of contact with *the most threatening agent* (the first colliding agent saved in the list during the first step).

Following exactly the instructions provided in [13] we can create sets of speed and orientation (Figure 5.4). Most of the variables in these formulas are constants in the prior work and attributes of agents in our extension. We will explain their meaning in the section about profiles. Other variables are: t_c which represents the time of contact with the most threatening agent; Θ^{des} and u^{pref} are respectively the desired orientation and speed; u^{max} means the maximum speed for agents that, in CADIA Populus, has a value of 0.7; finally Δu_i^{max} is the minimum of u^{pref} and the difference of u^{max} and u^{pref} .

Summarizing the idea behind the formula, the more the impact is imminent, the more ranges increase. To do this, the first time of contact is compared with different amounts of time ($t_{c_{min}}$, $t_{c_{mid}}$, $t_{c_{max}}$), reducing/increasing intervals.

We can provide a better explanation of this step describing a particular case study. Consider that each variable has an assigned value, as showed in the second table of the next page; the table on the right shows the calculated orientation and speed ranges for different times of contact (Figure 5.5). The first row describes a situation in which agents are very far and they do not care about each other. Ranges are extremely limited; the unique possible combination of orientation and speed is equal to the desired velocity. From the second case, agents start to consider the possibility of collision. If there is still enough time before the contact, each agent generates a small set of candidates trying to avoid the collision and hoping that the other is doing the same, exactly as humans. Until the collision is solved, ranges become wider achieving an amplitude of 104 degrees in total for the orientation and any possible value of speed (0.0, 0.7), taking into account the maximum limit related to u^{max} .

5.1.3 Optimal Velocity

The width of ranges is expressed just with their extreme values, that represent the output of the previous step. The problem is that between two extremes there are infinite numbers of speeds and orientations and accordingly infinite feasible candidates for a new velocity vector. For this reason, the first task of this step consists of discretizing the ranges, as showed in the simplified picture below (Figure 5.6), with three possible speeds and three orientations.

It is a tricky phase of the algorithm: making mistakes at this point means destroying the realism or on the other hand slowing down the execution significantly. To find a good balance, [13] suggests to use a discretization step (distance between adjacent samples)

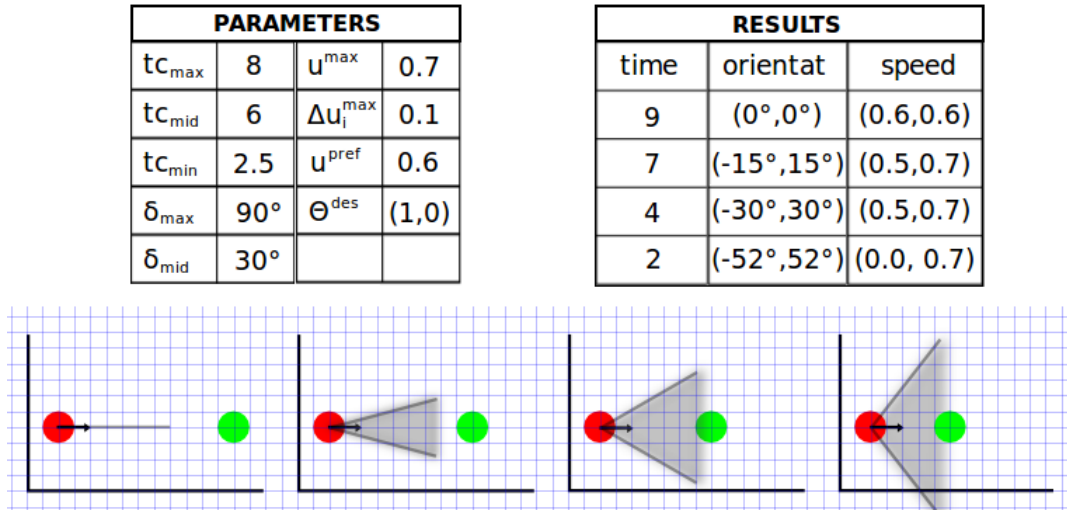


Figure 5.5: Width of ranges related to the first time to contact. Tables contain parameters used in the algorithm and geometric details of the picture

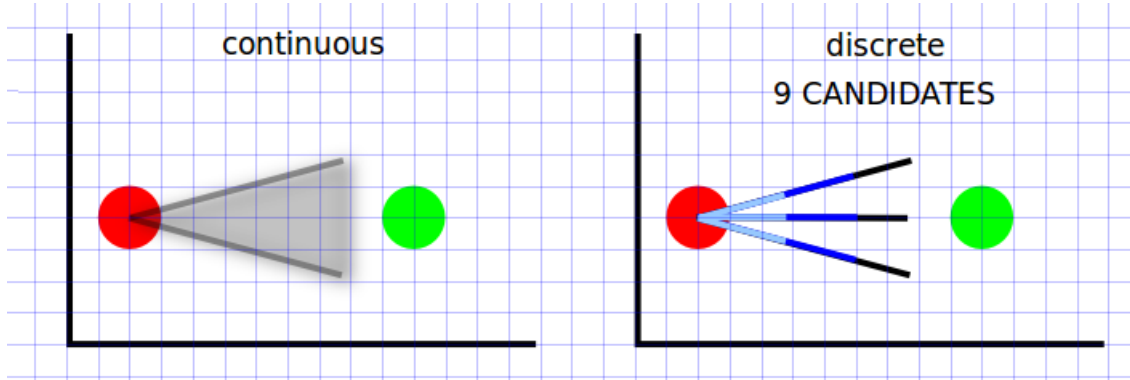


Figure 5.6: Continuous and discrete range of velocities

of 0.078 radians to the set of orientations and 0.1 to speeds. In our project these values depend on the total number of agents and the complexity of the environment, based on the length of the list that contains colliding agents.

For each possible couple of samples, we can generate a candidate velocity, called v^{cand} , multiplying the normalized vector of orientation by the scalar that represents the speed. Each candidate is evaluated using a particular heuristic function that includes:

- **energy consumption**, considering the previous velocity vector v_i of the agent;
- **deviation angle** between candidate and desired velocity;
- **risk of collision** with colliding agents in the list.

The complete formula (Figure 5.7) shows that all parameters (energy, deviation and collisions) are assigned a weight, given by α , β , γ and δ ; changing them it is possible to

set another order of priority and to simulate diverse types of behaviors. These variables represent other attributes of each agent in order to create customized profiles.

$$\mathbf{v}_i^{new} = \arg \min_{\mathbf{v}^{cand} \in FAV_i} \left\{ \underbrace{\alpha \left(1 - \frac{\cos(\Delta\phi)}{2}\right)}_{\text{Energy}} + \underbrace{\beta \frac{|\|\mathbf{v}^{cand}\| - \|\mathbf{v}_i\||}{u^{\max}}}_{\text{Energy}} + \underbrace{\gamma \frac{\|\mathbf{v}^{cand} - \mathbf{v}_i^{des}\|}{2u^{\max}}}_{\text{Deviation}} + \underbrace{\delta \frac{tc_{\max} - tc}{tc_{\max}}}_{\text{Collisions}} \right\}$$

Figure 5.7: Formula to compute the cost of each velocity [13]

The best velocity v_i^{new} is the candidate that minimizes the cost of the heuristic function. It represents the new vector assigned to the agent that should help to produce a fluid and natural path.

This is an exhaustive explanation of the third step described in [13]. Our approach is based on it adding some improvements:

- we take advantage of the flexibility of the heuristic function to develop a social path following based on prediction of interactions;
- to avoid too rigid and mechanical behavior the discretization is affected by possible distortions, basically stochastic factors;
- there is a shared protocol for the collision avoidance, known to all agents, to facilitate movements.

5.2 Social Awareness

A social path following consists of a path following procedure, able to solve the collision avoidance but improved to simulate real social interactions and/or extended to develop more human-like behaviors. These new features produce corrections of locomotion, making the environment more credible and effective.

In our project we powered our virtual humans with the skill of predicting future social interactions. Through a probabilistic model that analyzes typical signals before conversations,

each agent can identify future conversants while they are approaching each other and avoid interrupting/hampering their interaction. Agents are also able to see a conversation as a single-composed obstacle to avoid, without going through it, respecting human territories. We also apply distortions and rules to emphasize the realism. All these features represent a specific behavior called **avoidHumanObstacle**, our extensions to the state of the art of path following behavior from [13].

Using projections of positions, a moving agent can check if there is enough space to continue walking along the current path. In particular, in stuck situations due to conversation groups cluttering the space, it alerts the blocking group, using gaze. On the other hand, people in those groups interpret the gaze and react accordingly. As result we obtain two complementary behaviors that we call **alertHumanObstacle** and **clearHumanObstacle**. It is important to highlight that these three behaviors are completely independent and they cannot generate conflicts or incoherences. First of all, they are used at different times by subjects: **avoidHumanObstacle** and **alertHumanObstacle** are active in moving agents while **clearHumanObstacle** applicable only to standing agents. Secondly, since **alertHumanObstacle** does not generate correction of motion, it can run in parallel with the path following behavior.

5.2.1 Memory

Especially to predict future conversations, visual perception is not enough. To have a higher level of knowledge, each agent needs to collect data about other virtual agents. For this reason we developed something similar to a **memory**. It is a container of information that each agent owns and updates during the execution of the system. It is completely independent of other behaviors; in fact it should be kept apart regardless of whether they are active or not.

Avatars need to store *conversations*, *walls*, other *agents* and their *positions*. Since the sub-system of perception in CADIA Populus cannot manage walls and conversations yet, we save them, in a shared data-structure between all memories, at the beginning, when the simulation starts. It means that agents cannot perceive conversations generated dynamically during the execution; they are not visible but being a compound obstacle, they will be avoided anyway. Agents and their positions are collected in separate lists as well; they are used in order to build their paths and to predict their future intentions. These data are the minimum amount of information needed to perform our behaviors; for example we do not need to save previous velocities of each agent, because it is a derived data, computed as difference of current and previous positions.

Exactly as humans, agents store/update information about only perceived things. To reduce

the computation, the update of memory is not done every frame but frequently, at regular intervals. In this way we can also simulate the human mental process. Finally agents need to know only the recent past; positions of few iterations ago are enough to predict motions, approximately. It is also done in order to save memory and to make the system less resources consuming.

5.2.2 Territory avoidance

Hampering a future encounter between two people while they are approaching each other is not a socially acceptable behavior. Perceiving and understanding intentions, polite people wait for the encounter to start before moving or at least they deviate their trajectory, avoiding to become a disturbing obstacle. Of course this particular behavior, callable **territory avoidance**, requires several skills such as reconstruction of movements based on memorized data, right interpretation of body language and coherent reaction.

In our project we have developed a probabilistic model that simulates similar interactions. Being a behavior that affects the locomotion, it consists of a refinement of the path following. In other words, it is not a new behavior (implemented separately) but it extends the general path following procedure. In this way, we are sure of avoiding conflicts and oscillatory motions.

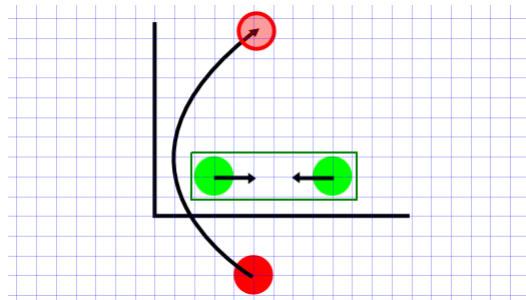


Figure 5.8: Correct social behavior perceiving an approaching space

In our project we represent the approaching space through a thick segment whose endpoints are computed considering positions of the future conversants, as we can see in the picture (Figure 5.8). It is a *probable invisible obstacle* for the path following algorithm. Being probable and invisible, avoiding it has less priority than avoiding a real obstacle.

First of all to build an approaching space, each virtual human filters perceived agents considering distances; in fact, we assume that future conversants must be close enough to generate a relevant approaching space, exactly as in the reality. “Close enough” means that we fixed a maximum distance to discriminate potential couples of interlocutors. This represents the first layer of filtering.

The second layer is modeled as a probabilistic evaluation on interpretation of behaviors. As we said in previous chapters, typically an encounter is preceded by *approach*, *mutual gaze* and *salutation*. Obviously these actions do not always all occur before any encounter. For example if a person has not perceived the other that is approaching, there is no mutual gaze. Or an interaction between two people already close is not preceded by the approach. Moreover there are factors more relevant than others; for instance, the salutation preannounces the conversation more than the mutual gaze.

From all of these considerations, we have built a model that describes the relation cause-effect between behaviors and conversation. Through it, each agent is able to compute a probability of conversation from a particular sequence of factors and to react accordingly. The probability is related to how much agents are sure about the future interaction of others. The table below summarizes our model.

| Approach | Mutual Gaze | Salutation | Probability |
|-----------------|--------------------|-------------------|--------------------|
| True | True | True | 1.0 |
| True | True | Unknown | 0.7 |
| True | Unknown | True | 0.8 |
| True | Unknown | Unknown | 0.1 |
| Unknown | True | True | 0.95 |
| Unknown | True | Unknown | 0.15 |
| Unknown | Unknown | True | 0.3 |
| Unknown | Unknown | Unknown | 0.0 |

A similar model allows us to manage any situation. Our first idea was to use a *Bayesian networks* to represent probabilistic relationships among variables of interest, reducing the number of cases. Since Bayesian networks are based on independencies of variables and we could not easily describe the causal relationships among gaze, approach and salutation, we discarded that alternative.

From our basic model an issue comes out: because of the salutation occurs just before the conversation, being the last preliminary phase, during evaluations made by agents, it has assigned an *unknown* value for most of the time. In other words, being a "future" event, agents can really verify the salutation too late, just when the collision is imminent. For this reason, despite there not being a real dependence among variables, while its value is still *unknown* we flip it to *true* with a certain probability given the approach. During the locomotion, agents monitor potential couples of conversants (already filtered) and update probabilities of conversation according to changes of state, as follows.

Approach

Monitoring virtual humans that are approaching each other, each agent needs to use its memory; the process consists of build paths, given their current and previous positions. In particular, positions represent points in space and the path is basically a straight line computed through **linear regression**.

Paths of agents that are moving without encountering obstacles, are simple lines that contains starting points and current positions. It seems simple but in case of collisions, real intentions of agents are hidden because of corrections of paths produced by the collision avoidance. Accordingly the linear regression process is an essential requirement: it creates a line that approximates motion, reducing effects of the collision avoidance as much as possible.

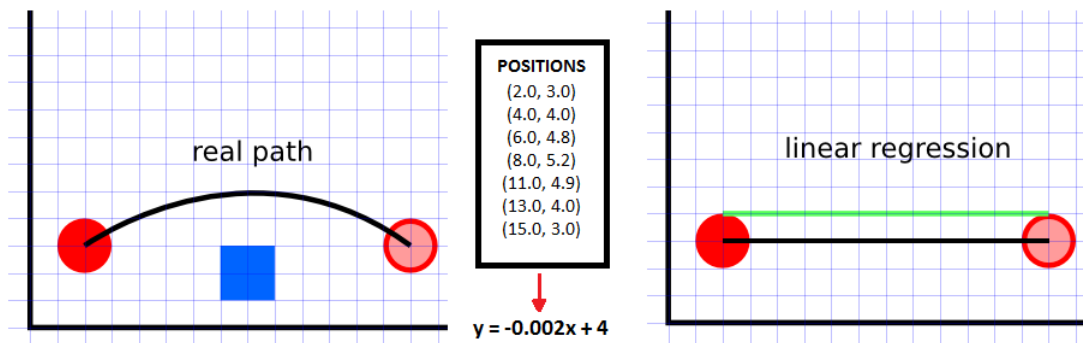


Figure 5.9: The first picture shows the path followed by the agent, avoiding the obstacle; in the second one we show the difference between the desired movement of the agent, in the absence of obstacles (black line) and the linear regression result (green line) derived from the path in the image on the left

The picture (Figure 5.9). shows the benefit of the linear regression. The green line on the right is computed as approximation of the path on the left using the table that contains intermediate positions of the red object. As we can see the approximated line is really similar to the path in absence of obstacles. It means that the linear regression can extract desired movement from a certain amount of data; of course it should be enough to provide good approximation and to maintain the efficiency at the same time. This is the task of the memory.

After computing all paths, we check if they are generating an approaching space of a future collision. In particular there are two different situations to consider:

- one agent is still standing and the other is approaching;
- two agents are approaching each other.

In the first case we just need to check if the position of the standing agent belongs to the approximated path of the other. In the other situation, the two paths should be similar but polar opposite. Of course the comparison should be flexible, taking into account small differences due to collision avoidance.

Mutual Gaze

Mutual gaze means simultaneous eye contact. Each agent should be able to check if two agents are looking at each other, through its perception, estimating its duration. Note that it may also be a non-continuous contact.

Following social theories, the real mutual gaze before an encounter is an eye contact that persists for two or three seconds. In our project we decrease its duration to just one second because it is more adequate to other times of the current system and in order to assign a right weight to the gaze factor. In fact, setting two or three seconds, in most of the simulations, the territory avoidance is ruled only by the approach-state (waiting two/three seconds, the contact with the approaching space becomes imminent and inevitable). The duration is just a simple parameter that we can tune as we see fit.

The literature explains that some encounters are preceded by a particular form of gaze, called **intimate gaze**, which covers the upper part of the body and does not imply eye contact. It is not taken into account in our approach. We have simplified the problem considering only the mutual gaze; this behavior is already implemented in CADIA Populus.

Salutation

The step immediately prior to any conversation is the close salutation. It is recognizable from specific gestures, smiles and few words exchanged between people and represents the last preliminary phase. For this reason, most of the time, in our approach it is simulated by an additional probability, as we said before.

Since CADIA Populus does not yet include particular animations and features about greetings, it is almost impossible to perceive that phase during virtual interactions. To solve the problem, we assume that the close salutation is always related to a reduction of velocity considering the locomotion of agents. Approaching each other, their speeds should start to decrease until they stop and initiate the conversation.

Our assumption can produce the wrong interpretation. Sometimes dodging an obstacle produces a reduction of speed that is completely unrelated to the salutation. To avoid

errors, we consider the salutation a *continuous, progressive* reduction of speed that persists for many iterations.

Reaction

Every time there is an update of approach, mutual gaze or salutation, a new probability of conversation is assigned to each couple of agents, using the table showed before. Generating a random value or using a threshold value (depending on the initial setting) and comparing it with the computed probability, the current agent decides whether to consider them future conversants or not. This produces a new list of couples, completely different from the previous one; we do not need to monitor them considering all preliminary phases, being confirmed couples.

For each confirmed couple of agents, we generate their approaching space. First of all we compute the path between them, represented by a linear equation. Secondly, finding the time of contact between the current agent and the path, we can attempt to predict future position of agents on impact. They are the extreme points of the approaching space, as we can see in the picture (Figure 5.10).

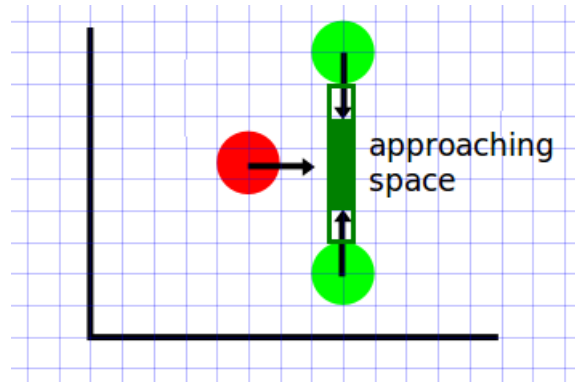


Figure 5.10: Representation of the approaching space considering future positions

The picture shows that actually the approaching space should be a bit bigger, considering also the radius of agents, not just positions, in order to build new effective obstacles.

Now we are able to extend the prior work. Having other obstacles to avoid, we add a new cost to the heuristic function, as we can see in the next page (Figure 5.11). It is similar to the risk of collision with real objects with few differences. Firstly η has assigned a lower weight than δ ; its default value is 0.8. In this way agents prefer always to avoid real obstacles than approaching spaces; in critical situations it is extremely important to make the right decisions. Secondly, since $dist_{max}$ is the maximum length of any approaching space and $dist$ consists of the distance between future position of the current agent and point of contact, this piece of the formula is an additional weight to produce a particular

effect: the larger the approaching space the less agents care about not crossing it, because the violation of private space is not as relevant. The last part follows the intuition of the prior work that, through the time, produces realistic motion. In particular $tcAS$ represents the time of contact between subject and approaching space.

$$\begin{aligned}
 \mathbf{v}_i^{new} = \arg \min_{\mathbf{v}^{cand} \in FAV_i} \{ & \overbrace{\alpha(1 - \frac{\cos(\Delta\phi)}{2}) + \beta \frac{|\|\mathbf{v}^{cand}\| - \|\mathbf{v}_i\||}{u^{max}}}^{\text{Energy}} \\
 & + \underbrace{\gamma \frac{\|\mathbf{v}^{cand} - \mathbf{v}_i^{des}\|}{2u^{max}}}_{\text{Deviation}} + \underbrace{\delta \frac{tc_{max} - tc}{tc_{max}}}_{\text{Collisions}} + \underbrace{\eta \frac{dist_{max} - dist}{dist_{max}} \frac{tc_{max} - tcAS}{tc_{max}}}_{\text{PathAvoid}} \}
 \end{aligned}$$

Figure 5.11: New formula to compute the cost of each velocity, taking into account the territory avoidance

Sometimes it happens that the impact with an approaching space is imminent. Being an "soft" obstacle, we prefer that agents cross it directly, instead of moving around. This particular feature is taken into account in our approach before using the formula.

5.2.3 Negotiation of Space

Especially in crowded or small environments, human beings act in order to prevent themselves and others from getting stuck. A typical example is the behavior of two opposite flows of people that need to use the same gate simultaneously: each person waits or moves according to a brief negotiation with other individuals, trying to balance protection of spaces/needs and a prevention of a total jam.

People negotiate space in several different ways. Usually the negotiation occurs with non-verbal communication transmitted through gaze and body language. A negotiation can be seen as a particular type of territory avoidance because it implies a prediction of intentions, evaluation of human territories and reactions.

In our project we focus on the behaviors of standing (conversing or not) and moving agents in tight places. In particular we are interested in what happens when there is not enough space to reach a destination because of human obstacles. We have developed two different behaviors associated to passive and active subjects, trying to simulate this particular form of interaction.

Active Behavior

The active behavior is activated by virtual humans that are moving around the environment. They should be able to perceive if a particular movement could produce stuck situations and to react consequently requesting more space through non-verbal signals towards standing and/or conversing agents. We call this behavior **alertHumanObstacle**.

Imagine that we need to go through a narrow passage. To check if there is enough space, our mental process creates a projection of ourselves along the path, finding inevitable overlaps of projection and objects. We have developed something similar: working in two dimensions and using discs that represent the minimum space of bodies, each agent evaluates portions of the environment considering positions of walls, other agents and conversations. Only stationary obstacles are taken into account during the computation.

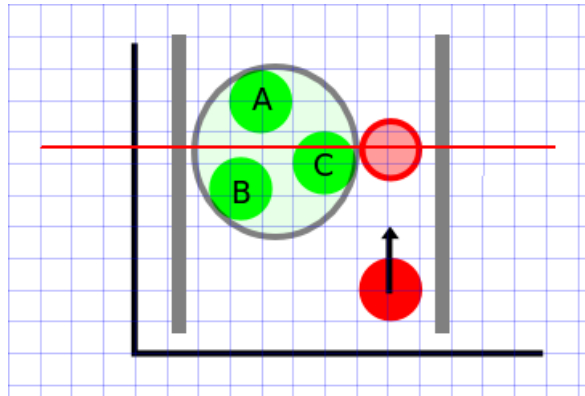


Figure 5.12: Scenario related to the negotiation of space

In the picture (Figure 5.12), we can see a corridor bounded by lateral walls and partially occupied by a group of agents. Knowing its path, the red avatar is able to perceive potential colliding obstacles and critical zones. After computing the nearest point of contact, the perpendicular line is used to find overlaps between the projection and any obstacle. In this situation we can see that there is enough space to pass since the projection does not collide with the wall on the right, so the agent does not need to request space.

Studying human interactions, usually people that are joining a conversation exchange few glances with the group and start looking at the spot (the center of the conversation space). A completely different behavior is associated with passersbys that request more space: they gaze continuously at the individual that obstructs its movement forcing him/her to move out of the way. This would be the reaction of the red avatar if in the previous example there was not enough space.

Passive Behavior

On the other hand, we need to model the behavior of passive subjects. Standing agents need to perceive if they are potential obstacles for moving agents and to prevent stuck situations.

The **clearHumanObstacle** behavior is quite simple: after finding collisions between paths (predicted using information in the memory) and its personal space, each standing agent starts monitoring the actions of others. In particular, perceiving a continuous gaze from a moving agent, the other understands the request and reacts by increasing the space dedicated to the passageway. If in the picture (Figure 5.12) there was not enough space for the red agent, the agent C would move on the left, rearranging the whole conversation. Standing agents react just when the moving agent is close enough to the human obstacle because early reactions can produce a new reduction of space, due to the rearrangement of conversants. This adds a realistic effect to the simulation by increasing the level of social cooperation.

5.2.4 Other improvements

As we said in the introduction of the section, in our version of the path following we have introduced particular features that should increase the realism and improve the performance.

First of all, agents can decide to reduce/increase the amount of information that affects their choices. We are talking about the size of the list that contains nearby obstacles and the discretization of ranges computed during the third step of the algorithm. According to properties of the environment and the total number of agents, automatically the system balances realism and performance adjusting some parameters.

In particular, the list of neighbors has a minimum length of five agents; it is a good value for relatively simple worlds with a limited number of agents. In complex environments, making right decisions becomes harder and agents need more information. Thus, increasing the complexity, the size of the list increases as well. But to maintain good performance the rate of increase should be *less than proportional*.

The discretization depends on the size of the list, they are *inversely proportional*. Since increasing its length makes the computation more time consuming, we need to reduce the number of candidates; it means a wider discretization step.

The path following algorithm generates a uniform set of candidates with regular intervals, given ranges of speed and orientation computed in the second phase of the procedure. In fact the discretization step is constant. This property can cause too rigid and mechanical

movement of agents: it ensures a realistic movement that sometimes seems too perfect and unnatural considering human beings. It is possible to solve the problem, through stochastic changes of speed and orientation that widen/restrict slightly the discretization step (Figure 5.13).

Little alterations of motion, that we call **distortions**, should also facilitate the negotiation of motion. Imaging two agents facing to each other with the opposite desired velocities. Exactly as humans, perhaps they need some iterations to solve the collision because of specular decisions, especially if they have a similar profile. Distortions can make agents more/less determinate, making the collision avoidance easier.

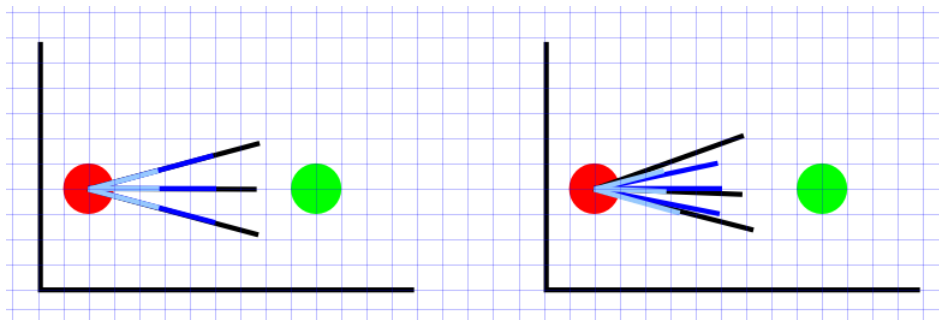


Figure 5.13: Different velocity vectors produced without and with distortions

Obviously the rate of distortions should be limited, avoiding overlaps of discretization steps and problems with the algorithm. For this reason its default value represents the ten percent of the step. Each agent can specify the rate of distortions in its profile.

Movements of people are ruled by particular protocols. For example they prefer to avoid an obstacle going on the right instead of dodging it on the left. Our social path following takes into account this particular rule. There is a specific order of processing candidates in the third step of the algorithm: we start with the desired velocity, we process all candidates on the right from the center to the border and finally we apply the same principle with left candidates. We are giving a specific priority to them; maybe in this way it is possible to avoid the computation of the deviation cost. Another possible improvement in the future is a policy of saving time that reduces the number of candidates to process. For instance if certain orientation and speed produce a collision, any increase of speed combined with the same orientation must produce the same collision, so we can avoid to process this candidate.

Sometimes, especially in crowded environments, agents do not need to consider social rules. In our approach it is always possible to avoid prediction of intentions and to use the general path following formulation. The algorithm evolves depending on the amount of information stored in the memory of each agent.

5.2.5 Profiles

More than once we highlighted that our approach supports generating several stereotypes of people. Each agent has a profile that contains specific attributes. Although we use the same algorithm for every subject, adjusting these parameters it is possible to simulate different types of locomotion. It is one of the best features of our approach.






| | | | | | | | | |
|---|------------|-----|----------------|-------------|----------|------|----------|-----|
|  | tc_{max} | 8 | δ_{max} | 90° | α | 1 | δ | 1 |
| | tc_{mid} | 6 | δ_{mid} | 30° | β | 0.05 | η | 0.8 |
| | tc_{min} | 2.5 | incr | 1.5 | γ | 1 | dist | 0.1 |
|  | tc_{max} | 8 | δ_{max} | 50° | α | 1 | δ | 0.1 |
| | tc_{mid} | 6 | δ_{mid} | 20° | β | 0.05 | η | 0.1 |
| | tc_{min} | 2.5 | incr | 1 | γ | 3 | dist | 0.1 |
|  | tc_{max} | 12 | δ_{max} | 150° | α | 1 | δ | 4 |
| | tc_{mid} | 6 | δ_{mid} | 30° | β | 0.05 | η | 0.8 |
| | tc_{min} | 5 | incr | 3 | γ | 1 | dist | 0.1 |
|  | tc_{max} | 5 | δ_{max} | 90° | α | 1 | δ | 1 |
| | tc_{mid} | 3 | δ_{mid} | 30° | β | 0.05 | η | 0.2 |
| | tc_{min} | 2.5 | incr | 1.5 | γ | 1 | dist | 0.8 |
|  | tc_{max} | 8 | δ_{max} | 120° | α | 0.5 | δ | 1 |
| | tc_{mid} | 6 | δ_{mid} | 60° | β | 0.02 | η | 0.1 |
| | tc_{min} | 2.5 | incr | 1.5 | γ | 1 | dist | 0.1 |

Figure 5.14: Several profiles of agents

This is the complete list of attributes that belong to agents:

- **alpha** (α): weight of energy-consumption computed as difference between the current candidate velocity and the previous movement of the agent.
- **beta** (β): weight of energy-consumption computed as difference between the current and previous speed.
- **gamma** (δ): weight related to the risk of collision considering times of contact.
- **eta** (η): weight of the territory avoidance taking into account prediction of future conversation.

- **tcMax** (tc_{max}): it represents seconds. Each agent starts to consider a collision if the time of contact is less than tcMax. In particular when it is between tcMid and tcMax, the collision has a low priority.
- **tcMid** (tc_{mid}): it represents seconds. Any collision with a time of contact between tcMin and tcMid has a medium priority.
- **tcMin** (tc_{min}): if the time of contact is less than tcMin, collisions have high priority. Exactly as previous attributes, its value represents seconds.
- **deltaMid** (Δ_{mid}): medium width of angle computing the range of orientations.
- **deltaMax** (Δ_{max}): maximum width of angle. It is used with imminent collisions.
- **incr** ($incr$): percentage increment applied on the radius of each agent for calculating a intermediate space between k-space and personal space.
- **distortion** ($dist$): weight related to stochastic factors that affect the motion.

The picture of the previous page (Figure 5.14) shows five examples of different stereotypes of agents. The first one is the default profile. Most of these parameters are indicated in [13], the others are necessary to simulate the social path following. They should generate a balanced behavior considering all costs of the heuristic function used in the algorithm. The second profile is related to people that are always late. They do not care about collisions and want to achieve the destination as soon as possible, giving a higher priority to the deviation cost. Robbers are extremely worried about collisions. They try to prevent any contact increasing the wideness of angles. There is also a huge increase of the body space.

```
<AvoidHumanObstacle>
  <alpha default="1.0" value="1.0" description="..."/>
  <beta default="0.05" value="0.05" description="..."/>
  <gamma default="1.0" value="1.0" description="..."/>
  <delta default="1.0" value="1.0" description="..."/>
  <eta default="0.8" value="0.6" description="..."/>
  <tcMax default="8.0" value="8.0" description="..."/>
  <tcMid default="6.0" value="5.0" description="..."/>
  <tcMin default="2.5" value="2.5" description="..."/>
  <deltaMid default="0.5235987" value="0.5235987" description="..."/>
  <deltaMax default="1.5707963" value="1.5707963" description="..."/>
  <incr default="1.5" value="1.5" description="..."/>
  <distortion default="0.1" value="0.0" description="..."/>
</AvoidHumanObstacle>
```

Figure 5.15: Example of a profile stored in a XML file

We can simulate drunk people, as we can see in the fourth profile. They are characterized by unpredictable and strong changes of direction associated to the distortion parameter and a reduction of perception. Finally, young people are full of energy without caring about its consumption. Usually they have a wider amplitude of motion and of course they are not expert in social behaviors.

Each profile is loaded from an XML file. The image (Figure 5.15) offers a typical example of it. Actually the real document contains the description of each parameter that we removed for reasons of space. It is possible to create/save profiles using a particular GUI included in our project. Through profiles we can accentuate particular movements of specific types of people; it produces effects on path and motion. Of course we need to include animations and special graphical features to create better simulation with a high level of believability.

Chapter 6

Implementation and Demos

Our implementation is an extension of an already existing system called **CADIA Populus**. It consists of a platform to simulate social behaviors and interactions between agents in a virtual environment. Being a first prototype, we have built a demo to show the main features of our approach in several scenarios, hoping that in the future the social path following will become an important module of the whole system.

This chapter is about main features of the implementation. Firstly we will give some details about CADIA Populus and its potential; secondly we want to describe the design of our code emphasizing particular choices and interesting properties; finally each scenario will be presented evaluating our results.

6.1 CADIA Populus

CADIA Populus, or simply Populus, is a social simulation platform and is part of the *“Humanoid Agents in Social Game Environment (HASGE) project”* funded by the Icelandic Research Fund. This research is conducted by several researchers at the *Center of Analysis and Design of Intelligent Agents (CADIA)* of Reykjavik University and it is oriented towards making believable virtual humans.

Most of the code is written in **Python** which ensures flexibility and extensibility; few functions in C++ are useful to make particular operations faster. The virtual environment is developed using **Panda3D**, an open source game engine used in several commercial products. The system also includes **NVIDIA PhysX** to perform rigid body dynamics, fast spatial queries and high performance physical simulation, especially combined with NVIDIA graphics cards.

With Populus, users can navigate around the environment controlling their agents and start

interactions with other autonomous agents simulating real conversation ruled by social behaviors. In other words it provides all possible aspects of social environments. CADIA Populus is extremely useful also for developers; creating new behavior demos is easy and does not require the knowledge of the whole system.

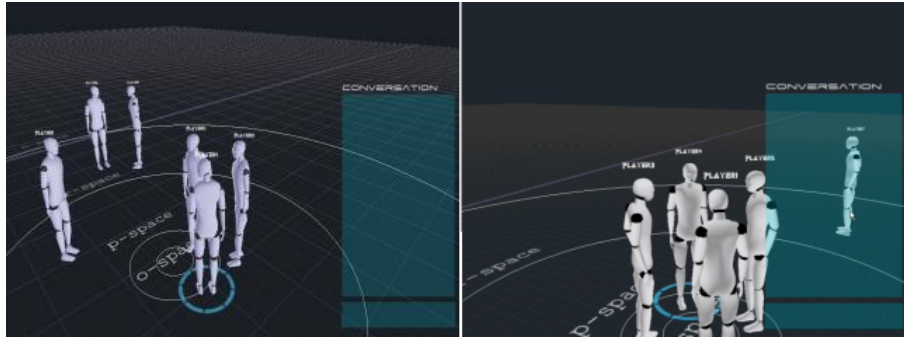


Figure 6.1: Screenshots of Populus

The main goal of Populus is to manipulate social situations providing extensible and innovative automation of behaviors. Each avatar makes decisions based on the typical perception-reaction model. Decisions are converted into motivations and then into concrete actions. Behaviors can run in parallel, be started/stopped or scheduled in order of priority, taking into account the context and the current state of the simulation.

The project is mainly focused on face-to-face conversations. It is possible to simulate how people behave considering the social territoriality and to compare results with the real world. In particular visual annotations are really useful to understand how the perception works and to highlight human territories, as we can see in the picture (Figure 6.1).

CADIA Populus is a composition of several modules. The core of the system models reactive humans, perception and behaviors especially in terms of territories and geometric computations; it is extended by specific modules related to the graphic visualization of the environment that provide gestures, animations and basic features of any simulation. Regarding our context, Populus includes particular libraries for path finding, a basic collision avoidance procedure and autonomous gaze behavior.

The whole system is currently work in progress. Constantly there are new updates and improvements made by other researchers. Our project works perfectly with one of the last versions of Populus but naturally we cannot ensure that it will continue to work properly with any future release.

6.1.1 Path Planning

The path planning problem is partially solved in Populus. Regarding the path finding, there is an implementation of the Theta* algorithm; it works through a uniform grid that represents the environment viewed from the top. It is possible to create walls as impassible areas and to generate the grid from a geometric model; after this step every agent is able to reach a destination finding the best feasible path.

The current system contains also a basic collision avoidance procedure. It is the minimum indispensable requirement to take into account dynamic obstacles. Of course it is too simple to produce realistic results in many situations. It can be replaced by the new social path following to ensure better results.

6.1.2 Gaze Behavior

A sophisticated gaze behavior is an essential requirement to develop a social path following based on prediction of intentions. Fortunately it is provided by a special work on Populus described in [7].

Collecting and analyzing videos of pedestrians, they have studied *what*, *where* and *how long* people gaze during their movements. In particular the research is focused on nine directions of gaze and reactions perceiving several types of targets. From this statistical analysis, establishing a probability of gaze and a range of duration, they have built a probabilistic model to simulate a fairly realistic gaze behavior.

6.2 Structure and Features

Developing the social path following demo in Populus, we have worked on a specific directory that contains all our code, leaving the core of the system independent. Actually, during the implementation, we have needed to add a new method in the system to disable the previous collision avoidance behavior. It does not produce any conflict with the work of other developers.

The design of our project is quite simple. To have completely separated behaviors they are developed in different classes that extend movement and skills of agents. All is modular and easily upgradable. Diverse concepts are defined by diverse classes; for example we have built special libraries to provide geometric and mathematical functions, used during the execution of each behavior. More details about each file are in next subsections.

Demo

It is the main class of our simulation containing all settings and main features of each scenario. It extends *PopulusDemo* which makes it easy to create a basic environment and to build on particular stuffs of the visual scene.

Executing our demo the initial configuration of the environment has a unique agent in the middle of the screen without visible interactions. *Demo* allows the user to run particular situations: using the keyboard it is possible to load six scenarios by pressing keys from one to six. In each of them there are a set of agents with different colors that move and interact in a socially acceptable way. Of course all environments point out main results of the social path following. Moreover, using annotations we show the complete path that each agent finds and follows.

The class has the task to update behaviors of agents for each iteration. For this reason we save them, their positions and their destinations in several lists. Being a file independent to our project (it is used just to show our results but the social path following does not need it), these collections of data are stored apart from the memory of agents.

It also provides the graphic interface to save profiles creating special types of agent.

AvoidHumanObstacle

This class contains the main behavior of the social path following, in other words the prior work extended with the territory avoidance.

In order to leave *Populus* as independent as possible, we have built a sort of **decorator** improving social attitude and skills of each agent. Decorator means a class that extends agents by updating their actions during the execution of the system. It is the best way to develop a new behavior in a prototype and to add it in the whole system in the future.

AvoidHumanObstacle is the translation of our approach in a programming language. We have developed the three main phases of the previous work and our extension. Giving a file that contains a profile it is possible to run the behavior of specific types of people, otherwise the algorithm is executed considering default values of each parameter. Thus, all is customizable depending on the stereotype of agent except settings that affect the efficiency (explained in the previous chapter) that are variables shared between all agents inside the environment.

ClearHumanObstacle and AlertHumanObstacle

Being the two complementary behaviors that produce the negotiation of space between moving and standing agents, they are developed in a unique file.

Exactly as `AvoidHumanObstacle`, they are not real behaviors but decorators and their implementations are in accordance with the description given in the previous chapter.

Memory

As we said previously, *Memory* is a container of information. Some of them are shared between agents, such as the complete list of agents, and others are owned by each agent. It should contain data about conversations, walls, agents and their positions that are stored in lists and updated using the visual perception. Being a container, most of the functions of the class are used to update/get information.

FutureConversants

Similarly to the memory, we need to store data of couples of potential future interlocutors. `FutureConversants` are objects that contain just names of agents and the approximated path between them; this is the minimum amount of information to realize any kind of operation. In fact, through their names, it is possible to get information from `Memory`, avoiding redundant data and saving space. These objects also contain the state of approach, salutation and mutual gaze that is expressed using keywords (for example `GSa` means mutual gaze and salutation without approach) and updated during the execution.

Prediction

It is a class that provides functions useful to monitor couples of agents and to manage approaching spaces. *Prediction* consists of a particular sub-module of the path following behavior that facilitates operations, being a sort of interface between behavior and collections of data. It represents the main class that associates a social attitude to the path following behavior.

LinePath

An object *LinePath* represents linear equation that approximates the path of an agent applying the linear regression. Given a sequence of positions it is possible to compute

the slope and the intercept in order to build the equation. This class provides also other functions to create a line from a couple of points, to compute the parallel line that contains a specific point, to get the normalized vector from the path and so on.

ApproachingSpace and Wall

ApproachingSpace is a sort of a dynamic obstacle to avoid, activating the social path following. It is basically a segment defined by two endpoints that are inputs of its constructor. It does not have a 3D model but it is represented by a simple annotation line that highlights its position. The color of the line indicates the relevance of the approaching space. In particular a red line expresses a path absolutely to avoid because interlocutors are very close to each other; whiter lines represent weaker obstacles.

Wall is another type of obstacle used in our demonstrations. They are represented by three dimensional models. We are using an implementation of walls that belong to previous demos made by other developers.

Other Libraries

Our project needs particular libraries that provides several mathematical functions. First of all we have implemented a class called *PrecalculatedMathFunction* that provides approximated values of sine, cosine and arccosine. It is a way to avoid expensive functions of the "math" library of Python through two lists of pre-computed values. Of course we are approximating operations that do not need precise values ensuring better performance.

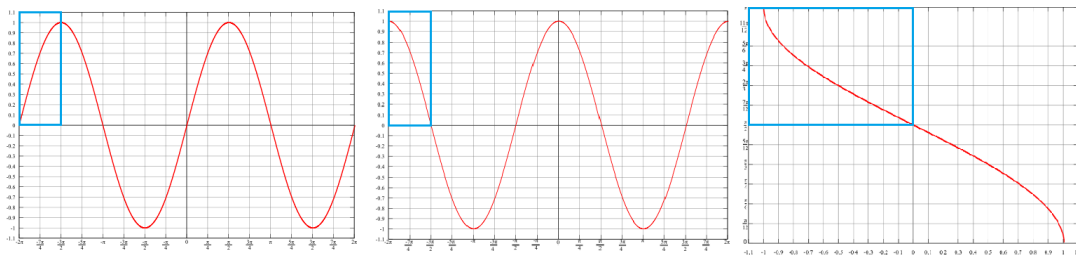


Figure 6.2: Intervals of values saved in our lists

As we can see in the picture (Figure 6.2), these trigonometric functions have sinusoids with periodic repetitions of the same wave. To manage any possible angle/value we just need to store data of small intervals. Naturally they should be discretized: we have chosen a discretization step that balances memory usage and approximation.

A similar library is called *Geometry*. It contains methods to convert vectors into polar

coordinates and vice versa, to keep values bounded considering a maximum/minimum limit and other mathematical operations.

Finally we have developed *CollisionDetection*: it is a class used to check collisions and find times of contact. It works with discs and lines. During these geometric computations we tried to minimize the usage of the square root; every time it is an indispensable operation we use a faster implementation of the square root written in C++, already in Populus.

6.3 Demo

In this section we want to evaluate our approach presenting all six scenarios. Naturally they are characterized by a set of agents that navigate driven by the social path following. In every scenarios we use a camera that shows the scene from the top since it provides a better visualization of paths and behaviors.

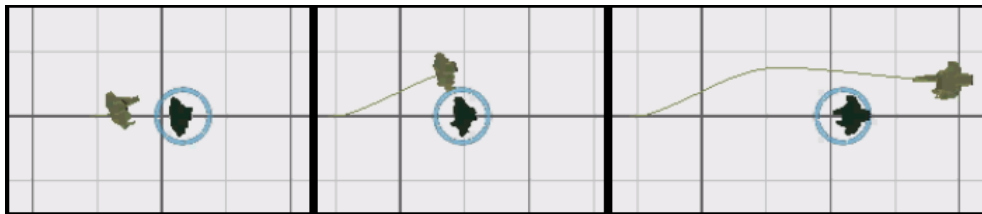


Figure 6.3: First scenario: basic overtaking scenario

The first picture (Figure 6.3) shows three relevant moments of the first demonstration. There are two avatars that have the same final destination but different speeds. Because of agents are very close, the collision is imminent; for this reason the brown agent selects a wide variation of orientation in order to avoid contact and overlap with the obstacle. After overcoming the obstacle, it continues with the normal trajectory to reach the destination, without cutting across the path of the black agent. We can see a really realistic behavior, similar to real pedestrians.

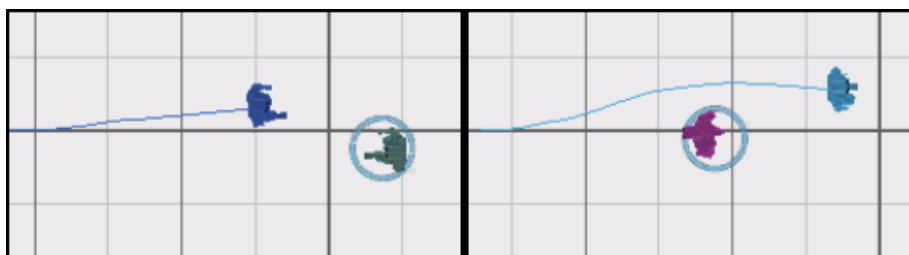


Figure 6.4: Second scenario: robust collision avoidance moving a user controlled avatar

The second scenario is another environment to test the general path following, without including social behavior. In this case we want to show that the algorithm is able to manage collision avoidance in different situations. In particular there is an agent driven by the social path following that moves continuously along a path (alternating starting point and goal) and another one controlled by the user. Every kind of movement made by the user produces a good reaction of the autonomous agent (Figure 6.4).

The third scenario might be useful to evaluate the algorithm in critical situations. A portion of the environment is full of pedestrians: their positions are very similar and having opposite intentions, their movements converge in a specific area. It is a typical crowded simulation. This scene represents the most problematic situation to manage. Four columns of agents is almost the same as three columns, in fact the risk of collision increases extremely when we set three instead of two columns. The same happens considering rows. Moreover it is the worst situation because the arrangement of agents is very rigid (almost improbable in reality): they have the same speed and distance. Although a critical scenario, our implementation provides good results.

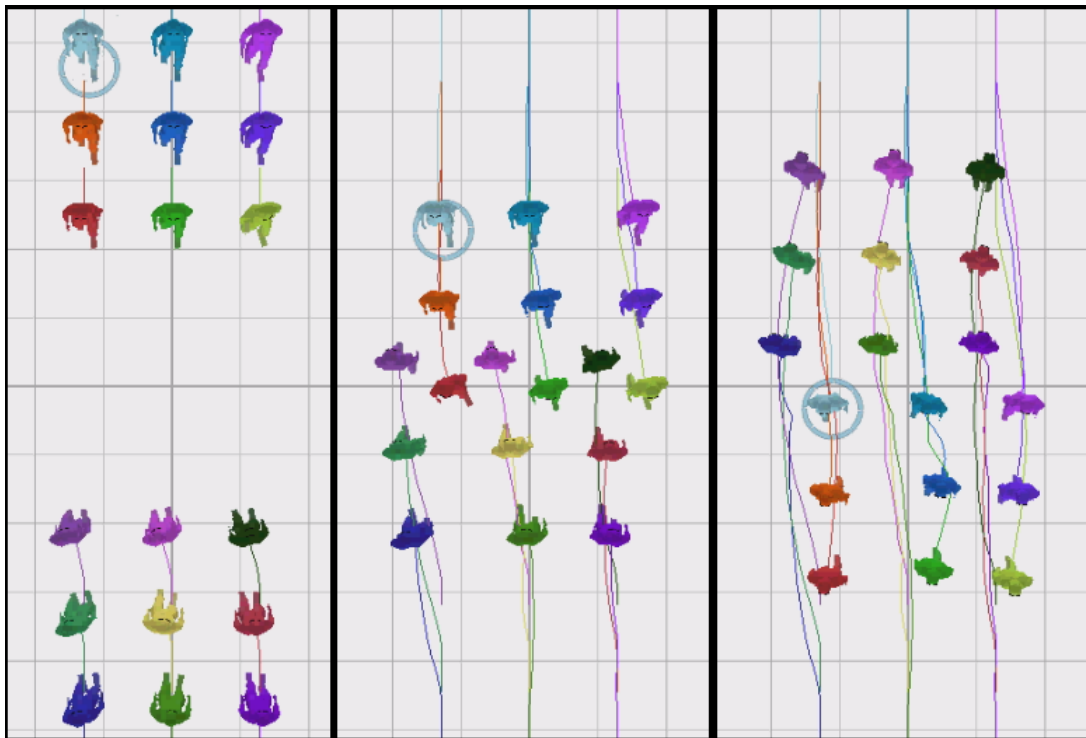


Figure 6.5: Third scenario: crowded environment

The picture (Figure 6.5) shows some screenshots of this demo. Sometimes results are different; it is due to distortions and non-deterministic factors related to the parallel execution of agents as threads. Despite few changes of trajectory, almost all simulations offer credible behaviors. In our demo it is possible to change the number of rows/columns

of agents easily, simulating other situations.

In the fourth demo we want to show an example of the social path following. There are two future conversants and an agent that needs to reach a destination point. The problem is that its path collides with the approaching space of future conversants, as we can see in the first image (Figure 6.6).

Analyzing the simulation, the moving agent starts monitoring behaviors of the other virtual humans and after few iterations it is able to detect a future conversation and to build the "obstacle" to avoid; this assumption is based on mutual gaze and motion. We can see a short approaching space because we compute future positions of agents at the moment of the impact; from this feature we can understand that it is more convenient avoid the obstacle going on the left obtaining realistic territory avoidance. Of course variations of speed produce variations of the length of the approaching space. Moreover its line is very red and expresses a strong necessity of avoidance.

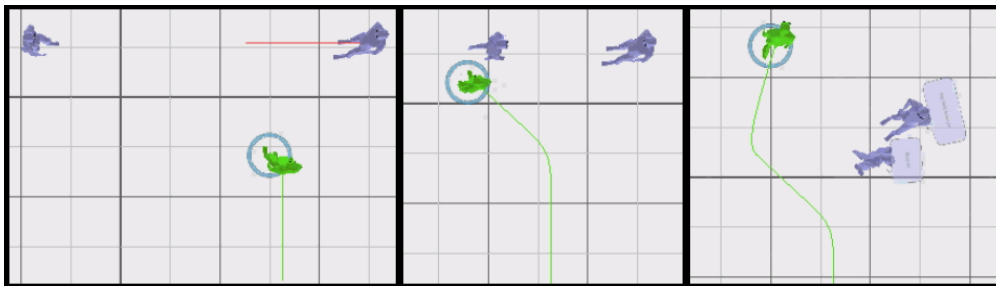


Figure 6.6: Fourth scenario: territory avoidance

The other screenshots show the complete path followed by the green agent. It is just an example; to test the robustness of our algorithm we have simulated other situations adjusting some parameters such as positions of agents and distances. Every test produces interesting results comparing them with the real world.

The fifth demo is completely dedicated to the negotiation of space. We have built a partial environment composed by a narrow corridor, delimited by walls, a conversation and a moving agent (Figure 6.7). It wants to reach the other side of the corridor avoiding obstacles and respecting social rules.

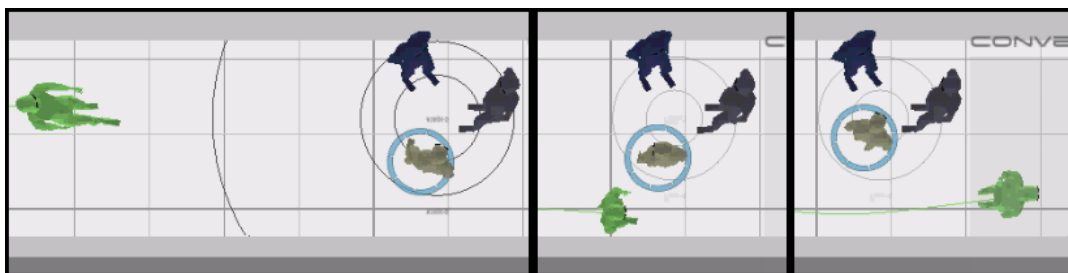


Figure 6.7: Fifth scenario: negotiation of space

Immediately the agent understands that the conversation is a composed obstacle and adjusts its trajectory avoiding it. Perceiving a small passageway, through gaze, the green agent requests more space. The conversant notices the request looking at him because it has perceived him from the gaze of other interlocutors (second screenshot) and moves accordingly (last screenshot). The movement produces a strange shape of the conversation; it is not a problem because after few seconds there will be a natural rearrangement that restores equal roles between participants.

The last scenario is a complex environment that summarizes our project (Figure 6.8).

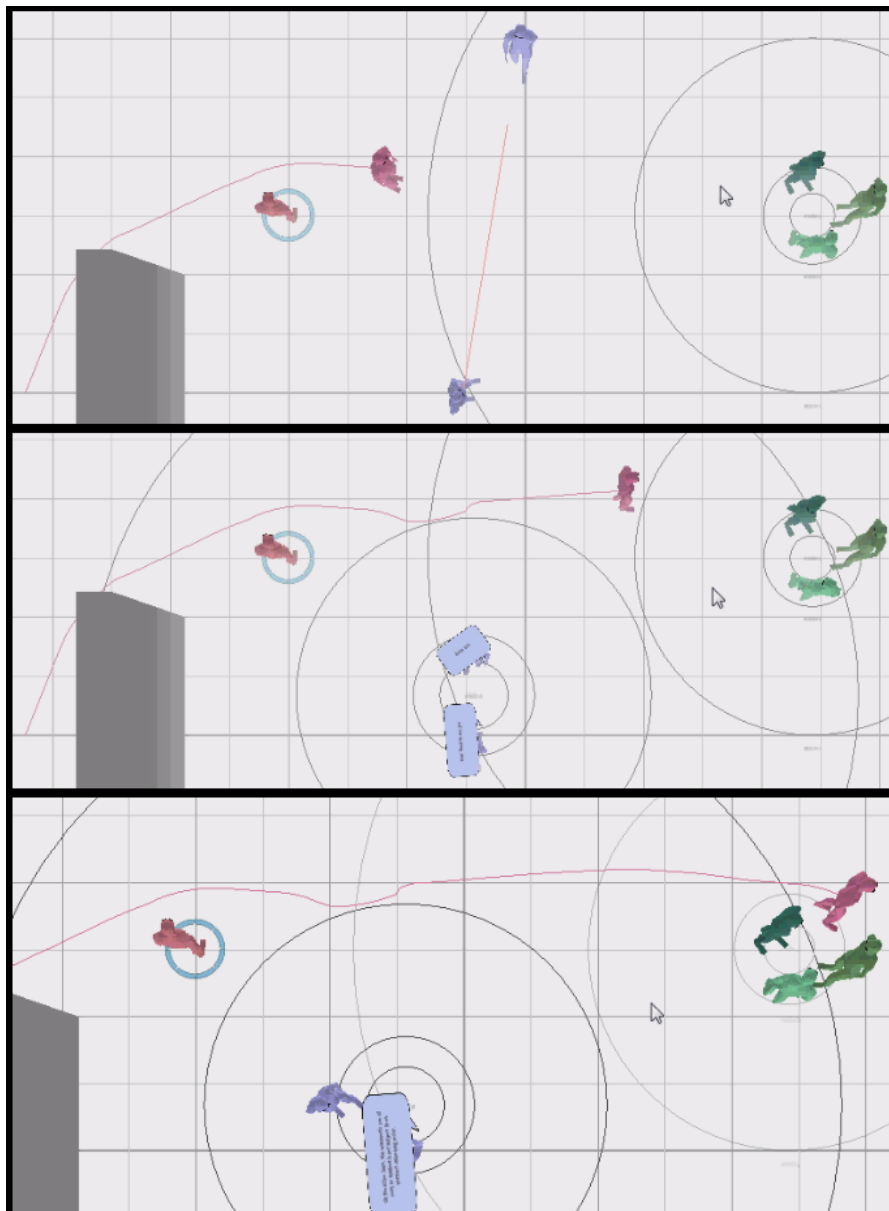


Figure 6.8: Sixth scenario: complex environment

At the beginning the main agent is behind a wall and avoids it applying path finding. The destination is near the conversation, so the agent prefers to dodge around the wall going

on its left, minimizing the length of the complete path. After the wall there is a human obstacle. Despite it should be more convenient dodging him on the right, he goes on its left again, avoiding strong changes of direction. It is typical in human movements. Perceiving future conversants, the main agent adjusts its trajectory avoiding the approaching space. Near the conversation there is enough space to pass but because of the destination point is close to one conversant it produces an invasion of personal space. For this reason, the conversant could protect his territory going closer to other participants interpreting a simple gaze of the moving agent as a request of space, even if there is not a real negotiation.

6.4 GUI

Users can create profiles of agents modifying already existing files or using our interface. In fact we have developed a particular GUI in order to facilitate this operation. Running our demo and pressing zero on the keyboard it is possible to visualize the interface, exactly as the picture below (Figure 6.9).

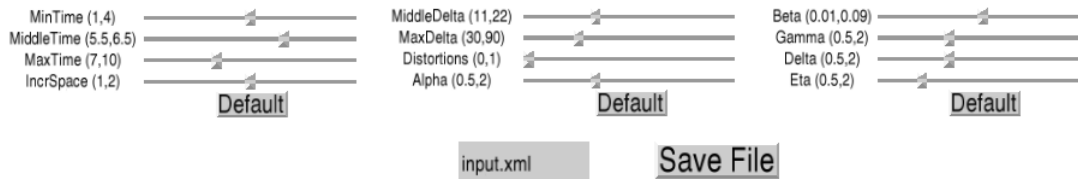


Figure 6.9: GUI to create profiles

In our GUI users can change any attribute of the profile through sliders. Near labels are useful to understand the limited range of values that each slider has. We also created buttons to cancel any modification, setting default values. In the text area it is possible to type the name of the profile and the button saves it into the directory of the demo.

Right now, the graphic interface is used just to create the file and not to load a new profile at run time. In this first prototype the association can be generated giving the name of the file as input of the social path following decorator, through the code.

Chapter 7

Conclusion and Future Work

This thesis provides a point of encounter between social theories and the current state of art about path planning and in particular path following. From several studies about social interactions and human behaviors, we have developed artificial intelligences powered with social attitudes that affect their movements. Exactly as people, each agent navigates around the environment protecting its territory, respecting spaces occupied by others and eventually activating non-verbal negotiation, due to particular exigences of space.

After studying the state of the art of path following, we adapted it to the context of social environment. For this reason, our project can be useful reproducing locomotion of pedestrians or generic worlds populated by virtual humans. Furthermore, deactivating social behaviors, our implementation can be useful also in several other contexts like robotics and simulations.

Social environments require a high level of realism, keeping users engaged and interested. It represents the main purpose of our research, always kept in mind. Our approach is also flexible and extensible with possible improvements or new behaviors. It supports loading/saving several profiles of virtual people modeling, many different personalities and reactions, evident during their movement. All these features are extremely important to create a real virtual world.

Moreover, observing movements of people, comparing data and results with our current scenarios could drive us towards corrections and improvements. Maybe it is a good way also to understand possible omissions or inaccuracies in the literature about human territories and sociology, especially for theories that cannot be formally proven. This could mean a reduction of the gap between theories and the real world provided by the technology and virtual environments.

7.1 Future Work

Despite realism, flexibility and various personalizations make our project interesting, of course there are several improvements that we would like to include in the future.

First of all, CADIA Populus can facilitate our future work by providing more extensive perception of objects and interactions; in this way the collision avoidance can run with any obstacle (not only agents and walls) and we can improve social behaviors regarding dynamic conversations.

Usually, avoiding private conversations politely, people dip head and eyes; it would be great to include a similar animation being faithful to reality.

We can think about improving the prediction of intentions considering also voice, gestures and all other typical non-verbal behaviors that are activated before a conversation. Another possibility is to modify the current probabilistic model considering other percentage values; this implies a formal evaluation of virtual simulations in comparison to videos of real social interactions.

Potential initial collisions could break the realism. Right now each agent is able to escape from already existing collisions with others; since overlaps of private spaces are not visible, all works perfectly. On the other hand initial inter-penetrations of bodies should be absolutely avoided through a controller of initial positions.

Developing something similar to a “Social A* Algorithm” could be really fascinating, especially using an heuristic function based on social costs. In this way we can achieve a perfect feeling between finding and following.

A really interesting improvement consists of dynamic personal spaces, especially during interactions. It means variable size of personal spaces based on relationships with other participants and level of involvement. It might be difficult right now but it would be fascinating to know how the social path following accepts this arrangement.

Other possible improvements are new social behaviors and a more efficient space model to have better performance, achieving the same level of the realism.

Bibliography

- [1] Champandard Alex J. <http://aigamedev.com/insider/presentations/recast-teaser/>, 2009.
- [2] Champandard Alex J. <http://aigamedev.com/open/reviews/hpa-future-multicore/>, 2010.
- [3] M. Argyle, R. Ingham, F. Alkema, and M. McCallin. The different functions of gaze. *Semiotica*, (Journal Article), 1973.
- [4] Geoffrey W. Beattie. *Sequential Temporal Patterns of Speech and Gaze in Dialogue*, pages 298–320. Approaches to Semiotics. Mouton Publishers, New York, 1981.
- [5] Simon J. Bennett, Robin Baures, Heiko Hecht, and Nicolas Benguigui. *Eye movements influence estimation of time-to-contact in prediction motion*, volume 206. Springer, Heidelberg, ALLEMAGNE, 2010.
- [6] Adi Botea, Martin Müller, and Jonathan Schaeffer. Near optimal hierarchical path-finding. *Journal of Game Development*, 1:7–28, 2004.
- [7] Angelo Cafaro. Simulating the idle gaze behavior of pedestrians in motion. 2009.
- [8] Mark S. Cary. The role of gaze in the initiation of conversation. *Social Psychology*, 41(3):269–271, 1978.
- [9] Alex R. Colburn, Michael F. Cohen, and Steven M. Drucker. The role of eye gaze in avatar mediated conversational interfaces. Technical report, Microsoft Research, 2000.
- [10] Christer Ericson. *Real-time collision detection*. Morgan Kaufmann series in interactive 3D technology. Elsevier, Amsterdam ; Boston, 2005.
- [11] Rachel Gockley, Jodi Forlizzi, and Reid Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 17–24, 1228720. ACM.

- [12] Lewis Hedwig. *Body Language: A Guide for Professionals*. Sage Publications Pvt. Ltd, New Delhi, 2000.
- [13] Ioannis Karamouzas and Mark Overmars. A velocity-based approach for simulating human collision avoidance. In *International Conference on Intelligent Virtual Agents (IVA 2010)*, volume 6356, pages 180–186. Springer Berlin / Heidelberg.
- [14] Sotaro Kita. *Interplay of gaze, hand, torso orientation and language in pointing*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2001.
- [15] Chris L. Kleinke. Gaze and eye contact: A research review. *Psychological Bulletin*, 100(1):78–100, 1986.
- [16] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–118, 1272706. Eurographics Association.
- [17] Patrick Lester. <http://www.policyalmanac.org/games/astartutorial.htm>, 2005.
- [18] Peter E. Marsh. *Eye to eye : how people interact*. Salem House Publishers, Topsfield, Mass., 1988. 87035581 edited by Peter Marsh. ill. (some col.). Bibliography: p. 249-251. Includes index.
- [19] Ronald A. Metoyer and Jessica K. Hodgins. Reactive pedestrian path following from examples. In *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, page 149, 791856. IEEE Computer Society.
- [20] Alex Nash, Kenny Daniel, Sven Koenig, and Ariel Feiner. Theta*: any-angle path planning on grids. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, pages 1177–1183, 1619835. AAAI Press.
- [21] Claudio Pedica and Hannes Högni Vilhjálmsson. Spontaneous avatar behavior for human territoriality. In *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pages 344–357, 1612602. Springer-Verlag.
- [22] Steve Rabin. *AI game programming wisdom*. Charles River Media, Hingham, Mass., 2002. 2001008585 edited by Steve Rabin. ill. (some col.). + 1 CD-ROM (4 3/4 in.) System requirements for accompanying CD-ROM: Windows 95 or Windows NT4 or later. Includes bibliographical references and index.
- [23] Craig Reynolds. Big fast crowds on ps3. In *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pages 113–121, 1183333. ACM.

- [24] Craig Reynolds. Interaction with groups of autonomous characters. In *Game Developers Conference*, pages 449–460. CMP Game Media Group.
- [25] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 25–34, 37406. ACM.
- [26] Craig W. Reynolds. Steering behaviors for autonomous characters. In *Proc. of the Game Developers Conference*, volume San Jose, CA, pages 763–782. Miller Freeman Game Group.
- [27] Stuart J. Russell and Peter Norvig. *Artificial intelligence : a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall/Pearson Education, Upper Saddle River, N.J., 2nd edition, 2003. 2003269366 (Stuart Jonathan) Stuart J. Russell and Peter Norvig ; contributing writers, John F. Canny ... [et al.]. ill. Includes bibliographical references (p. 987-1043) and index.
- [28] Albert E. Schefflen and Norman Ashcraft. *Human territories : how we behave in space-time*. A Spectrum book S-375. Prentice-Hall, Englewood Cliffs, N.J., 1976. 75020472 Albert E. Schefflen, with Norman Ashcraft. ill. Bibliography: p. 201-202.
- [29] Wei Shao and Demetri Terzopoulos. Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 19–28, 1073371. ACM.
- [30] Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE.



School of Computer Science
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is
ISSN 1670-8539