Future Virtual Particle Method for Pedestrian Navigation

Castiglione Gonzalo, Agustin Marseillan, Daniel Parisi

December 7, 2014

Abstract

We present a novel method to simulate virtual pedestrians based on the social force model. In our model, each pedestrian has a point in front of him called Future Virtual Particle (FVP) which represents where the pedestrian is headed to and at what speed.

keywords:
pedestrian, collision avoidance, future virtual particle, force model

1 Introduction

1.1 Motivation and previous work

Navigation of biological, sinthetic or virtual agents is a relevant problem in several fields such as pedestrian dynamics, moving robots and animation of characters for videogames and motion pictures.

Modelling and simulating the displacement of agents througth arbitrarily complex enviroments may be stated in an hierachical structure of mecahnisms depending mianly on the distance from the agent. This level has been named, from closer to further, as operational (walking, lowst level physical-computaitnal model for displacement), tactical (wayfinding, route chocie) and strategic (general activity planning) [Hoogonen and Bovy 2004]. These levels are not independient, factors affecting one level may impact in the following and vice versa, for example, the route choice may vary due to congestion of agents producing from previous route chice and walking behavior. Also, obstacles can impact on the operational level or tactical level depending on the particular geometry of the enviroment. The particular mechanism we want to address is the avoidance of obstacles being fixed or moving (another agent) which involves operational and tactical aspects of the navigation.

A general approach is to take an existing operational model and equip it with a higher level model which allows better and smoother collision avoidance behavior. Exisiting low level models can be taken from pedestrian dynamics field and in general this models can be classified into rule based and force based, discrete and continuous space description, etc. [Schadschneider 2009].

A famous example of continuos and force based model is the Social Force Model [Helbing 1995, 2000]. In this model the dynamic for virtual pedestrians is derived from the Newton equation's considering the total force exceterd over each agent is the result of three forces: Contact, Social and Driving Force. While the driving force points towards the final objective of each pedestrian, the social force is repulsive and acts as a kind of collision avoidance force. However this social force term introduce several artefacts in some configurations [see for example Lakoba et al. 2005, Parisi et al 2009].

Cellular automata models make use of a spacial grid, which can be occuied or empty, along with a set of rules determining the evolution and conflict resolution of virtual pedestrians moving over the cells of the grid. An emblematic celluar automata model is the one proposed by Kirchner and Schadschneider [2002].

It have been also proposed hibrid models such as the Contractile Particle Model [Baglietto and Parisi, 2011] in which a continuos description of the space is combined with a set of simple rules governig the dynamcis of the system.

The basic operational model as the ones described above can be improved if higher level mechanisms were added to manage more complex issues as eficient avoidance. Some recent examples can be found in the literature.

Karamouzas et al. (2009) proposed a method for collision avoidance modifying the social force model, basically, replacing the social force term by a new "evasive" force which tends to avoid future collisions. The magnitud and direction of this force is calculaed considering the predictions of these possible collisions.

Kretz et al. 2011 have arrised the point that the key ingredient in social force model is the driving force instead of interaction force, so in this work the authors propose a method for dinamically adjust the desired velocity following the gradient of a field given by a time map, in other words, the desired velocity is choosen as the quickest path to the objective taking into account the geometry and other agents (collision, congestions, jams, etc.). Also mounted on the SFM, Moussaïd et al. [2011] have presented a model using a "cognitive heuristics" to determine the norm and direction of the desired velocity for each agent dinamically during the evolution of the system.

In the same line, we also proposed that the navigation capacity of virtual agents should be concentrated in the on-line decision of the desired velocity. Similar to this approach, we postulate a new method derived from the methods used in robotics/artificial inteligence... The method deals with two aspects of the navigation mechanism. The first one is geometrical problem of finding intermediate goals when the final goal is not visible from the current position of the virtual agent. The second aspect is the local avoidance of other agents, which is performed via a reactive mechanism inspired in (robot navigation / artificial inteligence) aproaches ¿?¿?¿.....?¿?¿ [Ref.??? ???].

The method proposed could be mounted on different basic displacement models like de SFM or the CPM, in the present work we have chosen the first one.

1.2 Social Force Model

[Helbing 1995, 2000] presents a Model in which each pedestrian is subject to “social forces”. This forces are a measure for the internal motivations of the individual to perform certain actions. In its simplest form, the Social Force Model consist on three main terms. Each of there term represent a force that a pedestrian will suffer in order to reach his objective.

The first term describes the acceleration that a pedestrian keeps towards his desired velocity of motion. This term is known as the “Desire Force” and it’s calculated as follows:

$$F_{D_i} = m_i \frac{v_{di} e_i - v_i}{\tau}$$

In this latest formula, v_{di} represents the desire velocity for the pedestrian i, v_i is the current velocity and e_i represents the desired direction of motion of the pedestrian i and is calculated as follows:

$$e_i(t) = \frac{r_i^k - r_i(t)}{||r_i^k - r_i(t)||}$$

where $r_\alpha(t)$ denotes the actual position of the pedestrian i at the time t. The goal is represented as an area, and the vector e_i will be pointing towards the closest pont. Figure 1.1 shows en example of a pedestrian currenty moving in direction of v but correcting its trayectory towards e.

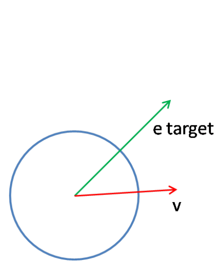

Figure 1.1: Driving force

The second term represents the fact that a pedestrian keeps a certain distance to other pedestrians and borders. This term is known as the “Social Force” and it’s caluclated as follows:

$$F_{S_i} = \sum_{j=1, j\neq i}^{N_P} A exp(-\frac{\epsilon_{ij}}{B})\, e_{ij}^n$$

In this formula the terms A and B represents constants that where determined by simulations and ϵ_{ij} represents the distance from pedestrian i towards pedestrian j. Figure 1.2 shows this vector grafically.

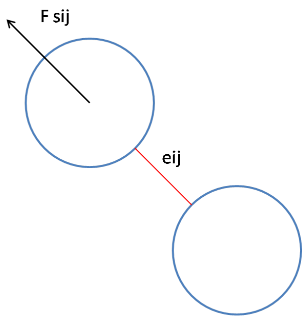

Figure 1.2: Social Force

The third terms represents the granular force. This force represents the physical force that a pedestrians feels when in collision with another objects (pedestrian or wall). This granular force term is defined as follows:

$$F_{G_i} = \sum_{j=1, j\neq i}^{N_P} \left[-\epsilon_{ij} k_n e^n_{ij} + v^t_{ij} \epsilon_{ij} k_t e^t_{ij}\right] g(\epsilon_{ij})$$

Where: $\epsilon_{ij} = r_{ij} - (R_i + R_j)$

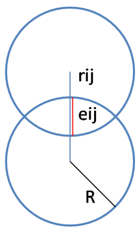

Figure 1.3: Example of two pedestrians in collision

Afterwards $F_i = F_{G_i} + F_{S_i} + F_{D_i}$ is calculated for each of the pedestrians on each simulation step and applyed to the pedestrian until it reaches his goal.

Fixed parameter values: $A = 2000\,[N]$, $B = 0.08\,[m]$, $k_n = 1.2\,10^5\,[\frac{N}{m}]$, $k_t = 2.4\,10^5\,[\frac{kg}{m/s}]$ y $\tau = 0.5\,[s]$.

While this model doesn't present a very real behaviour for pedestrians, it worked as a starting point for numerous projects.

1.3 Future Virtual Particle Model

Given that the SFM adds a fictional force on pedestrians, navigation stops imitating reality when there's a big quantity of pedestrians. Pedestrians also show a collision avoidance method that resembles magnetism, with movements that are clearly governed by the squared distance to the other pedestrian. Also, SFM didn't have the same values as well known metrics for real-case scenarios such as the flow of pedestrians going out a door and the fundamental diagram.

Because of this, we present a new model.

2 The Model

2.1 Hipotesis

The main effects that govern the motion of a pedestrian are the same as Helbin's:

1. The pedestrian wants to reach his goal in the shortest possible path.

2. The pedestrian's movement is influenced by other pedestrians. Depending on the distance between the two of them and the predicted trayectory, the pedestrian will feel the need to change his route to be able to avoid the other pedestrians. It is because of this effect that pedestrians will need to recalculate their route as new pedestrians get closer to them.

3. Movement speed will be influenced by needs.

2.2 Geometrical definition

A pedestrian is defined as follows:

- Circular shape

  Represents the space that this pedestrian occupies. Circle's radio is generated randomly to represent different types of pedestrians. The range of values is distributed uniformly in $[0.25, 0.29][cm]$.

- Long term objective

  Represented by a static area. When it is touched by a pedestrian, it is considered as accomplished. Multiple objectives can by defined in a list, in this case, each of them must be reached in order.

- Short term objective

  Called FVP, it represents a point at a relative distance from the pedestrian's center. It's a dynamic objective.

  It is defined as a1 $[kg]$ mass. Not collisionable.

- Desired speed

  Represents the speed the pedestrian would walk if he was alone. Varies randomly between$[1.2, 1.4]$ $[m/s]$.

- Reaction distance

  Maximum distance between a pedestrian and his FVP, it represents the distance at which a real pedestrian would react from an obstacle.

Figure 2.1 shows a pedestrian geomtrical definition:

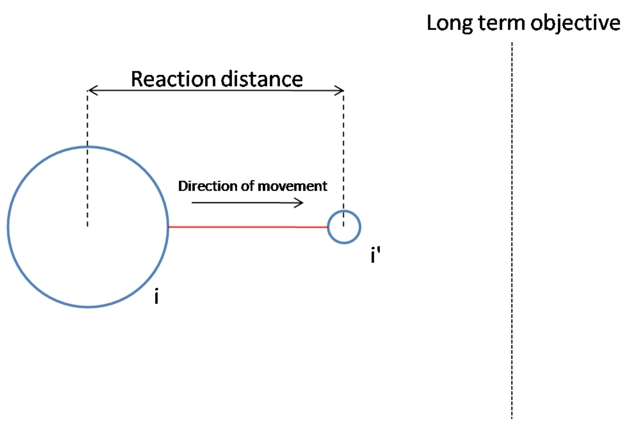

Figure 2.1: Pedestrian top view

Figure 2.2 shows all the vectorial definition that will be used:

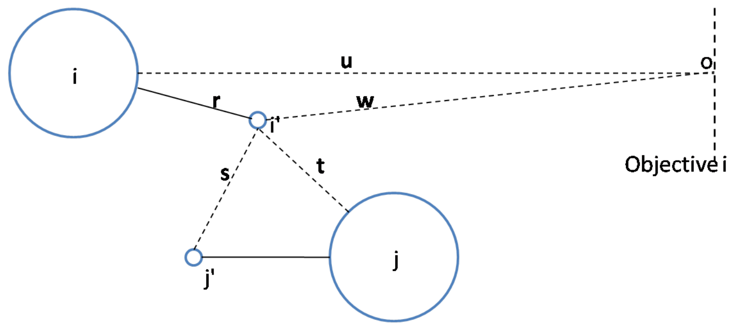

Figure 2.2: Vectorial definitions

2.3 Dynamics of the FPV Model

Each pedestrian has to reach the long term objective at some point, to ensure this, the FVP always feels the need to be aligned with the shortest path to the long term objective. On the other hand, there are sometimes obstacles in the way, which will make this impossible, in this cases, the route will have to change depending on the situation.

Each step is defined as follows:

2.3.1 Forces

- Dyamic of the FVP

  For each pedestrian i, the force that his future i' will suffer is calculated in this step. The magnitude of this force is calculated using two factors:

  The fist factor is caluclated for each of the pedestrian j ($j \neq i$) who is in the range of sight of pedestrian i. This restriction is verified using the following condition:

  $$\mathbf{r_{ii'}} \bullet \mathbf{r_{ij'}} < 0$$

  This filter is represents the fact that a pedestrian (in most cases) is not aware of what each of the pedestrians that are behind him are doing. Hence, they are ignored. The formula for

  $$F_{ext}(i) = \sum_j F_{i',j'} + F_{i'j} = \sum_j (\alpha_{ff} e^{-dist(i',j')/\beta_{ff}} + \alpha_{fp} e^{-dist(i',j')/\beta_{fp}})$$

  where α_x and β_x are predefined constants.

  Here there are two terms in the *sum*. The first term represents the fact that i does not want to be in the same location as j in the near future. Note that j' represents the predicted location of j inferred by another pedestrian (in this case, i) and because of this, it moves it's near future away of that predicted location. The second term, represents the same fact but taking in mind the current location of the pedestrian j. The same law applies to each of the walls on the simulation, the difference is that the closest point on the wall and and the future has to be calculated before applying the formula. Also, the constants α_{fw} and β_{fw} have different values.

  Figure 2.3 shows the external forces that a pedestrian i suffers because of another pedestrian (j) and also the direction in wich i desires to move.

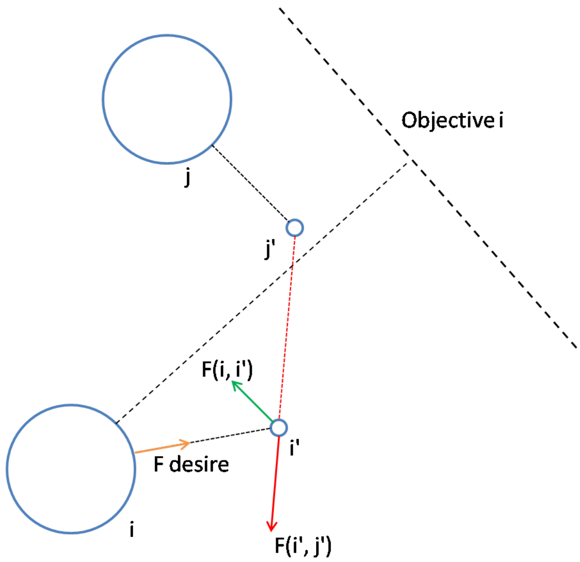

Figure 2.3: External forces affecting future i'

Then, the desire force that pedestrian i will apply to his future is calculated. This force is modeled using two springs. The first spring has its origin attached on i and its end attached on i' with a spring constant K_1. This spring will regulate the velocity that will be calculated on a posterior step at witch the pedestrian i will advance. The second spring, has its origin on i' and the end attached to $objective_i$ and has a spring constant K_2. In order to avoid an oscilatory movement (in example, with a single pedestrian), a damping γ was added to this second spring. This spring represents how much te pedestrian wants to reach his target. A larger spring constant will force a more straight path, but possible with more collitions on its way.

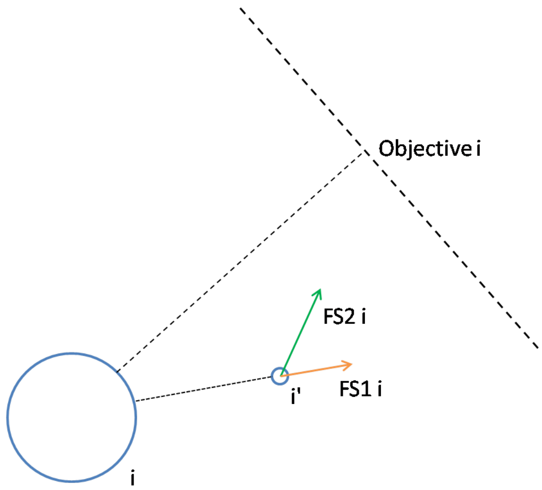

Figure 2.4: Pedestrian internal forces

Because a pedestrian always tryes to reach his goal in in the sortest possible path (hipotesis 1). If it has to take a big detour of his ideal path, it will try to reduce his velocity drastically in order to avoid making a long travel. In orer to recreate this fact, the spring constant K_2 has to be variable and also dependant of the deviation angle. This constant is defined as $K_2 \alpha \frac{1}{\theta}$, whereas $\theta$is the angle between $\overrightarrow{ii'}$ and $\overrightarrow{i - objective_i}$. The constant in this relation is δ.

After this, the final $F_{i'}$ is computed by adding all the terms. $F_{i'} = F_{ext}(i) + F_{i\,s1} + F_{i\,s2}$.

To avoid high simetry situations, a low noise $P = 10\%$ is added to F. There are two ways to apply this noise:

- Radial noise:
  * A value p is taken randomly from a uniform distribution $[-P, P]$ and calculate: $FL_{i'} = F_{i'} * p$
  * Angular noise:
    · A value $sgn = \{-1, 1\}$ is taken randomly from a uniform distribution and a value p from $[-P, P]$. Then $FA_{i'} = rotation(F_{i'}, \pi * sgn) * p$ is calculated.

  At last, we find $F'_{i'} = F_{i'} + FL_{i'} + FA_{i'}$ and apply movement equations.

- Dynamic of the pedestrian

  The pedestrian always wants to move in the direction his FVP is pointing and its magnitude is defined as F_d or desire force:

$$F_{desire_i} = m_i \frac{\frac{dist(i',i)}{dist_{react}} e_i - v_i}{\tau}$$

  Where $\tau = 0.5$

2.3.2 Algorithm

The pedestrian movement is calculated in four steps:

1. Calculate forces for each FVP.
2. Update positions for each FVP.
3. Calculate forces for each pedestrian.
4. Update positions for each pedestrian.

3 Calibration

3.1 Metrics

The results where compared to the SFM using the values proppposed by Dirk Helbing [2]. The test scenarios where crossing and hallway for they present the main two types of simmetry (90 degrees and 180 degrees).

The metrics used where:

1. Number of collisions

   When a pedestrian's body is touching another one, a counter is incremented, it won't be incremented again unless it separates and touch again. This counter represents the amount of collisions.

2. Total duration of collisions:

   When a pedestrian's body is touching another one, a counter is incremented by the time step of the simulation. This counter will represent the total time pedestrians were colliding.

3. Average walking speed:

   //TODO

4. Average travel time:

   When a pedestrian appears in the map, this time is recorded, once he reaches his goal, the travel time is calculated with this value. The average of all this times represents the average travel time.

5. Average travel distance:

   Each step, the distance a pedestrian has traveled from the source is saved. When this pedestrian reaches his goal, that distance is considered as finished and it is saved elsewhere. Afterwards, an average of all finished distances is calculated.

6. Average turn angle:

   Each step, the angle of the previous velocity of a pedestrian and the current one is saved and added to a total. When this pedestrians reaches his goal, this total is considered as finished and it is saved elsewhere. Afterwards, an average of all finished turn angles is calculated.

3.2 Values

To calibrate the model, runs varying parameters were made. A wide spectrum of values was covered, testing every combination of every possible one. After seeing clear preferences towards certain values, the values were refined within that scope. After numerous iterations of this process, the values that best suit these metrics are$\alpha = 800$ y $\beta = [0.65, 085]$ uniformly distributed.

//TODO todos los demas parametros

4 Results

Values	1	2	3	4	5	6
$\alpha = 1000$, $\beta = [0.4, 0.5]$	(1.800, 0.748)	(34.600, 8.333)	(1.024, 0.016)	(1.910, 0.022)	(2.392, 0.005)	(111.062, 19.122)
$\alpha = 2000$, $\beta = 0.08$	(5.333, 2.625)	(12.333, 8.340)	(1.052, 0.003)	(1.876, 0.003)	(2.391, 0.002)	(106.185, 13.778)

Table 1: Metrics comparing SFM vs FVPM. Average of 10 runs.

// Poner Graficos indicando distancias y esquemas del future y la particula.

5 Conclusions

// agregar al final futuras opciones que se abren con este trabajo

6 References

- [1] ... Karamouzas ...
- [2] ...Helbing ...