Multi-Agent Simulation of Civilian's Evacuation

Yutaka Ishimoto, Hiroyuki Kobayashi, Shigehiro Hashimoto

Department of Biomedical Engineering, Osaka Institute of Technology,
Osaka, 535-8585, Japan
hashimoto@bme.oit.ac.jp http://www.oit.ac.jp/bme/

ABSTRACT

Civilian's evacuation has been simulated with a multi-agent system to propose an evacuation plan in a megalopolis. In the simulation, floating-point number coordinates are employed to minimize calculation errors. A megalopolis is comparted into small areas to be distributively computed by numerous computers. Behavior of human in evacuation is characterized as "an agent", which is a robot program that acts autonomously. A global city structure is modeled as topography data of a "map". The topography data, which consist of "nodes" and "edges", are input into virtual space on a computer in real number coordinates. Nodes and edges construct routes for escape, which correspond to a crossing of a road and to a straight-line road between nodes, respectively. The agents evacuate with circumventing obstacles and another agent in human-like behavior on the designed simulation primitive, which is a unit of distributed processing. The multi-agent simulation system might be effective to simulate civilian's evacuation in a megalopolis.

Keywords: Biomedical Engineering, Simulation, Multi-Agent Simulation, Evacuation, Distributed Processing, Floating-Point Number Coordinates, Computer Clustering

1. INTORDUCTION

Natural disasters occurred frequently all over the world recently. Civilian's smooth evacuation is one of the most important themes to be studied to minimize damage in the megalopolis. Evacuation is not easy in the complicated structures of megalopolis, which causes a panic and secondary disasters in civilians.

Although various simulations have been proposed on evacuation, some problems remain to be solved [1, 2]. Because of the great mass of data and complicated structures of megalopolis on evacuation, employing integer coordinates causes accumulated error, and mass of calculation exceeds capability of a computer.

In the present simulation, floating-point number coordinates have been employed to minimize calculation errors, and human evacuation has been simulated with agents in a simulation primitive.

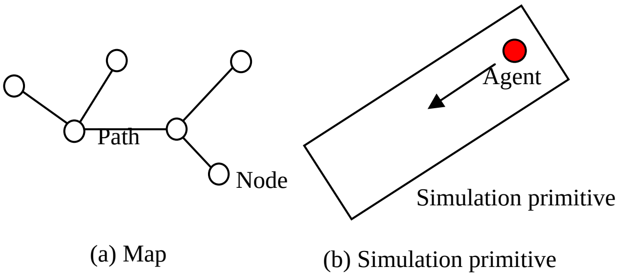
2. ARCHITECTURE OF SIMULATOR

Multi-Agent Simulation System

Behavior of human in evacuation is characterized as "an agent", which is a robot program that acts autonomously. An appropriately designed agent makes the simulation system reliable.

Virtual Space

An agent moves in the virtual space. A global city structure is modeled as topography data of a "map". The topography data, which consist of "nodes" and "edges", are input into virtual space on a computer in real number coordinates. Nodes and edges construct routes for escape, which correspond to a crossing of a road and to a straight-line road between nodes, respectively (Fig. 1). Each node has a node ID (identification) number relating to a coordinate,



(a) Map (b) Simulation primitive

Fig. 1: Virtual space.

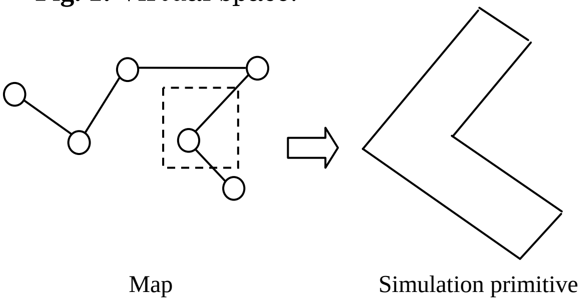
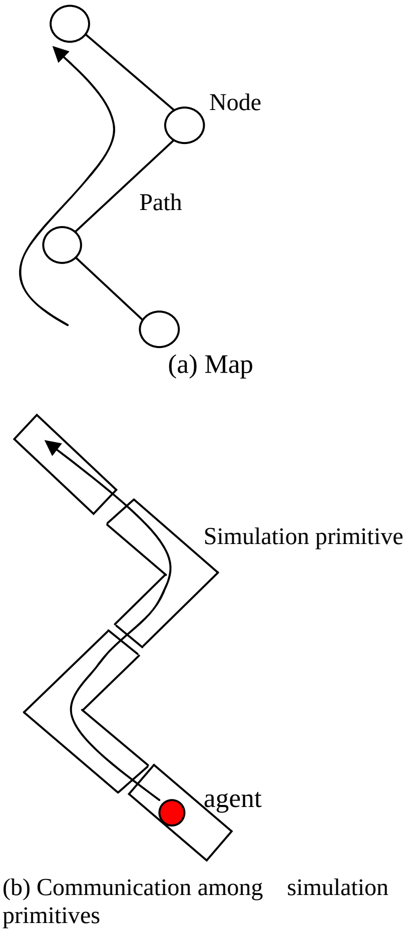


Fig. 2: Simulation primitive.



(a) Map

(b) Communication among simulation primitives

Fig. 3: Simulation primitive constructs virtual space.

and edge has width.

Simulation Primitive

A "simulation primitive" corresponds to a processing unit in distributed processing. The simulation primitive is converted from the map data. Agents move in the simulation primitive.

The map data have width and a coordinate. The simulation primitive of polygon is made with these data. The simulation primitive is defined in the area from the middle of a node to the middle of an adjacent node (Fig. 2). The simulation primitive of polygon consists of an ID number, apical coordinates of the polygon, agents in the polygon, and ID numbers of the adjacent simulation primitives. The simulation primitive constructs virtual space like mosaics (Fig. 3).

When the number of agent increased, all simulation primitives are not handled with one computer, but with multiple computers (Fig. 4).

Agent

Each agent has an ID number, speed, size, coordinates in a simulation primitive, temporary objective coordinates in a simulation primitive and an ID number of an objective simulation primitive.

The ID number of agent belongs to the agent itself and does not change during simulation. The system of coordinates in a simulation primitive is the x-y coordinates at the location of the agent. The temporary objective coordinate

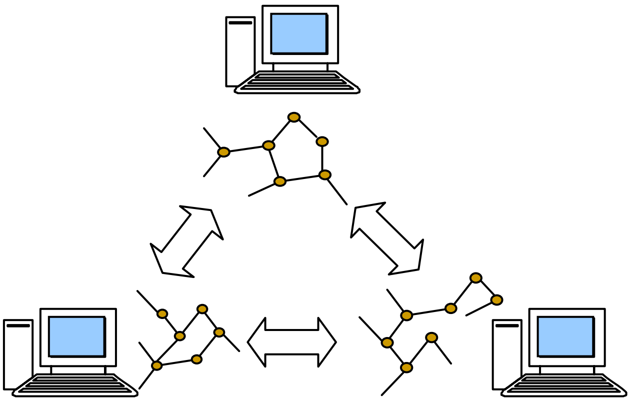


Fig. 4: Computer clustering for Distributed Processing.

in a simulation primitive is the coordinate nearest to the objective adjacent simulation primitive. The agent moves in a local area, and circumvents an obstacle. The agent chooses the next point to move for evacuation, and moves to the appointed coordinate, when there is no obstruction. The agent circumvents an obstacle, when the agent encounters obstructions: another agent or an obstacle.

Algorithm to Circumvent Obstacle

Define the coordinates (x, y), where an agent is seated, when the central coordinates of the obstacle on the way of the agent's moving direction are (x_0, y_0).

$$dx = x - x_0, dy = y - y_0 \qquad (1)$$

The agent moves to the point nearer to the destination, after calculation of the coordinates of two points: $(x - dy, y + dx)$ and $(x + dy, y - dx)$. These coordinates are located in a tangential direction against the obstacle, which enables the agent to circumvent the obstacle. There is a possibility for two agents to choose the same direction to move, when two agents encounter each other. The agent choose the moving direction at random to avoid two agents moving to the same direction, crossing each other's path, falling into "deadlock" (Fig. 5).

Test

A simulator was developed in the environment shown in Tables 1 & 2.

A simulation was performed, whether an agent could avoid an obstacle and another agent, and advance to the refuge direction in the simulator developed.

Ten agents evacuate from the left side to the right side, and the other ten agents evacuate from the right side to the left side; where each agent moves at the speed of 0.4 m/s, and has a radius of 1 m in the simulation (Fig. 6).

Table 1: Development environment of main simulator

Development language	C++
Software	GNU C++ Compiler

Table 2: Development environment of simulation viewer

Development language	Java
Software	Java Development Kit

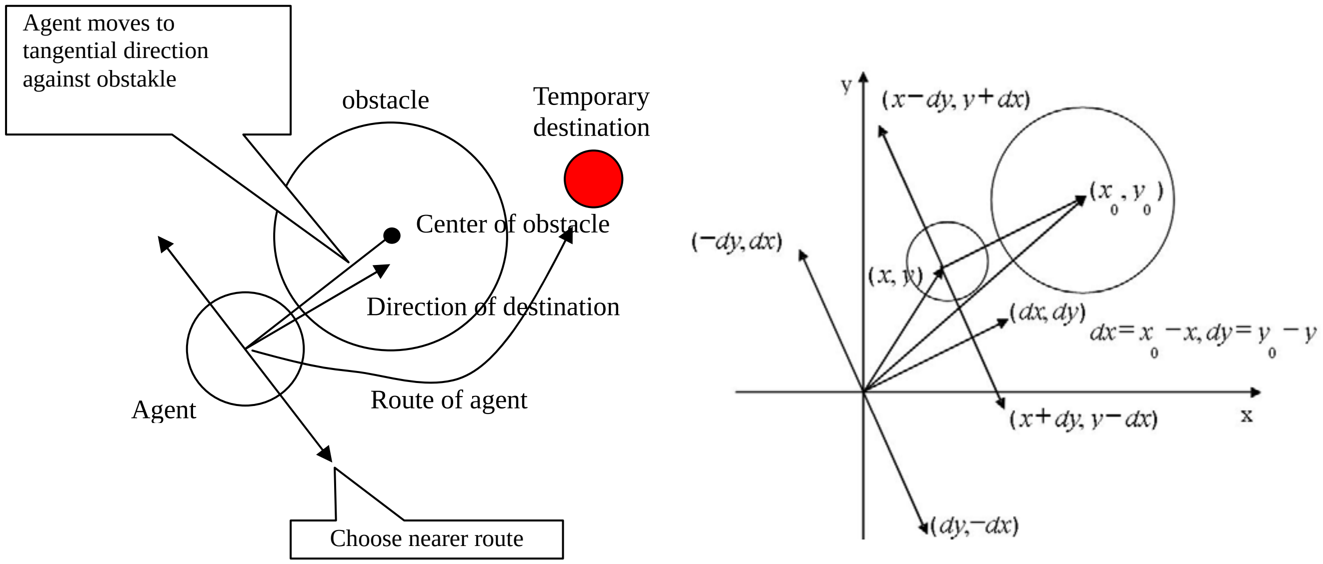


Fig. 5: Algorithm to circumvent obstacle.

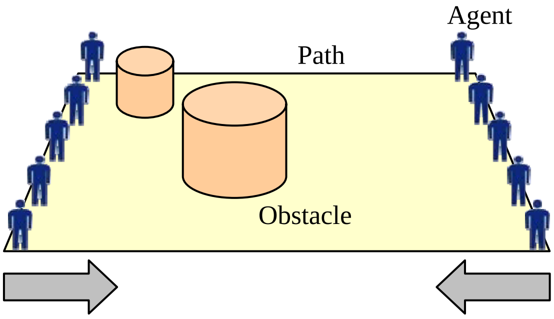


Fig. 6: Simulation to circumvent obstacle.

3. RESULTS AND DISCUSSION

The results of the simulation show that every agent circumvents obstacles smoothly without deadlock like a human being, and that every agent moves to the evacuation direction (Fig.7). The simulator with the floating-point number coordinates successfully simulated evacuation of human with agents in the simulation primitive. The algorithm for an agent to circumvent an obstacle is confirmed in the simulation.

In the present simulation, floating-point number coordinates have been employed to minimize calculation errors. The global simulation has been divided into comparted simulations for distributed computing under clustering computers, because approximately 500,000 is the limit number for agents to be handled with a current computer [1, 2].

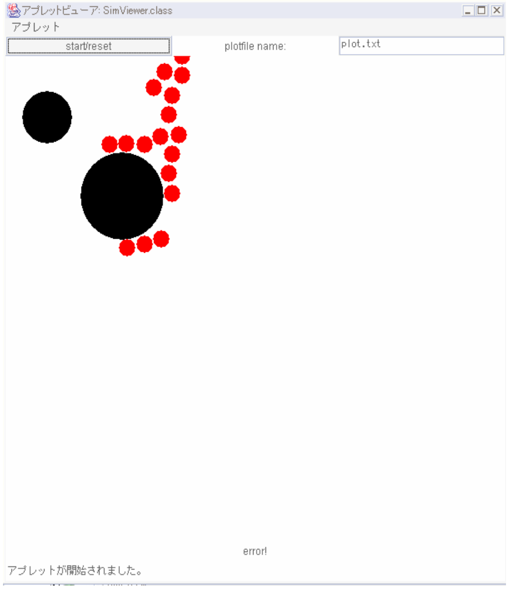

アプレットビューア: SimViewer.class
アプレット
start/reset
plotfile name:
plot.txt
error!
アプレットが開始されました。

Fig. 7: Agents are circumventing obstacle.

4. CONCLUSION

In the present study, the agents evacuate with circumventing obstacles and another agent in human-like behavior on the designed simulation primitive, which is a unit of distributed processing. The multi-agent simulation system might be effective to simulate civilian's evacuation in a megalopolis.

REFERENCES

[1] M. Fujioka, K. Ishibashi, "Multi Agent Simulation Model for Evaluating Evacuation Management System against Tsunami Disaster", *Journal of Architecture, Planning and Environmental Engineering, Transactions of AIJ*. No. 562, 2002, pp. 231-236.

[2] M. Fujioka, K. Ishibashi, H. Kaji, K. Tsukagoshi, M. Fujioka, "The Information Dependability for Evacuee on Multi-Agent Based Simulation Model", *4th Workshop on Agent-Based Simulation*, 2003, pp. 64-69.