Hinomiyagura Infrastructure Competiton TDP: Platform of rescue simulation using GPGPU

Toshinori Niwa, Masaru Okaya, and Tomoichi Takahashi

Meijo University, Aichi, Japan
{e0930064, m0930007}@ccalumni.meijo-u.ac.jp, ttaka@ccmfs.meijo-u.ac.jp
http://sakura.meijo-u.ac.jp/ttakaHP/Rescue_index.html

Abstract. We propose a new platform that consists of new traffic simulator and scenario generator. The traffic simulation system using GPGPU that enables to simulate rescue and evacuation simulation with more agents and faster than the present system. And it can simulate agents' motions in a three-dimensional map. Our proposal provides a platform to widen the applications of RCRS, such as crowd evacuation from buildings. The evacuation simulations help security offices to prepare manuals for emergencies.

1 Introduction

RoboCup Recuse Simulation (RCRS) system was designed to achieve the the objectives of RoboCup Rescue project at 1998 [2]. RCRS version 0 and 1 systems have been used in Rescue simulation league. It has been recognized that there are issues to realize simulations of rescue operations and human behaviors at disaster situations of a kilometer size area with the same number of people and facilities in the area. the objectives intended at the beginning of the project [6].

Didaitoku project started to solve the issues by dividing the area into small areas [7]. The target of the Didaitoku project was to simulate evacuation of 10,000 people and rescue operations on real 4km square area. The project revealed that a lot of computation power is required to simulate rescue and search operations in a realistic scale.

In RoboCup Rescue community, various kinds of ideas have been proposed and have been implemented [4][5]. At 2009, Java based kernel(version 1) was proposed at Infrastructure competition at Graz. The version 1 system was adopted as a platform of the competition and has been used, however, the basic architecture of the RCRS version 1 remained the same as the previous version 0. So it is difficult to simulate disaster situations with large number of agents involved and evacuations from three dimensional buildings.

Our team has participated in Rescue Simulation league with a dream that RCRS will become one of decision support systems for rescue operations and tools for security offices to save people and to decrease damages from disasters. We have proposed following ideas to realize our dream: map generation system form public GIS data, a traffic simulator (that have been used in Ver.1 system), a flood simulator, and some tools to improve RCRS functions. This year, we propose a new platform that consists of new traffic simulator and scenario generator. The traffic simulation system makes use of GPGPU that enables to simulate rescue and evacuation simulation with more agents and faster than traditional one. And it also supports three dimensional (3D) data and can simulate agent motions on open spaces and transits from one floor to other floors in a building. The proposal provides a platform that can simulate agent behaviors inside buildings and widen the application of RCRS, such as crowd evacuation from buildings. The evacuation simulations help security offices to prepare manuals for emergencies.

2 GPGPU based Evacuation Simulation

2.1 Architecture of RCRS and traffic simulation system

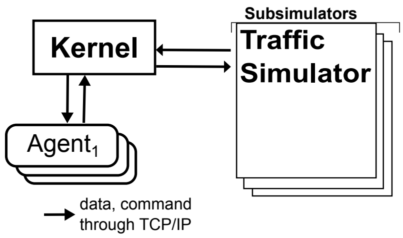


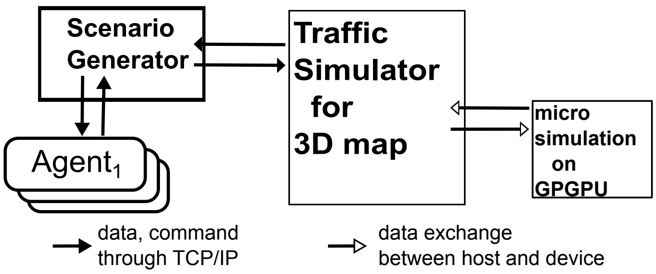
Fig. 1. Architecture of RoboCup Rescue simulation

Figure 1 (a) shows a basic architecture of RCRS. RCRS consists of kernel, sub simulators and agents. They are connected with TCP/IP protocol, so they can be executed on one computer or on several computers that are connected with a network. This architecture makes RCRS run on distributed computers when computer resources are needed. Actually, at the beginning of Rescue simulation leagues, several computers were used to run simulations at the competitions.

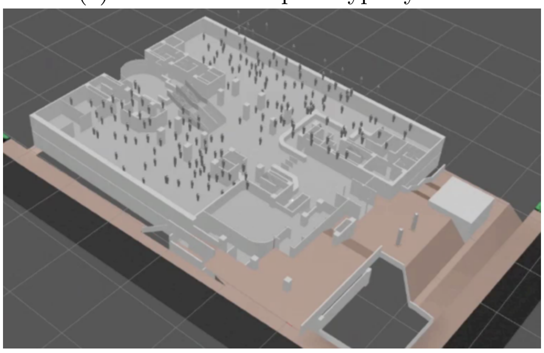
Kernel exchange data and commands with agents and sub simulators at every simulation step (Δt). Δt is a time of agent's sense-decision-action cycle. Traffic sub-simulator of version 1 is a micro simulator that calculates the motions of agents in one simulation step (Δt) at $\Delta\tau$ step that is finer than the step Δt, $\Delta t = N_{step} \times \Delta\tau$. In default setting, N_{step} is 6,000. The motions of agents are calculated with Helbing social force model [1]. The social force model is one of particle models and calculates the movements of agents to destinations that are determined in a sense-decision-action cycle. And it enables calculation of motions on open areas that were not possible to be calculated in network model of version 0.

The computations of agents' position have been done for every agents and are iterated N_{step} times in one Δt. Simulations involving a crowd of agents requires computations of $O(n^2)$ where n is the number of agents, because there are message exchanges between agents and calculation of repulsive forces from near agents. More than one thousand people are involved in real situations. It requires a lot of computer resources: CPU powers, memory and communications among the agent.

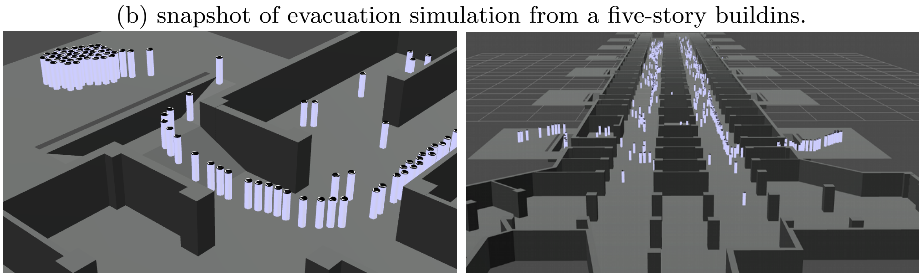
2.2 Prototype System of GPGPU based traffic simulator



(a) architecture of prototype system



(b) snapshot of evacuation simulation from a five-story buildins.



(c) snapshot of evacuation simulation from a subterranean shopping mall. (left: a stair way to ground level from street in the mall. right: a birds-eye view of the mall.)

Fig. 2. Prototype system

General-purpose graphics processing unit (GPGPU) is the utilization of a graphics processing unit (GPU) and provides a parallel computing platform for computation done by the central processing unit (CPU). Open Computing Language (OpenCL) is a framework for programming on GPGPU [1]. Figure 2 (a) shows an architecture of our system that we proposed at RoboCup2011. The system is a subset of RCRS and specializes in crowd evacuation at emergencies. Figure 2 (b) and (c) are snapshots of a thousand agents' evacuation simulation from a five-story building and subterranean shopping mall.

Our prototype system consists of three followings:

agent is the same as RCRS version 1.

3D traffic simulator is based on the traffic simulator of version 1 that we developed at 2009. The new features of the traffic simulator are (1) it can handle the movements of agents in three dimensional buildings, and (2) it can use GPGPU to calculate the social forces of agents and the behaviors of mass agents in parallel. Program codes executed on a host at every Δt are written by Java and codes executed on a device (GPGPU) at $\Delta \tau$ are written by OpenCL.

scenario generator plays the same role of the kernel of RCRS and controls related the three dimensional data and commands, for example, oral communication and vision are limited on the same floor. The initial positions of agents are also set as three dimensional coordinates by this scenario generator.

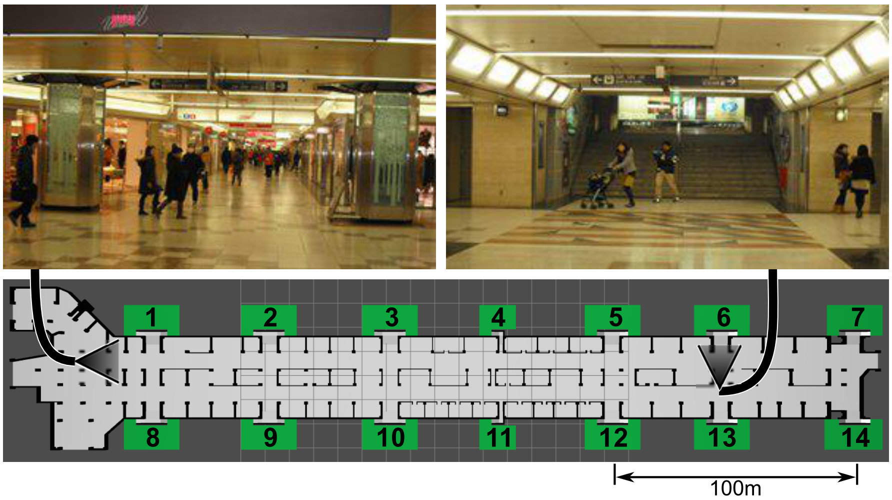
2.3 Simulation Experiments and Performance

Figure 3 (a) shows one of the subterranean shopping malls in our city. About 90 shops are in three rows, and two main streets are between the rows. Exits to the ground are located every 50m. A thousand agents are randomly situated in the mall. Fire alarms are announced to the agents and they start evacuation. Two different evacuation messages are announced. Message 1 says to evacuate by using exits number 1 and 8. Message 2 is to evacuate using nearest exits except the exits number 2 and 10. The scenario shows that good guidances make evacuation smooth and shorten the evacuation time. Such simulations provide good tools to make manuals at emergencies.

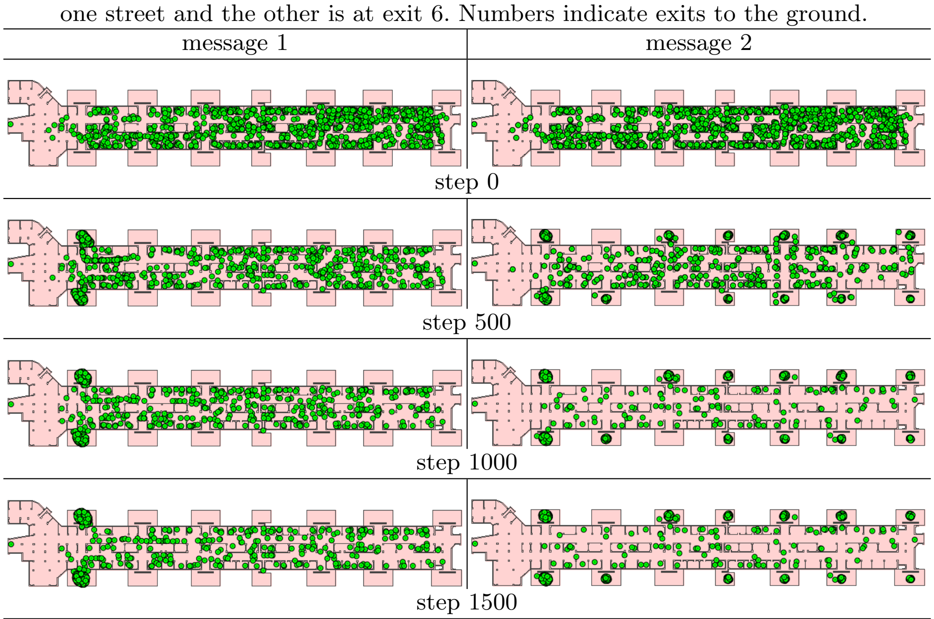
It takes time and memory to exchange data between CPU and GPGPU. Map data is transferred to GPGPU (the traffic simulator) only at start time. It means that conditions of roads are buildings remain the same condition at the start time. Simulations are run on one PC with Fedora 14. CPU is Intel Core i7-3960X 3 (30GHz, 6core, 64GB memory (heap 48GB)) and GPU is NVIDIA Tesla C2075 (448core, 1.15GHz, 6GB memory).

Figure 4 shows order of thousands simulation can be calculated. The y axis show the performances of execution of a Δt. Line (A) shows simulation results

[1] There are other software environments for GPGPU, such as CUDA. We use OpenCL because it is a vender neutral environment and it runs programs without GPGPU hardware.



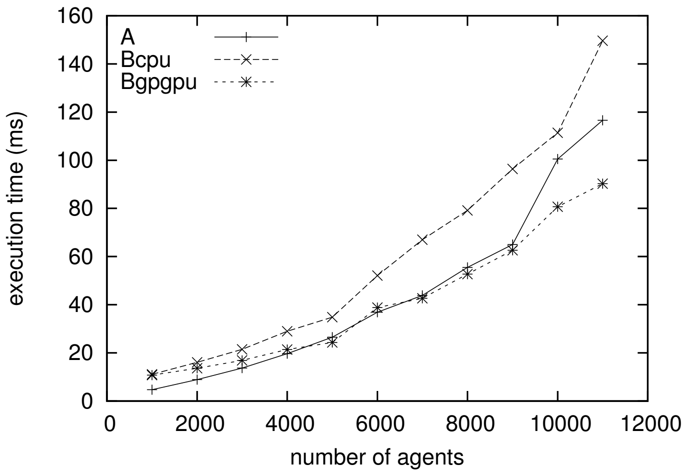
(a) Layout of subterranean shopping mall and photos: one is taken at the left end of one street and the other is at exit 6. Numbers indicate exits to the ground.



(b) Time sequence of evacuation behaviors

Fig. 3. Simulations of evacuations at subterranean shopping mall

of RCRS version 1. Lines (B_{gpgpu}) and (B_{cpu}) are result of simulation coded with OpenCL and run with GPGPU and without respectively. The performance is trade-off cost of data transfer between CPU and GPCPU and the parallel programming in GPGPU. It explains why the simulation by OpenCL without GPGPU(line (B_{cpu})) is the worst. On the other hand, the merits of paralell computation are shown by that simulations by OpenCL with GPGPU(line (B_{gpgpu})) become fastest as the number of agents becomes large.



x-axis: number of agents, y-axis: execution time of Δt

Fig. 4. Execution time of simulation per step.

3 Release Schedule

We release our proposing simulation system on our web site as follows:

2013 August: our prototype system (as in a configuration of Figure 2 (a)) .

4 Discussion and Summary

Figure 5 shows a future architecture of RCRS when RCRS maintenance team adopt our this year propose. While, as mentioned in Introduction, ideas applying distributed computing or parallel computing to RCRS have been proposed, RCRS has not scaled up enough to simulate situations involving a large number of people. Because communications among agents cover all the area and it makes difficult to divide the target ares into small ones.

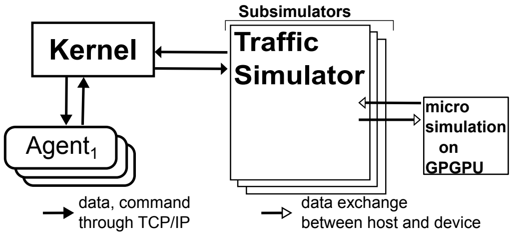


Fig. 5. Architecture of RoboCup Rescue simulation with GPGPU

There are many points to deserve considertion in using GPGPU, for example, task allocation between CPU and GPU. Our paper of RoboCup Symposium 2013 gives a detailed explanation [3]. Our proposal provides followings;

1. simulation of crowd evacuation from three dimensional facilities is possible,
2. with GPGPU, simulation is faster than the present RCRS.

References

1. D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *NATURE*, 407:487–490, Sep. 2000.
2. H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE International Conference on System, Man, and Cybernetics*, 1999.
3. T. Niwa, M. Okaya, and T. Takahashi. Parallel computation using gpgpu to simulate crowd evacuation behaviors - planning effective evacuation guidance at emergencies -. In *RoboCup Symposium 2013 (to be appeared)*, 2013.
4. C. Skinner and M. Barley. Robocup rescue simulation competition: Status report. In *RoboCup 2005: Robot Soccer World Cup IX*, pages 632–639. Springer, 2006.
5. T. Takahashi. http://sakura.meijo-u.ac.jp/ttakaHP/MyPresentation/agentCompetition09.pdf.
6. T. Takahashi, S. Tadokoro, M. Ohata, and N. Ito. Agent based approach in disaster rescue simulation - from test-bed of multiagent system to practical application. In *RoboCup 2001:Robot Soccer World Cup V*, pages 102–111, 2002.
7. I. Takeuchi, S. Kakumoto, and Y. Goto. Towards an integrated earthquake disaster simulation system. In *First International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster*, 2003.