**Crowd Simulation for Mobile VR**

Marios Ntoulas

University Of Crete

Heraklion, Greece

ntoulasm@csd.uoc.gr

**1. Introduction**

Crowd simulation is the process of simulating the movement and dynamics of a massive amount of virtual characters called agents. It is commonly used to create virtual scenes for visual media like films and video games, but there are many applications where it is necessary to model virtual crowds such as architecture and training to improve pedestrian safety and comfort in building management. Many people have the feeling that human behavior is chaotic or at least very irregular and unpredictable. This is probably true for behaviors that are found in complex situations. However, at least for relatively simple situations, behavioral models may be developed.

2. Exposition

Simulation of crowds requires two interactions to consider, the interaction between agents and the interaction between agents and obstacles. Agents are driven by a velocity which is the sum of multiple distinct forces. These forces are not directly exerted by the agents’ personal environment but they are a measure for the internal motivations of the individuals to perform certain actions (movements). In Helbing’s Social Force Model [1] the force that drives an agent to his destination is calculated as:

F = Fdesired + Fsocial + Flookahead + Fobstacle

Thus, the required forces are, a force that push the agent to go straight to his destination, a force that forbids agents to come too close, or pass through each other, so its magnitude needs to grow when they come closer, and a force that makes agents to go around obstacles.

The idea that is presented here is simple and it is inspired by the Social Force Model and Magnetism. Each agent is considered as a positively charged particle and repels the agents. It uses the Coulomb’s law to calculate the repulsive force between agents and a modification of Coulomb’s law to calculate the repulsive force between agents and obstacles.

2.1 Elaboration

The desired force as described above, is the intention of an agent to move to his destination at full speed. In Okazaki’s Magnetic Force Model[2], goal is a negative magnetic pole. Each agent moves to his goal by the attractive force caused by the negative magnetic pole at his goal. In that way all the agents must have a common destination or an agent will be attracted by all the destinations. To overcome this the desired force of

our model is calculated as it is calculated in the Social Force Model [4], described by the equation:

Fdesired = direction \* maxSpeed

where the maxSpeed is the maximum speed of the agent and direction is a normalized vector from the agent to his destination:

direction = pd – pa / |pd – pa|

where pd is the position of the destination and pa is the position of the agent.

To calculate the repulsive forces between agents, a structure that stores all the agents was needed. Then for each agent their neighbors needed to be found, and summarize all the repulsive forces the neighbors exert to him. The neighbors of an agent are all the agents in a range called the neighborhood range. In a variation of the Magnetic Force Model [3] a Field of View was used to find the neighbors of an agent. The repulsive force that an agent a exerts to an agent b is calculated as:

Frepulsive = k \* Qa \* Qb \* direction / sqr(d)

where k is a constant Qa and Qb is the charge of the agents or in our case a factor of how much an agent repels other agents and sqr(d) is the the distance between the agents squared.

The direction is a normalized vector from a to b and it is calculated as:

direction = (pb – pa) / |pb – pa|

where pa is the position of the agent a and pb the position of the agent b.

The direction vector is needed to give a direction to the force otherwise the equation computes only the magnitude of the force.

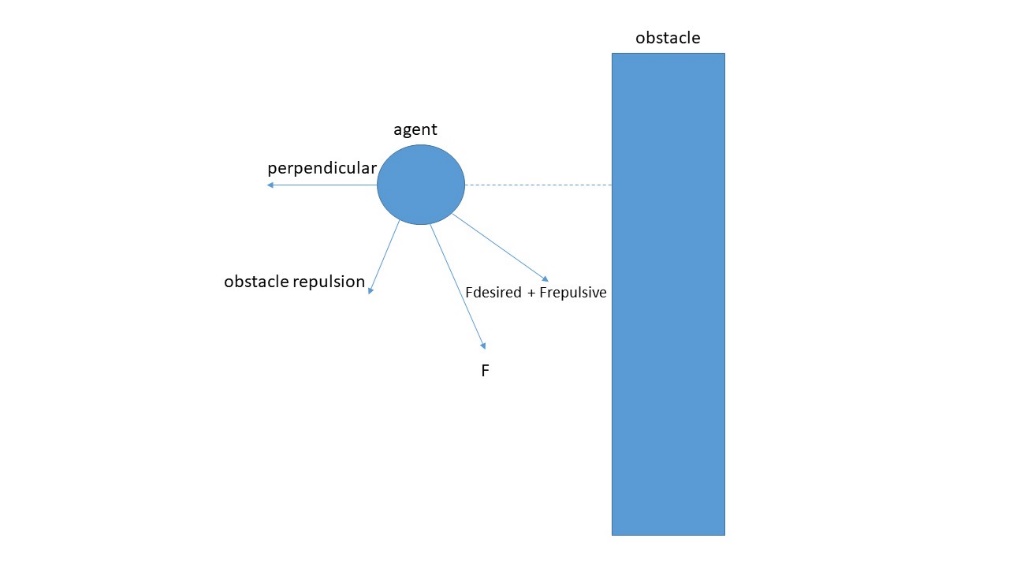
To calculate the repulsive force between agents and obstacles as before a structure to store all the obstacles was needed. In the same way, a neighborhood range was used to find the obstacles near the agent.

The repulsive force between an agent and an obstacle is calculated as:

Frepulsive = k \* Qa \* Qo \* direction / sqr(d)

Where the direction is the halfway vector between the desired force of the agent and the perpendicular vector

from the obstacle to the agent. Those forces are marked in Figure 1. This way the agent walks in parallel with the obstacle.



**Figure 1.** *Obstacle Repulsive Force and the sum of all forces.*

Also, some optimizations was needed to make the algorithm more efficient.

3. Results

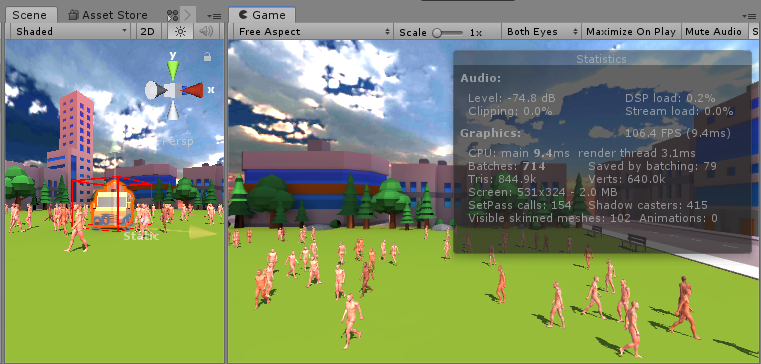
To test the Magnetic Force Model described above, a Virtual Reality application was created using the Unity Game Engine and the Google Virtual Reality Software Development Kit that simulates a crowd as you can see in Figure 2 and Figure 3. The application creates a random number, greater than 100, of different looking agents in different random positions. Each one of them has a different random path of destinations to follow inside a park of a town. Also, some static obstacles can be found in the scene.

A variety of characteristics of the agents can be controlled and thus randomized like the minimum and maximum speed of an agent or the offset of the walking animation. Changing these characteristics from agent to agent results to a more realistic simulation as each agent has its own movement behavior.

The discussed Magnetic Force Model is working well, the agents avoid collisions between them and can walk by obstacles in a natural way without affecting the visual quality. The demo application including the source code of the described Magnetic Force Model can be downloaded at: <https://github.com/ntoulasm/crowd-simulation-unity>––



**Figure 2.** *VR Crowd Simulation*



**Figure 3.** *Unity Crowd Simulation*

4. Conclusions

In conclusion, the discussed force model, is based on a really simple idea that the agents are just charged particles. It is quiet clean and straightforward to understand and to implement. Also, the mathematics are not complex and there are not so much performance-critical operations, something required for real-time crowd simulation where the amount of operations is massive, as the forces between all agents and obstacles has to be computed between each frame.

References

[1] Dirk Helbing and Peter Molnar 1998. Social Force Model for Pedestrian Dynamics.

[2] Shigeyuki Okazaki and Satoshi Matsushita 1997. A study of simulation model for pedestrian movement with evacuation and queuing.

[3] Nurulaqilla Khamis, Hazlina Selamat, Rubiyah Yusof, Fatimah Sham Ismail 2017. Magnetic Force Model Approach with Path Finding Feature for an Improved Crowd Movement Simulation.

[4] Microscopic Pedestrian Simulation:

<http://pedsim.silmaril.org/>