Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

CROWD DYNAMICS MODELING AND COLLISION AVOIDANCE WITH OPENMP

Albert Gutierrez-Milla,
Francisco Borges,
Remo Suppi,
Emilio Luque

Universitat Autònoma de Barcelona, Bellaterra, Barcelona, Spain
Escola d'Enginyeria, Carrer de les Sitges,
Cerdanyola del Vallès, 08193, Barcelona, Spain

ABSTRACT

This paper deals with the problem of simulating crowd evacuations in multicore architectures. Manage evacuations is an important issue that involves lives, and in the case of crowds, thousands of lives. We present our model, able to hold thousands of agents walking through the scenario avoiding dynamic and static obstacles. We test how the model behaves with agents falling down, becoming an obstacle. Simulators are widely used in the area of crowd evacuations. In the present work we introduce a crowd model and a HPC simulation tool. We used OpenMP to program a multicore architecture.

1 INTRODUCTION

Evacuation planning is a key factor when big events are prepared. Crowd events are getting more and more usual and prevention is a key factor in order to avoid disasters. Human behavior is not always predictable, and evacuations are complex cases were there are a lot of cases. Crowd evacuations is a special case inside evacuations where a big population is involved (hundreds or thousands). In the particular case of crowd evacuations the complexity increases. Bodies get harmed and they become dynamic obstacles, panic stampedes, domino effect (Helbing and Mukerji 2012), Crowds have special issues as "crowd turbulences" (Moussad, Helbing, and Theraulaz 2011), stop and go waves, etc.

Drills are approaches were the repeatability and the characteristics of the agents cannot be controlled. Fire drills lack of repeatability and control of the human emotions that is why simulators are an increasingly getting the interest of the people who take decisions. Referring to the complexity of evacuations Helbing said: "Despite the sometimes more or less chaotic appearance of individual pedestrian behavior, some regularities can be found" (Helbing and Molnar 1998) which extracts patterns of motion of the agents. We use an agent based model (ABM) approach. ABM and the self organization emerging from the population behavior and can implement the models described above adapting the theoretical model to the implementation. Extreme densities is one of the most important topics, because of that we are interested in models that allow densities, instead of grid-based models. And we will use the Velocity Obstacles model which implements the free navigation among agents without collisions and the multiagent cases.

Crowd evacuation simulators have performance needs over normal evacuations. In the present work we present a crowd evacuations model able to support thousands of agents. In the model decision are decoupled and every agent takes decisions according to its environment, which is more similar to the reality than a centralized decision making system. The goal of this work is to provide a scalable software breaking data dependencies and make it suitable for multicore architectures like GPUs. Multicores have gained popularity in desktop and even mobile architectures. The paper presents the results that show the scalability of our HPC tool and the linear speedup.

2 THE MODEL

The characteristics of scape panic described by Helbing include: *People show a tendency towards mass behavior, that is, to do what other people do* and *Alternative exits are often overlooked or not efficiently used in escape situations*. Both points are related to a tend of mass behavior. Pressures come from agglomeration and interactions come from pushing. Group behavior emerges when a group of people share the same objective. In our model we describe a goal for every agent. Agent are influenced by other agents. Some agents have a exit set, and other do not. Agents are influenced by other agents with more knowledge.

All the times are independents and the past time is not analyzed. So the only steps calculated is from $t \rightarrow t+1$. The agent is modeled by a circle that represents the area. Current position, the goal, and speed retrieves the speed vector v that determines. We can represent densities. We model the knowledge (k) that the agent has about the exit. The knowledge is a value in the interval [0,1]. Agents compare their knowledge to other agents knowledge. When their knowledge is behind a threshold their speed decreases. When agents follow an agent with more knowledge their knowledge increases.

As mentioned above in the social force model, a set of social forces determine the influence in every agent in the social force model. Current position, the goal and the speed gives the vector $\vec{v}$. This vector is transposed from -60 to 60. As well as in RVO all agents always chose by default the same side. v is recalculated from v_{max}. Injured agents have a $v_{max} = 0$. δt is the simulation time step. The con is that positions are not accessed directly as in a grid representation. Every agent iterates over all the other positions. Other models[Avi] precomputed the global path in the GPU. In this model we do not include path finding algorithm. The atributes defined for every agent are the following: Speed (s), Diameter: d), Angle (ϕ), Position: (P_i) and Goal (G_i)

We implemented the model described for a high performance architecture. We model one agent as a structure with attributes that will be what defines the movement and interaction. Agents are modeled as circle and the scenario is represented by a bi-dimensional map. Agents are spread randomly among the space avoiding initial inconsistencies as collisions.

3 RESULTS AND CONCLUSION

The experiments have been carried sing population of 3.000, 6.000 and 12.000 agents. Agents move towards a bi dimensional space represented by coordinates (x,y) stored in an array of structs (AoS). The model was implemented using C and OpenMP. The simulations were carried out using 1, 2, 4, 8, 16, 32 and 64 cores. The results showed a speedup close to linear. From the research that has been carried out, it is possible to conclude that the proposed method can be readily used in practice. The model avoids collision and supports crowd clogging. Shared memory parallelization is a suitable computational model for crowd evacuations parallelization. The algorithm scales efficiently as we increase the number of cores.

ACKNOWLEDGMENTS

This research has been supported by the MINECO (MICINN) Spain under contracts TIN2011-24384 and TIN2014-53172-P.

REFERENCES

Helbing, D., and P. Molnar. 1998. “Self-organization phenomena in pedestrian crowds”. *arXiv preprint cond-mat* 9806152.

Helbing, D., and P. Mukerji. 2012. “Crowd disasters as systemic failures: analysis of the Love Parade disaster”. *EPJ Data Science* 1.1:1–40.

Moussad, M., D. Helbing, and G. Theraulaz. 2011. “How simple rules determine pedestrian behavior and crowd disasters”. In *Proceedings of the National Academy of Sciences*, 6884–6888.