EUROGRAPHICS 2007 / D. Cohen-Or and P. Slavík
(Guest Editors)

Volume 26 (2007), Number 3

Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach

Sébastien Paris Julien Pettré Stéphane Donikian

IRISA, Campus de Beaulieu, F-35042 Rennes, FRANCE
{sebastien-guillaume.paris, julien.pettre, donikian}@irisa.fr

Abstract

This paper addresses the problem of virtual pedestrian autonomous navigation for crowd simulation. It describes a method for solving interactions between pedestrians and avoiding inter-collisions. Our approach is agent-based and predictive: each agent perceives surrounding agents and extrapolates their trajectory in order to react to potential collisions. We aim at obtaining realistic results, thus the proposed model is calibrated from experimental motion capture data. Our method is shown to be valid and solves major drawbacks compared to previous approaches such as oscillations due to a lack of anticipation. We first describe the mathematical representation used in our model, we then detail its implementation, and finally, its calibration and validation from real data.

1. Introduction

This paper addresses the problem of virtual pedestrian autonomous navigation for crowd simulation. One crucial aspect of this problem is to solve interactions between pedestrians during locomotion, which means avoiding inter-collisions. Simulating interactions between pedestrians is a difficult problem because its complexity grows rapidly with respect to population density. Also, obtaining realistic results is challenging: humans are used to observe navigating pedestrians in the real life and immediately detect artifacts in simulations. We present a reactive navigation technique for application in the domains of architecture, security, space ergonomy, and also the entertainment industry. We expect natural crowd motion emerging from a realistic microscopic pedestrian simulation.

Our solution for solving interactions between pedestrians is predictive and agent-based. Inputs are the definition of an environment, the current state and the destination of each pedestrian - destination is a desired direction derived from a navigation plan. The method first checks on future interactions between pedestrians: the evolution of pedestrians' position is predicted from an extrapolation of their current state. When needed, a long term avoidance motion is computed by taking into account these predictions. Our resulting microscopic pedestrian simulation model is calibrated and validated using motion capture data. Data are acquired according to two successive protocols. First, we measure interactions between two participants and use the resulting data to calibrate our model. In a second stage, we push the number of participants to the limits of our motion capture system abilities and let them navigate among obstacles, allowing us to compare the measured data with our simulation results.

Our first contribution is to solve major drawbacks in previous microscopic approaches, such as oscillations and jams. We believe these drawbacks were due, firstly, to the lack of anticipation, and secondly, to the simplicity of the reaction computation technique from the observed situation. Our second contribution is to propose a motion capture-based calibration of the model and validation of our results. Validation is generally done using hand-processed video sequences. Motion capture data are more accurate than those derived from video sequences: this allows us to decompose precisely in time and space how humans react to potential collision with others. Particularly, we could extract a criterion to detect the need for a reaction and compute adequate corrections to the trajectory.

Section 2 states our contributions with comparison to previous approaches. Section 3 describes our method to solve interactions between pedestrians from a technical point of view: firstly, how it is integrated into a crowd simulator, sec-

ondly, how pedestrians perceive their surrounding environment, and thirdly, how a reactive motion is computed. Section 4 describes the protocols and the results of the experiments used to calibrate the model parameters and to validate the approach. We then present some simulation results in usual architectural configurations. Finally, we conclude and provide perspectives to our work.

2. Related Work

One of the most important skills of a human being is her ability to navigate inside her environment. Even if this navigation task is one of the most basic behavior in real life, it is not yet solved correctly in a virtual world. First to allow people to navigate, they should be able to perceive their environment but not only in a geometric way. Studies in psychology and urbanism have shown that visibility and topology are also important in the navigation task. A structured and informed environment has to be used for path planning and reactive navigation of virtual humans in real time. The simplest task, for a pedestrian walking in a street, consists in minimizing possible interactions, which mean avoiding static and dynamic obstacles. Goffman [Gof71] describes techniques used by pedestrians to avoid bumping into each other. The social link between strangers is characterized by silence and indifference and to perform that, different behaviors are used. The first technique called externalization concerns the way that people are constantly making others aware of their intentions in order to minimize the interaction. Pedestrians selectively gather externalized information from other people by a second technique called scanning. The third technique, called the minimization of adjustment, expresses that people adjust their trajectory several meters before the conflict to make it perceptible early by others with the objective to reduce interaction and avoid coordination. Goffman introduces the notion of an oval security region whose front distance corresponds to an anticipation area depending on the pedestrian speed, while the width is the accepted gap to pass beside a person or an obstacle or to follow a wall. He also defines the law of minimal change meaning that a pedestrian will try in its journey to reduce the amount and the amplitude of turns. These studies illustrate the importance of prediction and anticipation in the navigation task.

It is known that in crowd motions, pedestrian flows walking in opposite direction generate their splitting to create dynamically some bands of pedestrians walking in the same direction. When the density of pedestrians becomes very high, it is possible to approximate the overall behavior of the crowd by using the laws of fluid evolution [TCP06]. In panic situations, pedestrians wish to move more quickly than usual and, forgetting all social rules, accept to be in a physical contact with their neighbors. Due to this physical interaction situation, they are developing a mimetic behavior consisting in reproducing the behavior of preceding characters in the flow. For Yamori [Yam98], this is obligatory followed by the notion of regulation, learned as a normative element by people living together inside the same macro structure or institution. However, as emphasized by Musse et al. [MT01], some small groups can be the motor and modify the behavior of biggest units such as a crowd, playing the role of the core group. Boles [Bol81] has observed the existence of a band structure inside a crowd of pedestrians moving on a sidewalk and explain this by its optimal configuration to regulate opposite flows. One of the most crucial problems to be solved, as pointed by Yamori, concerns the relation between microscopic and macroscopic structures and behaviors inside the crowd. The goal is to explain how an individual entity is constrained by the institution and on the other hand how the community impacts on the individual behavior along time. Yamori focuses his research in the formation of macroscopic band structures and postulate that such kind of macroscopic structure requires a critical density of population to emerge from the set of individual behaviors.

Macroscopic simulation has been historically the first approach to be studied to simulate the pedestrian displacement, due to its low calculation cost. In this approach the pedestrian is not treated on its own but as a component of a more macroscopic element [Hen71, PM78]. These macroscopic models are often used for animation purpose, like by Sung et al. [SGC04], to provide a globally convincing crowd motion. Another approach called microscopic simulation consists in handling the individual navigation of all moving entities. In that case, a system allowing dynamic collision avoidance is necessary to achieve consistency and realism. Several approaches can be distinguished such as particle and flocking systems. Particle systems are based on physical laws describing attractive and repulsive forces that can be associated to obstacles and moving entities. Forces applied to an entity are summed to calculate its new motion direction and speed [HFV00, BMdOB03, LKF05]. This model assimilates the displacement of an entity in the case of a high density to the motion of a particle inside a restricted area. I. Peschl [Pes71] justifies the use of this model in the case of an emergency situation with a high density of population. Particle based models allow the generation of a macroscopically plausible behavior in case of a high density, but they do not take into account anticipation, perception, or social rules. Moreover, close inspection of individual trajectories show some oscillations and unrealistic behaviors such as backward motion of the last people repulsed by the preceding one in a queue and many change of orientation along the path due to the interaction with other moving entities and static obstacles. Another drawback of this approach is its requirement of a small time-step for convergence purpose. Flocks are rule-based systems defining the behavior of an entity according to the behavior of the nearest entities [Rey00, BLA02]. It is well adapted for the collective motion of a group of animals following a leader but less for the variety of behaviors that can be observed in a sparsely populated crowd of humans.

Loscos et al. [LMM03] use a fine regular grid to handle reactive navigation and to store information about pedestrian movements enabling the emergence of flows of pedestrians. In the same way, Shao et al. [ST05] use a quadtree map for the path planning and a fine regular grid for obstacle avoidance.

Lakoba et al. [LKF05] argue that two points are necessary to improve the existing models: add decision-making capabilities and compare simulation results against measured data on pedestrian dynamics. Instead of classical models of crowd simulation based on fluid dynamics or particle systems which are only valid in very dense crowds, S. Goldenstein et al. [GKM*01] have proposed a multi-layer approach to model the behavior of crowd participants. We are also working on a multilevel model of each human allowing us to simultaneously take into account attraction/repulsion mechanisms such as in particle systems, dynamic computation of the neighborhood for sparse crowds, the management of social rules, path planning and activity planning. In this paper, we are focusing on the reactive navigation model and on the use of experimental data to validate the approach and calibrate the model.

3. Prediction and Resolution of Interactions

3.1. Principle

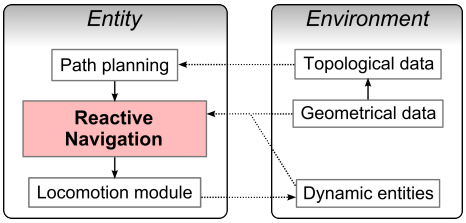

Figure 1: *The simulation architecture.*

The problem of reactive pedestrian navigation is part of the crowd simulation problem, and our method is included in a global architecture as shown in Figure 1. More details on the environment management and on dedicated path planning techniques we developed can be found in the literature [PDB05,PDB06]. The reactive navigation role is to steer entities in a realistic manner with respect to two possibly conflicting inputs: the goal of the considered pedestrian resulting from the path planning stage, and the current state of the environment, especially the presence of other pedestrians. The output we search for are updated speed and orientation allowing the pedestrian to avoid any static or moving obstacle while satisfying constraints of realism.

Our approach to this problem is a predictive one. For each entity, at desired rates, we search for a solution-move satisfying constraints and guaranteed to remain valid for a desired time window (at least the period at which reactive navigation is invoked). The key-idea is to model the environment

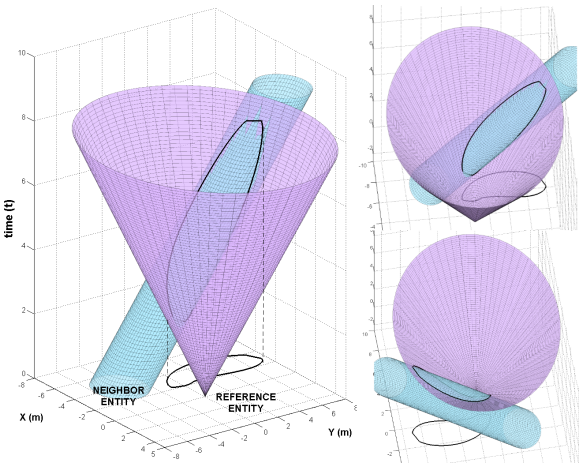

Figure 2: *Modeling the interaction between a reference entity and a neighbor entity in the (x,y,t)-space. The predicted trajectory for the neighbor entity is the blue cylinder, whereas the reachable space for the reference entity is the violet cone. Their intersection delimited in black is a future collision area (also projected in the plane).*

as shown in Figure 2 in the (x,y,t)-space, with (x,y) the horizontal plane and t the simulation time. We explore the reachable space of the *reference entity* in any direction and for a range of speed values, and search for possible collisions with *neighboring entities*. Figure 2 illustrates such an exploration for a given reference entity speed value. As any move direction is envisioned, the reachable space is then represented in the (x,y,t)-space as a cone whose opening angle depends on the considered speed value. *Neighboring entities* are then taken into account: they are represented as circles – whose radii are the sum of both the reference and the considered entity radii – moving along a predictive trajectory, computed from the current position, speed and orientation of the neighboring entity. Thus, the neighboring entity is modeled in the (x,y,t)-space as an elliptic cylinder. Consequently, the intersection of the cone and the cylinder delimits a *collision area* that the reference entity shall avoid. The difficulty of the problem is brought to mind looking at Figure 3, where different shapes of the collision area are displayed according to various solution speed values. Additionally, the figure does not represent the possible presence of several neighboring entities and of static obstacles which obviously increase the problem complexity drastically. As a result, we choose to base our solution on a discrete-time expression of the same modeling in order to avoid the problem complexity.

Three main steps, detailed in the next sections, allow us to compute the best speed and orientation for the reference entity:

1. Neighboring dynamic entities are first taken into account. From this, we deduce some sets of speed and orientation

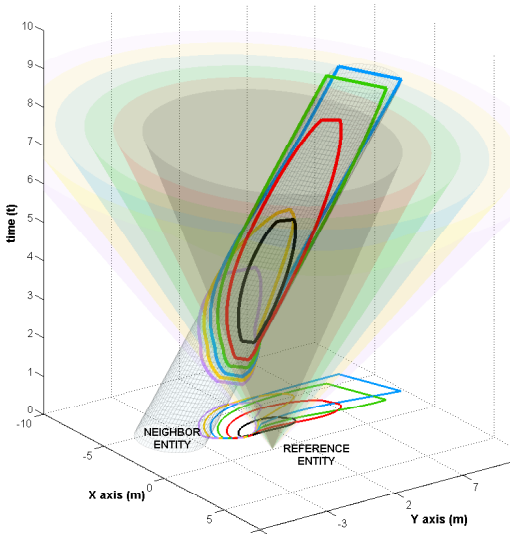

Figure 3: *We explore the reachable space for the reference entity for a given range of speed. The opening angle of the cone changes accordingly in the (x,y,t)-space, as well as the intersection with the predicted trajectory of the neighbor entity (each color corresponds to a different cone opening angle, and thus to a different reference speed value).*

ranges that allow collision free motion for a future time window.
2. In the same manner, static obstacles are considered. We deduce new valid sets of speed and orientation ranges.
3. Previous valid solution ranges are merged, scored and compared. The best one is returned as the solution.

3.2. Dynamic entities

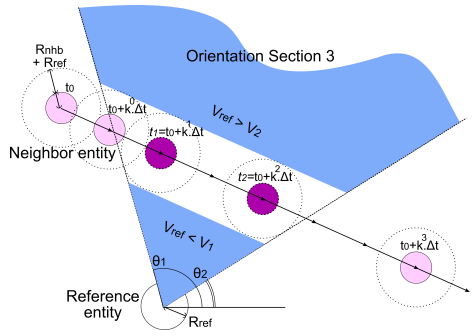

Figure 4: *Example of reachable space sectioning for a given time-interval. The resulting orientation section is characterized by several parameters: $t_1, t_2, \theta_1, \theta_2, V_1, V_2$.*

The objective of this first step is to compute a set of valid speed and orientation ranges for a given reference entity E_{ref}, with respect to the presence of neighbor moving entities. We describe our method for a single neighbor entity E_{nhb}, whilst the case of several ones is detailed Section 3.4.

Time discretization. We consider the environment state at $t = t_0$ and propose to search the E_{ref} reachable area for potential collision as explained previously. For that, we consider successively adjacent time-intervals having different durations: $[0, k^0\Delta t]$, $[k^0\Delta t, k^1\Delta t]$, $[k^1\Delta t, k^2\Delta t]$, $[k^2\Delta t, k^3\Delta t]$, etc. The $\Delta t > 0$ parameter defines the precision of the discretization, smaller being the best, and should correspond to the time needed by the entity to make one move. The $k > 1$ parameter is used to make the discretization non-uniform over the anticipated time, greater being the best, allowing the anticipation to be more precise in the near future than in the distant one. We use $\Delta t = 1$ and $k = 2$ in our model.

Reachable space sectioning. For each time-interval, we predict the E_{nhb} trajectory as a linear one and deduce the E_{ref} orientation range (*orientation section*) potentially leading to a collision with E_{nhb} as illustrated in Figure 4. We finally get as many sections as the number of time-intervals, each representing an orientation range for E_{ref}. The time-interval $[t_1\ t_2]$ used to compute each section is stored.

Critical speeds computation. For each orientation section, we compute the critical speeds V_1 and V_2 defined as follows: V_1 is the *maximal speed* allowed to avoid a collision by passing behind E_{nhb}; V_2 is the required *minimal speed* to avoid a collision by passing before (in front of) E_{nhb}. V_1 and V_2 have analytical expression, as solution to the following equation:

$$V_1 = \min_{t=t_1}^{t_2} \left((\left\| \overrightarrow{P_r P_n(t)} \right\| - R)/t \right)$$
$$V_2 = \max_{t=t_1}^{t_2} \left((\left\| \overrightarrow{P_r P_n(t)} \right\| + R)/t \right)$$

with $P_n(t) = P_n + \vec{v_n} t$ and where $\vec{v_n}$ is the E_{nhb} speed vector, P_r and P_n are respectively the positions of E_{ref} and E_{nhb} at $t = t_0$, and finally R is the sum of the bounding circles radii of the considered entities, eventually increased by a security factor to avoid strict contact cases.

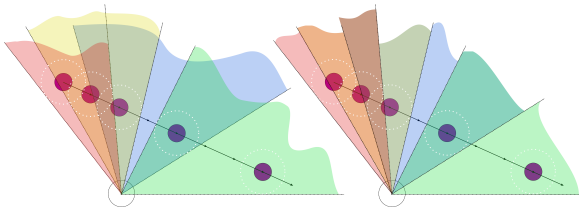

Figure 5: *The sectioning results in overlapping orientation ranges (left image). The merge of overlapping sections is easily done by subdividing orientation ranges in order to get adjacent sections (right image). The new characteristics of each subdivision are directly deduced.*

Merging several sections. By iterating previous computation for each time-interval, we get a set of orientation sections with their own characteristics (V_1, V_2, time-interval $[t_1\ t_2]$). A problem is to merge overlapping sections. Sections are finally subdivided in order to get a set of adjacent sections. Each new subsection characteristics are computed as follow: $V_{1new} = min(V_{1i}, V_{1j}), V_{2new} = max(V_{2i}, V_{2j}), t_{1new} = min(t_{1i}, t_{1j})$ and $t_{2new} = min(t_{2i}, t_{2j})$ with i and j indexing the two merged orientation sections. Note that if three or more sections overlap, this process can be reiterated successively considering pairs of sections until all are merged.

3.3. Static entities

The second step of the reactive navigation module consists of considering static obstacles. We handle static obstacles in approximately the same way than dynamic entities, but the problem is obviously simpler. In our environment database, obstacles borders are modeled as line segments. Let us consider the case of a single line segment in the vicinity of E_{ref}. Our first objective is to subdivide S_{obst} as shown in Figure 6. For that, we first compute P_0 the nearest point from E_{ref} belonging to S_{obst}. We define points P_1 and P_1' if existing so that the length $P_0P_1 = P_0P_1' = v_{ref}\Delta t$ where v_{ref} is E_{ref} speed, and Δt defined in the previous Section. Then, we define the point P_2 so that $P_0P_2 = k^1 v_{ref}\Delta t$, P_3 so that $P_0P_3 = k^2 v_{ref}\Delta t$, and finally P_4 so that $P_0P_4 = k^3 v_{ref}\Delta t$. This set of points is arbitrary, however, it allows us to evaluate the constrained speeds toward the obstacle with a more accurate precision near to E_{ref}.

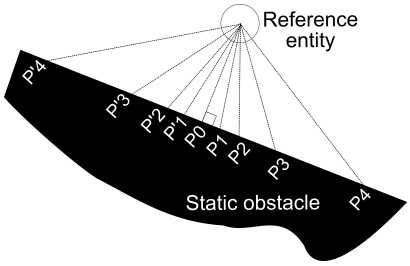

Figure 6: *Sections computation for static obstacles*

A set of adjacent orientation sections are computed as shown in Figure 6. We then compute the characteristics of each section in a same way than done previously with dynamic entities. However, V_1 is computed differently:

$$V_{1i} = \left\| \overrightarrow{P_iE_{ref}} \right\| / t_{2i}$$

where i indexes the considered orientation section. V_1 is thus the maximal speed at which E_{ref} can walk within the considered time-interval (depending on the considered section) without colliding the obstacle, analogically to the previous dynamic entity case. V_2 is here meaningless, and we set $V_2 = +\infty$.

3.4. Solving interactions

The third and final step of our reactive module consists of extracting a solution-move for the reference entity. The extraction is done in three successive steps. First, each orientation section previously computed is weighted using a cost function depending on sections characteristics. Second, superposed sections (related to different neighboring entities or obstacles) are merged (see Section 3.2) while accumulating costs. Third, the best section is used to compute the new speed and orientation.

Orientation section cost. The reference entity has to choose the best next speed and orientation according to the environment state and its goal. The function cost reflects the best choice among several criteria:

- Valid speed ranges (according to V_1 and V_2 of each section) must be close to the entity desired speed V_{des} and in its range of achievable speeds $[0; V_{max}]$.
- Orientation section limits $[\theta_1; \theta_2]$ must be as close as possible to the desired orientation θ_{des}
- Required accelerations to reach the new speed and orientations must be as limited as possible (limiting strictly them is not desirable because real humans are capable of important accelerations).
- The closer the section time-interval is in the future, the more confident we are in its cost.

The cost associated to speed variations and distance to desired speed is computed as follows:

$$C_{decel} = \begin{cases} 0 & if\ V_{des} \leq V_1 \\ 1 - \frac{V_1}{V_{des}} \end{cases} \qquad C_{accel} = \begin{cases} 0 & if\ V_{des} \geq V_2 \\ \frac{V_2 - V_{des}}{V_{max} - V_{des}} \end{cases}$$

$$C_{speed} = \alpha.min(C_{accel}, C_{decel})$$

where $\alpha \in]0;1[$ allows us to set a trade-off between speed changes and orientation changes. The cost associated to orientation changes (deviation) is computed as follows:

$$C_{dev} = (1-\alpha)\frac{1 - cos(\theta)}{2}$$

where θ is the minimum difference between the desired orientation and the orientation section limit angles. Note that $0 \leq C_{speed} + C_{dev} \leq 1$. According to t_1 the lowest bound to the time-interval of the concerned section, we finally compute a prediction confidence cost:

$$C_{pred} = 1 - \frac{t_1}{T + \beta} \qquad \text{with } 0 \leq C_{pred} \leq 1$$

where T is the maximal considered time for the prediction, and $\beta \in [0; +\infty]$ is a user-fixed parameter allowing us to get more or less confidence in predictions (this will change the pedestrian adaptation-time before a potential collision). The total cost of a given section is then:

$$C_{total} = (C_{speed} + C_{dev}).C_{pred}.C_{add}$$

where $0 \leq C_{add} \leq +\infty$ is an additive cost that can be introduced to take into account external factors, e.g., walking

close to another entity, preferential deviation to the left or the right, etc. Neutral value is $C_{add} = 1$.

Merging costs. Note, at this point, we have as many sectionings (whole sets of orientation sections) as the number of entities and static obstacles in the vicinity of E_{ref}. For each section of each set, we now have a cost. In exactly the same manner than presented in Section 3.2, we merge all the weighted sections. The cost of each subsection thus created is the sum of all the sections that were superposed and split to create it.

The fittest subsection has the lowest cost. In the corresponding range of valid speeds and orientations, we compute the closest to the ones desired by the reference entity, which is the final output to our reactive navigation module.

3.5. Discussion

Visibility of neighbor entities. As explained previously, our model predicts neighbor entities trajectories in order to decide on the best reaction to avoid them when necessary. We demonstrate in the next Section that real humans act in the same way, however, it is obvious they only do so for the humans they could visually perceive. In order to get a realistic reactive navigation, a perception field must be simulated. Our model distinguish two cases. When a neighbor entity is not seen because it is occluded by an obstacle, it is filtered out of the selection. As a result, an occluded neighbor entity has strictly no influence in the result. A limitation is the case were two pedestrians invisible one to the other converge toward the same place (e.g., at a street corner): they stop abruptly when finally perceiving one another, whereas a real human would anticipate this possibility and walks more carefully. When a neighbor entity is not seen because it is not in the field of view of the reference entity, we introduce it in the model, but having a null speed. Indeed, we consider that real humans feel someone is behind, but are unable to predict any trajectory. This also avoids backward motions provoking a collision with entities behind.

Connection with a locomotion animation module. As seen in Figure 1, output of our reactive navigation module is connected to a locomotion module. Our method may lead to important speed or orientation changes from which a realistic animation must be computed. In order to get as realistic animations as possible, we synchronize the animation module and the reactive navigation module, so that changes occur at the feet-land instants (left or right). Also, the locomotion module smooths variations itself.

4. Model Calibration and Validation

The previous Section described the technical basis of our model: resulting trajectories mainly depend on the parameters used in the cost functions (α, β). In order to get realistic behaviors, a calibration of the model is required. In previous works, such a calibration is achieved by analyzing video captures of real crowd motions. However, automatic analysis techniques are not always applicable due to lighting conditions, and analysis by hand requires great effort. Moreover, pedestrians goals are generally unknown in video sequences, which may prevent strict comparisons between real data and simulations. For these reasons, we prefer to use a motion capture system to collect our reference data, with protocols defining the goals of each participant.

In a first experiment, we measure the interaction phenomenon in the following situation: two pedestrians achieve navigation tasks in an empty environment, we force them to have more or less interacting trajectories and observe adaptations to avoid contact. Results allow us to demonstrate the need for prediction in a realistic reactive navigation model and to calibrate some crucial factors. Secondly, we attempt to reproduce some typical crowd navigation situations at a microscopic level, such as corridor following, gate crossing, X-crossing, etc. in order to validate the model behavior in more complex scenes. For that, we captured as many participants as possible executing navigation tasks in an environment made of obstacles.

4.1. A protocol for model calibration: interactions between two pedestrians

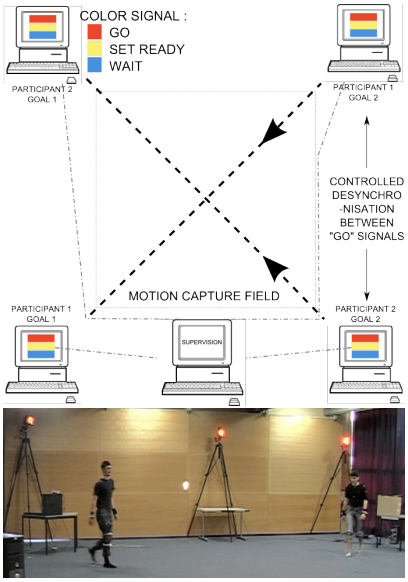

Figure 7: *Four computers displaying signals to participants controlling their start-time and goal. A motion capture system retrieves resulting interactions.*

The first experiment protocol allows us to control the interaction of two participants walking in an obstacle-free area, as illustrated in Figure 7. By interaction, we mean the

required trajectory adaptations made by each participant in order to avoid a collision. The objective of the experiment is to qualify the avoidance strategies developed by the participants given the conditions imposed by the protocol, and to quantify the trajectory corrections made in terms of velocity and orientation changes. For that, we place four computers at the corners of a square area visible for a motion capture system. Each participant must go from a given computer to the diagonally opposite one. We get a temporal control on the experiment by transmitting a start signal to participants using computer displays. All computers are synchronized in order to precisely control some delays between the start time of each participant, and to provoke more or less important interactions between them. We deduce the conditions for an interaction to occur or not. Participants always see each other but only perceive their own start signal. Finally, participants are equipped with 34 markers to get full body motion capture data. A total of 145 interactions were captured, 6 participants where involved, we placed computers in order to form $\pi/2$ or $\pi/3$ angles between trajectories.

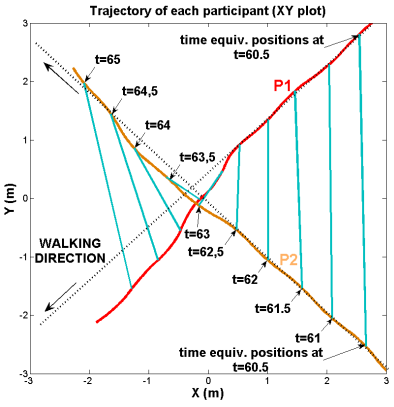

Figure 8: *Horizontal trajectory of 2 interacting participants.*

We detail our analysis method over a specific case whose results are shown in Figures 8 and 9. Here, as it can be seen in Figure 8 representing the horizontal trajectory of participants $P1$ and $P2$, a strong interaction occurred between experiment times 60s to 66s: trajectories are conspicuously deformed in order to avoid collision. We joined the respective positions of $P1$ and $P2$ at equivalent times (each half a second) in order to provide a temporal indication of events. A first look at the results leads one to think that participants reacted late (t=62.5s), just before collision: $P2$ passes before $P1$, $P1$ decelerates and turns to the left while $P2$ turns a little to the left to facilitate the passage.

But motion capture data allow a more precise analysis and the real intentions of participants appear clearly in Figure 9: top and bottom plots are the speed and orientation of each participant (respectively red and blue plots). In dashed horizontal lines, we represent some mean reference values for

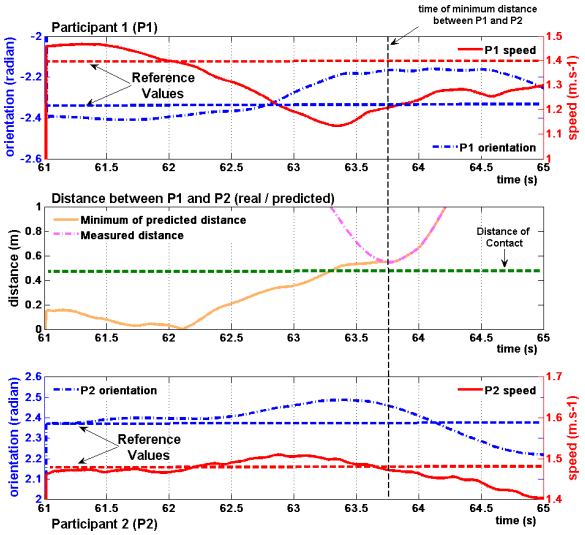

Figure 9: *Orientation and speed variations of participants $P1$ (top) and $P2$ (bottom). Comparing the predicted and measured distances between participants allows to detect the time of adaptation to avoid a collision (center).*

speed and orientation, measured during experiments where participants execute identical navigation tasks alone. We observe that two successive corrections finally compose this interaction:

- $t \in [61;62s]$: $P1$ has a higher speed and deviates to the right compared to mean reference values. This reveals her first intention to pass before $P2$, who has a normal behavior. But $P2$ naturally walks faster (looking at the reference mean values) than $P1$ and this first strategy fails.
- $t \in [62;62.5s]$: $P1$ decelerates.
- $t \in [62.5;64s]$: A combined reaction is now clearly visible: $P1$ increases its deceleration and deviates to the left, which will allow her to pass behind $P2$, who facilitates the success of this new strategy by accelerating and deviating to the left.
- $t > 64s$: After the time where distance between $P1$ and $P2$ is minimal, participants achieve their goal and no particular interaction is observable.

Previous analysis showed that corrections may appear early in the experiments ($P1$ for $t = 61$ to $62s$), and differences between reference and measured values may be minor and hardly detectable. We introduce a criterion to both qualify automatically situations of interactions in the different experiments and answer a crucial question: are corrections made by participants pertinent (i.e., does collision occurs if no adaptation is made)? This will allow us to conclude on the accuracy of participants in evaluating potential collisions. Figure 9 illustrates our criteria: the red curve is the measured distance between $P1$ and $P2$. The violet one is $MinDist_{pred}$

defined as follow:

$$Pred_{1,2}(t,u) = Pos_{1,2}(t) + (u-t)\overrightarrow{v_{1,2}}(t)$$

with t is the experiment time, $u \in [t;\infty]$ a parameter, $Pred_1(t,u)$ and $Pred_2(t,u)$ are respectively the predicted positions of $P1$, $P2$ in the future time u, according to their current position $Pos_1(t)$, $Pos_2(t)$ and speed vector $\overrightarrow{v_1}(t)$, $\overrightarrow{v_2}(t)$.

$$InterDistance_{pred}(t,u) = \| \overrightarrow{Pred_1(t,u)Pos_2(t,u)} \|$$

is the evolution of the distance between $P1$ and $P2$ according to the previously predicted positions.

$$MinDist_{pred}(t) = \min_{u=t}^{\infty} \left(InterDistance_{pred}(t,u) \right)$$

The plots reveal that the anticipation of $P1$ was accurate. Despite her initial faster speed and deviation to the right, she would not have been able to avoid $P2$ with a minimum predicted inter-distance below $0.2m$. Her change of strategy is revealed when $Mindist(t)$ is null at $t = 62.1s$ (from "passing in front of $P2$" to "passing behind $P2$").

By analyzing the whole set of experiments our first conclusions are:

- Reactions are observable for $MinDist_{pred}(t) < 0.5m$, which allows automatic distinction between cases where interaction occurs or not.
- Anticipation is up to several seconds before a potential collision. Our experimental conditions did not allowed us to determine an upper bound to anticipation time because of the size of the motion capture field.
- Reaction is a combination of speed and orientation adaptation. Deviations are bounded whereas decelerations can lead to a complete stop for one of the participants. This occurs especially when participants modify both their trajectory so that $MinDist_{pred}(t)$ remains below $0.5m$ (conflicting corrections). The closer the collision is in time, the more speed adaptation is preponderant.
- Interaction is an accurate phenomenon. If no collision is predicted by our criterion, no reaction appears in data. At the other extreme, participants detect interaction situations early.

We calibrate our model to synthesize interacting pedestrians trajectories in a realistic manner. Anticipation is set to 8 seconds. In order to find a trade-off between performance and precision, time is discretized in a non-uniform manner ($1s$ steps in the near future, up to $4s$ for the last period). Minimal and maximal velocities to avoid a collision are computed for a set of walking directions, allowing combination of speed and orientation variations to avoid other pedestrians. We tuned the cost function to fit experimental data in a pragmatic manner. Statistical analyses are still on-going in order to calibrate the model automatically from data. Moreover, we noticed individual factors and deeper analyses should allow us to determine a variety of individual profiles to calibrate our model.

4.2. A protocol for model validation: capturing microscopic crowd phenomenon

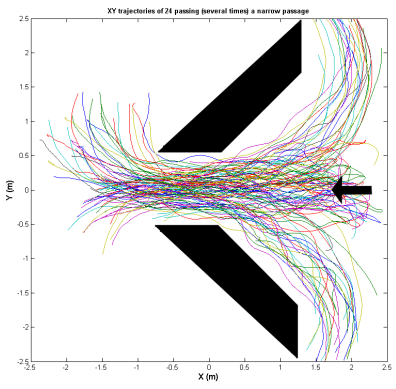

Figure 10: *A microscopic crowd phenomenon: crossing a narrow passage with funnel shape.*

In a second experiment, we capture 24 participants - each equipped with 5 markers placed on the chest - navigating in an area where we place static obstacles. The objective is to reproduce some frequently encountered situations where crowd flows meet, and to check for realistic emerging crowd behaviors in simulations (such as lane formations in corridors where opposite pedestrians flows meet). Figure 10 illustrates an example of obstacles setup, superposed with the set of recorded horizontal trajectories. Each participant crosses the narrow passage from the right to the left, and must circle the obstacle to reenter in the area from the right. We tested several setups: X-crossing with several flow directions repartitions, corridors with single or opposite flows, gate crossing, etc. We can draw first conclusions from comparisons between simulations and real data. Characteristic phenomenons could emerge from our model as observed in real data such as lane formation in corridors where two opposite flows meet, pedestrians going in same directions at X-crossing tend to group in order to facilitate their passing-by, high speed changes occur where flows meet with visibility limited by obstacles, etc. Thus, we validate qualitatively the basis of our model. However, quantitatively, simulation and real data still need to be accurately compared, and we are currently performing a deeper analysis.

4.3. Discussion

From real data, we can discuss the limitations of our model. During the calibration step, we found a balance between the cost of direction changes and speed changes (α parameter of the cost function Section 3.4) that allowed us to have similar results in simulation and experiments with identical conditions. However, we could not find a calibration that satisfies all situations. Simulation results were still valid (no collision, and no dead-locks), but realism was decreasing in some

specific situations. We think an always-valid calibration is impossible to find for two reasons. First, our anticipation time, T, is a constant parameter, whereas real humans act in a more reactive way where population is dense. Second, time-interval duration step, Δt, is also a constant parameter during the sectioning step (see Section 3.2). In other words, our model precision is constant and should depend on the situation. We believe that these two model parameters should be adapted according to local population density (at least) in order to improve the realism of the results.

5. Results

Performance We have run a variety of situations reproducing the ones we motion captured, as shown in the accompanying video. Implementation is in C++ and simulations ran on a Pentium M $1.6GHz$, $1GB$ memory, with an Nvidia 4200Go 64MB graphics card.

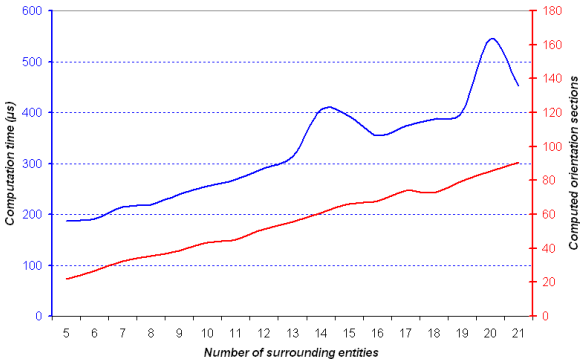

Figure 11: *Solution Performance. The blue curve plots the computation time required for solving* one *interaction according to the* number of neighbor entities *taken into account. The red curve plots the number of orientation sections created in order to compute the solution. From these plots, it is possible to compute global performances with respect to the density of people.*

The performance provided in Figure 11 does not correspond to the computation time functions of the total number of simulated entities, but to the computation time for one entity according to the density of people. In fact, this density of people implicitly defines the number of neighbor entities to take into account for the computation. Moreover, the plot limit of 21 surrounding entities has not been specified, but is a result of our benchmarks using by far more entities in different situations (an average of 100 moving agents, plus the walls). Then, the computation time for one entity, between 200 and 500 μs, may appear high compared to previous approaches, but is relatively stable for growing entities numbers. In addition, our model is only refreshed at each foot step, corresponding to an average rate of $1-2Hz$. Thereby, taking the worst cases, our model can easily handle 1,000 entities in real time, and its complexity is scalable and could be reduced to simulate more entities by decreasing the anticipation time. However, as previously mentioned, this model is integrated in a whole virtual human architecture, managing other tasks like path planning, rational behavior, animation, and so on. Based on our experiments, we are able to fully simulate and animate approximately 150 entities in real time. Moreover, the performance depends on the complexity of the environment, and on the density of the crowd, that is why it is difficult to provide representative performance considering the number of influent parameters.

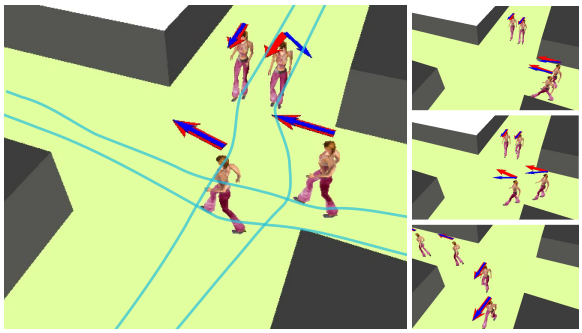

Figure 12: *Detailed view of a unit interaction case. Red arrows are desired directions θ_{des} while blue ones result from the reactive navigation. Virtual humans adapt both their speed and orientation in quite an optimal and realistic manner to avoid each other (see video for animation).*

Realism As mentioned Section 4.2, we could compare our simulations with real data, however, a quantitative validation of our model is still in progress. Our first results are promising. We solved several drawbacks observed in previous approaches. First, anticipated reactions in low density areas improve results realism (Figure 12): in such cases, conflicting states (typically, close entities face to face) never appear in simulations. Indeed, thanks to a sufficiently stable state of the environment, model predictions remain valid for a large enough interval to always avoid any close interaction. Second, in dense population situations, pedestrians do not have jerky trajectories with successive and contradictory reorientations or turn-backs. In this case, next-future predictions are accurate enough to avoid such unrealistic behaviors and dead-locks. However, distant time predictions appear to be useless because of the constantly changing situation.

6. Conclusion and Perspectives

We presented a novel approach to solve interactions between virtual pedestrians in the context of realistic crowd simulations. Our first contribution is to introduce long-time predictions in our model when accounting for other pedestrians moving around. Such a prediction allowed us to solve major drawbacks of previous approaches to this known difficult problem. Our second contribution is to propose experiments

to validate and calibrate our approach using a motion capture system. Compared to video-streams analysis, we reach a higher level of precision that allowed us to obtain a fine analysis of events occurring during real humans interactions. We also captured crowd behaviors at a microscopic level in order to validate our simulation results. Work is still in progress concerning this last point.

Our results are promising and we can identify our model limitations to get even more realistic results: this determines our future work directions. First, we want to dynamically adapt the model parameters to the variations of environment context. Especially, the prediction time-ranges and precision must fit the local population density. Second, the model does not account for social factors. We want our model to be able to consider couples of pedestrians or larger groups navigating among other pedestrians: such groups shall remain as gathered as possible during their navigation. We believe our implementation capable of supporting such evolutions easily, and work is underway. Finally, connections to the animation module must be enhanced. A feedback must emanate from the locomotion module in order to score the proposed reactions, purely in terms of realism of motion. This additive cost would help in taking realistic decisions. Recent results in animation evaluation techniques is an inspiration source to reach such a goal.

7. Acknowledgement

This work is funded by the French Research Agency ANR Project ANR05RNTL02501 *SIMULEM*, in collaboration with the French railroad company *SNCF*. We also thank members of our team who provided the animation module used in this work, *MKM* [KMA05], and who helped us during the motion capture experiments.

References

[BLA02] BAYAZIT O. B., LIEN J.-M., AMATO N. M.: Roadmap-based flocking for complex environments. In *Pacific Conf. on Comp. Graphics and Applications* (2002), pp. 104–113.

[BMdOB03] BRAUN A., MUSSE S., DE OLIVEIRA L., BODMANN B.: Modeling individual behaviors in crowd simulation. In *16th Int. Conf. on Computer Animation and Social Agents* (New Brunswick, NJ, USA, May 2003), IEEE, pp. 143–148.

[Bol81] BOLES W.: Planning pedestrian environments: A computer simulation model. *Man-Environment Systems 11* (1981), 41–56.

[GKM*01] GOLDENSTEIN S., KARAVELAS M., METAXAS D., GUIBAS L., AARON E., GOSWAMI A.: Scalable nonlinear dynamical systems for agent steering and crowd simulation. *Computers and Graphics 25*, 6 (2001), 983–998.

[Gof71] GOFFMAN E.: *Relations in public : microstudies of the public order*. New York : Basic books, 1971.

[Hen71] HENDERSON L. F.: The statistics of crowd fluids. *Nature 229* (1971), 381–383.

[HFV00] HELBING D., FARKAS I., VICSEK T.: Simulating dynamical features of escape panic. *Nature 407* (2000), 487–490.

[KMA05] KULPA R., MULTON F., ARNALDI B.: Specific representation of motions for interactive animation with several characters. *Computer Graphics Forum 24*, 3 (2005), 343–352.

[LKF05] LAKOBA T. I., KAUP D. J., FINKELSTEIN N. M.: Modifications of the helbing-molnár-farkas-vicsek social force model for pedestrian evolution. *Simulation 81*, 5 (2005), 339–352.

[LMM03] LOSCOS C., MARCHAL D., MEYER A.: Intuitive behavior in dense urban environments using local laws. In *Theory and Pratice of Computer Graphics* (2003).

[MT01] MUSSE S. R., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transaction on Visualization and Computer Graphics 7*, 2 (jun 2001).

[PDB05] PARIS S., DONIKIAN S., BONVALET N.: Towards more realistic and efficient virtual environment description and usage. In *First International Workshop on Crowd Simulation (V-Crowds'05)* (2005), VRlab, EPFL.

[PDB06] PARIS S., DONIKIAN S., BONVALET N.: Environmental abstraction and path planning techniques for realistic crowd simulation. *Computer Animation and Virtual Worlds* (2006), 325–335.

[Pes71] PESCHL I.: Passage capacity of door openings in panic situations. *BAUN 26*, 2 (1971), 62–67.

[PM78] PREDTECHENSKII V. M., MILINSKII A. I.: *Planning for foot traffic flow in buildings*, National Bureau of Standards ed. Amerind Publishing CO Pvt Ltd, New Dehli, India, 1978.

[Rey00] REYNOLDS C. W.: Interaction with groups of autonomous characters. In *Game Developers Conference* (2000).

[SGC04] SUNG M., GLEICHER M., CHENNEY S.: Scalable behaviors for crowd simulation. *Computer Graphics Forum, Eurographics'04* (2004).

[ST05] SHAO W., TERZOPOULOS D.: Autonomous pedestrians. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 19–28.

[TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 1160–1168.

[Yam98] YAMORI K.: Going with the flow: Micro-macro dynamics in the macrobehavioral patterns of pedestrian crowds. *Psychological Review 105*, 3 (1998), 530–557.