Reproducing crowd turbulence with Verlet integration and agent modeling

Albert Gutierrez-Milla[1,2] and Remo Suppi[2]
[1]Barcelona Supercomputing Center
[2]Universitat Autònoma de Barcelona
albert.gutierrez@bsc.es, remo.suppi@uab.cat

Abstract – *High density crowds are risk situations that already had turned some events into disasters. There are particular emerging events in these crowds that had led to dangerous situations. One important phenomenon is named "crowd turbulence". It is produced by a propagation of forces among the mass and has already been the cause of several tragedies. We present a new approach to its representation and understanding by a hybrid model composed by two parts: physical interaction among the agents, and psychological factors that produce voluntary interactions. The focus of the present work is contributing with a model able to reproduce such events in a computationally efficient way on SIMD architectures.*

I. INTRODUCTION

A crowd under high density conditions is a potential disaster situation. Understanding and developing models to characterize the crowd helps the security assessment process for building design, event planning, evacuation planning, etc. Previous disasters such as Love Parade (Duisburg), or Hajj (Mina)[1] showed that there is still a lack of knowledge and a there is a need of understanding such situations. There is a particular phenomena reproduced in the case of crowds were the pressure is propagated through the mass potentially causing crashed chests. This effect is named "crowd turbulence". We present a model capable of reproduce crowd turbulence and we implemented a simulator for SIMD architectures using OpenCL.

II. MODEL

In crowd turbulences, interactions among the bodies are described as a wave propagating forces and the mass inertia. We consider that this event can be model by particle simulation and to complete the model we include human behavior in every person. To model the physics of the movement, navigation and inertia we chose Verlet method which integrates Newton's second law of motion. For the psychological part we use Agent Based Modelling (ABM) to model the voluntary actions of the population. Consequently, bodies of the people are modeled as particles and as agents.

Using the second order central method we express the equation of motion as finite differences. Formula 1 shows the equations for coordinates in a bidimensional space. For simplicity we will not consider the mass as variable and the value of the acceleration is a constant defined as a parameter of the model and is applied until the agent reaches its v_{max}, then only the inertia phase is computed. The velocity chosen for the agents follows a normal distribution with a mean of 1.34m/s and a standard deviation of 0.26m/s[2], navigating towards a specific goal *e*. When two agents intersect the collision is solved by a simplified technique to solve inelastic collision losing kinetic energy. This is done by using a factor which depends on the intersection between two agents.

$$x_{n+1} = -x_{n-1} + 2x_n + a\Delta t^2$$
$$y_{n+1} = -y_{n-1} + 2y_n + a\Delta t^2$$

Even though agents are dragged by the mass in case of turbulence, people do not move as particles. They offer resistance to external interactions and also try to gain free space by pushing others expressing their will. Thus, we model intentional and involuntary pushes. Involuntary take place when a person is moved by the crowd and those are modeled by Verlet integration. Voluntary interactions come from psychological factors and are modeled in every agent. Moreover not every person has a tendency in pushing other. Apathy, empathy or neurotic behavior may have a direct impact in the behavior of people during evacuations. To map these non-homogeneous human treats to the model we describe their tendency to push others.

Every agent follows a path declared by a graph were there are some areas named "decisions points" which define the navigation route of the agent. These are nodes, and when the agent is close to it, the goal position *e* is updated to the next node. We assume that agents know the shortest path and there is no uncertainty added. The size of the agent is modeled by a circle with a dimeter of 0.4m and a standard deviation of 0.1m.

III. SIMULATION

To analyze the recreation of crowd turbulences we use the definition of "pressure" (Formulas 2 and 3) proposed by Yu[3] which reflects the irregular/chaotic motion attending the velocity variance and the density. This formula has been used by other studies to validate the reproduction of crowd turbulences[4]. We used as value of R the mean size of the agents.

$$\rho_i = \sum_j \frac{1}{\pi R^2} exp\left(\frac{-\|\vec{r_j} - r_i(t)\|^2}{R^2}\right) \qquad (2)$$

$$p = \rho_i Var(\vec{v_i}) \qquad (3)$$

We implemented the previously described model in a parallel simulator. The agents are initialized in the GPU and then sent to the GPU. To parallelize the simulation we selected agent division. Every thread is in charge of one agent, executing the same kernel in parallel for all of them during every time step. For the data layout we use a SoA (Structure of Arrays) instead of an AoS (Array of Structures) because the known coalescing issues. Every thread will access the other threads structures but it only will write on its own to avoid any data inconsistency. Transactions between host and GPU occur at each iteration to store the coordinates for postprocesing.

IV. RESULTS

We executed the simulator on the GPU and we also executed a sequential version on a CPU to compare the performance with the parallel version. The experimentation platform is an Intel CPU with 2.1 GHz, 8GB memory, GPU Nvidia 750 GTX, 1.1GHz 2GB memory. The compiler is GCC 4.4.7 with Nvidia libraries for OpenCL 1.2. We used the optimization flag O3 for the compilation process. The scenario we used for the simulation is a T shaped synthetic area.

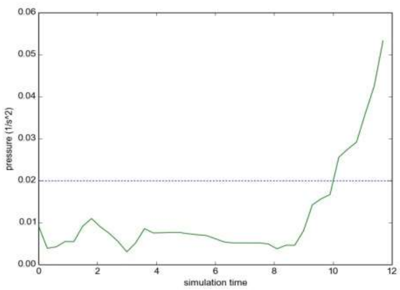

Fig. 1. Plot of the "pressure" in a crowd turbulence reproduced by our model.

Fig. 1 shows the "pressure" with the results of a crowd turbulence reproduced by our simulator. The marked threshold drawn by dots is the value of 0,02 m/s^2 which indicates the start of the turbulence. In the plot we see how the pressure lays behind the threshold from simulation steps 0 to 10. At time 9, pressure starts increasing and at time 10 overpasses the threshold starting the crowd turbulence increasing the local mass pressure and velocity.

GPU performance was compared with the sequential version executed in a CPU. Fig. 2 depicts the performance of both versions doubling the population between 512 and 4096 agent. The time is scaled in logarithmic scale. Increasing the number of agents the performance is comparatively improved in the GPU version making usage of idle SM and making a more efficient usage of the resources. Because of this, the speedup is of only 2 for 512 agents, but 11 for 4096.

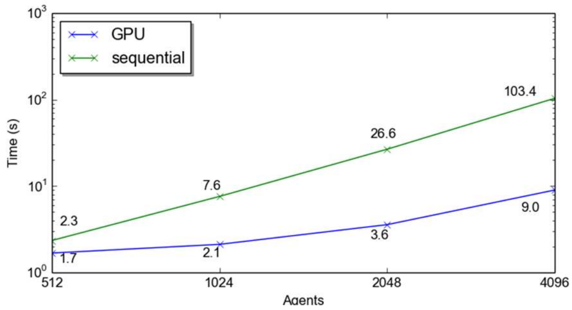

Fig. 2. Performance time in seconds of the sequential and GPU implementations of the simulator. The results are the total simulation time for 1,000 simulation steps.

V. CONCLUSIONS AND REMARKS

We presented a model able to reproduce crowd turbulences with an hybrid approach including a physical model and a psychological model. The simulation was able to reproduce crowd turbulences using the definition of "pressure" to validate the behavior. One of the main features of our model was the simplicity and the suitability for SIMD architecture. The performance of the GPU simulator showed a speedup ranging from 2 to 11 as we increase the number of agents and the work load. Future work may be focused on the improvement of the collision phase and the GPU performance as well as extending the model.

ACKNOWLEDGMENT

This research has been supported by the MINECO (MICINN) Spain under contract TIN2014-53172-P.

REFERENCES

[1] Helbing, Dirk, Anders Johansson, and Habib Zein Al-Abideen. "Dynamics of crowd disasters: An empirical study." Physical review E 75.4 (2007): 046109.

[2] Henderson, L. F. "The statistics of crowd fluids." Nature 229 (1971): 381-383.

[3] Yu, Wenjian, and Anders Johansson. "Modeling crowd turbulence by many-particle simulations." Physical review E 76.4 (2007): 046105.

[4] Golas, Abhinav, Rahul Narain, and Ming C. Lin. "Continuum modeling of crowd turbulence." Physical Review E 90.4 (2014): 042816.