Pedestrian Walking Agent Model Trained by Multilayered Neural Network and Parameter-free Genetic Algorithms

SHIMADA Yoko[1], KAWAMURA Hiroshi[2], TANI Akinori[1] and TAKIZAWA Atsushi[3]
[1]Graduate School of Science and Tech., Kobe Univ.
[2] Dept. of Architecture and Civil Eng., Faculty Eng., Kobe Univ.
[3]Dept. of Architecture and Architectural Eng., Faculty Eng., Kyoto Univ.
1-1, Rokkodai, Nada, Kobe, Japan 657-8501
030t027n@y03.kobe-u.ac.jp

***Abstract*—This paper describes a construction method of a pedestrian agent-based model used in simulations that is useful for architectural planning. In this paper, this agent model can learn rules of walking by Neural Network and Parameter-free Genetic Algorithms. They walk to destinations avoiding collision with each other in a straight passageway. It is shown that between the training by using one agent and that by using multiple agents, there are large differences on walking rules. Walking rules learned by using multiple agents make agents keeping to the right or left side. It can be considered that this tendency is observed generally as a kind of society rules.**

I. Introduction

In architectural planning, it is important to estimate and calculate performances of objective spaces such as safety, comfort, efficiency, functionality and so on. However, it is difficult to estimate these performances because human behaviors are complex especially in planning buildings used by the general public.

As for one of solutions to this kind of problem, simulations based on models of human behaviors reflecting human patterns of behaviors are employed in many researches, and usefulness of those systems is also proved[1][2].

In preceding studies in the authors' laboratory, models of human behavior based on agent model are introduced[3][4]. In these models, agents have abilities for physical obstacle avoidance and selection of walking routes assumed in each action. These methods are applied to simulations of human behaviors in a student hall and a museum. However, it is impossible to set up rules based on various behaviors of avoidance and selection.

Therefore, in this paper, an auto learning method on human behavior rules of agent-based model is proposed by using multilayered neural network. In this system, the agent model can learn behaviors to avoid obstacles and select routes to reach a destination place without collision with each other in a passageway. As for training method of the neural network, a kind of reinforcement learning methods is employed by using Parameter-free Genetic Algorithms (PfGAs).

II. Space

In this paper, a straight passageway is employed as an objective space as shown in Figure 1. As for objective human behavior in this space is assumed to be only walking. Agents, which learn behavior rules of walking, appears at the right-hand side and move to the destination defined at the left-hand side. Another models are assumed to be from the right-hand side to the left. Agents in this space are assumed to recognize both sides of the passageway as walls and try to move from the starting point to the destination as fast as possible.

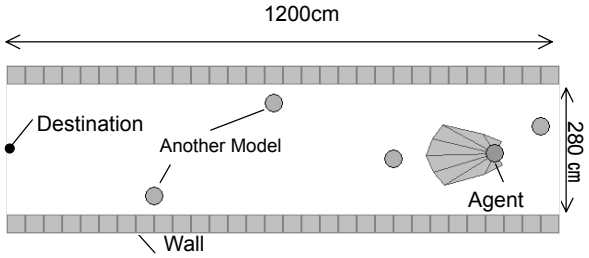


Figure1 Assumed straight passageway

III. Agent

A. Fundamental frame

Each agent model is represented as a circle with radius 20 cm, and has mass, velocity and its own position as a positional vector described with real numbers. The action of each agent is determined every 0.1 second and it moves to the determined place.

In order to determine an action in the next step, the agent has sensors to observe directions and distances of close-by obstacles. A shape of the sensor is assumed to be a polygon like a shell as shown in Figure 2. Each sensing area from A to H discriminates obstacles independently. Judging observation results of these sensors are used to determine the action of an agent.

Table 1 shows positions of apexes of the sensor, points 1 to 9, in Figure 2. In Table 1, 'ANGLE' denotes an angle to traveling direction of an agent, and 'DISTANCE' denotes a distance from the center of the agent.

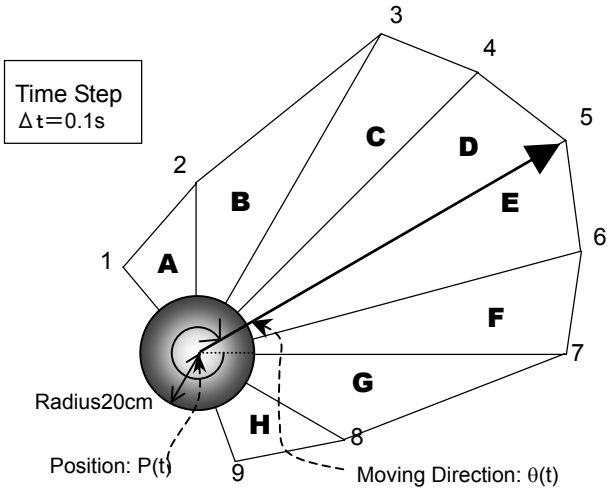


Figure2 Agent

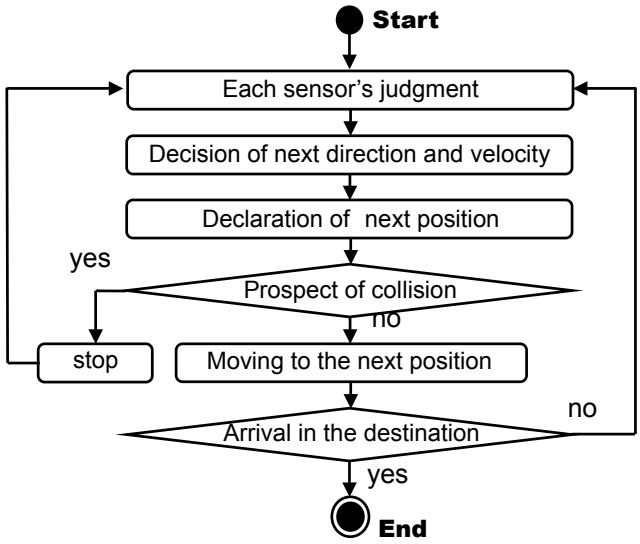


Figure3 Flow chart of agent walking behaviors

Table1 Position of sensor's apex

	ANGLE	DISTANCE
1	Left100 degrees	50 ㎝
2	Left 60 degrees	70 ㎝
3	Left 30 degrees	140 ㎝
4	Left 15 degrees	180 ㎝
5	0 degrees	200 ㎝
6	Right15 degrees	180 ㎝
7	Right 30 degrees	140 ㎝
8	Right 60 degrees	70 ㎝
9	Right 100 degrees	50 ㎝

B. Walking

The starting points and the destinations of agents are given beforehand. Agents learn human walking behaviors from starting point to the destination. In the real world, human is considered to set his destination himself and walk aiming at some sub-goals through some routes to the destination. However, in order to simplify the learning situations, these features are not taken into account.

Figure 3 shows a flowchart of walking behaviors of an agent. The agent determines a moving direction and velocity in the next step based on the information from the sensor and moves to a proper place. In this system, bumping of multiple agents is checked before moving. If collision between agents occurs, those agents are stopped and moving directions and velocities are calculated again. Agents repeat the procedure mentioned above until reaching their destinations.

C. Neural network

In this system, an agent is not given walking rules beforehand but able to learn rules for itself by multilayered neural network. Figure 4 shows a configuration of proposed multilayered neural network with 3 layers.

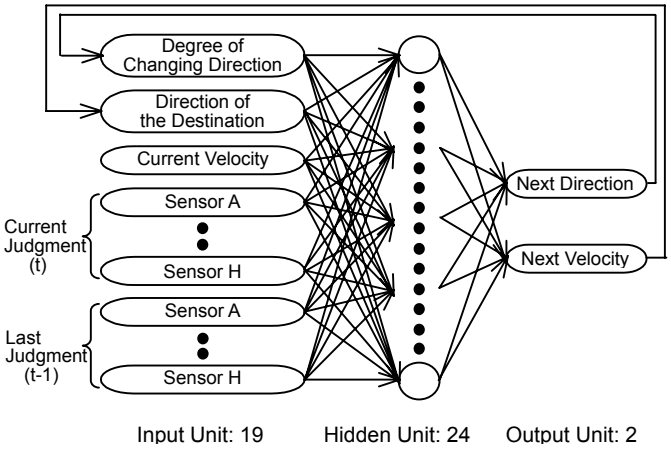


Figure4 Structure of multilayred neural network

The Input/Output (I/O) values of this network are assumed to be real numbers between 0 and 1. Details of units in the I/O layers are explained in the following.

a) Units in the input layer

*Degree of changing direction

One of the outputs of this neural network 'Next direction' at the last step is input.

*Direction of the destination

Relative direction of the destination observed from an agent is input. Absolute x-y coordinate is converted into x'-y' coordinate whose origin is an agent's position and y' direction is moving direction as shown in Figure 5. Relative direction angle of the destination between 0 and 2π in this coordinate is converted to a real number between 0 and 1.

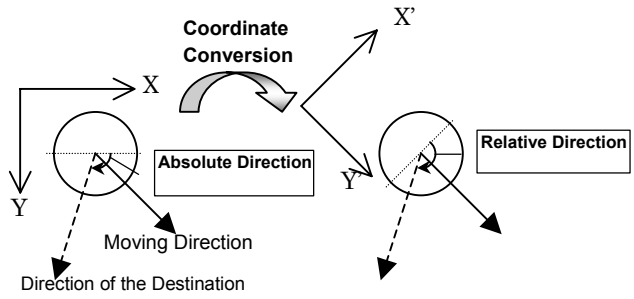


Figure5 Coordinate conversion

*Current velocity

Current velocity is defined between 0 to 0.1 m in 0.1 second and decuple value of this is input.

*Each sensor's judgment

If there are no obstacles in the area of a sensor, 0 is input. When the center of the other agent is in this area, distance (d_o) between both agents is estimated. A value 'a' defined as equation (1) is calculated and is input.

$$a = 1 - d_0 / Max(d_o) \quad (1)$$

Here, $Max(d_o)$ is the maximal value of d_o. The more obstacles approach, the more this input values become large. When an agent touches obstacles, 1 is input. When two or more obstacles exist in the area of a sensor, the nearest one is selected as a target.

For example, in case of Figure 6, 'a' value calculated by equation (1) is input by the judgment of the sensor D, and 0 is input by judgments of sensors A to C and E to H.

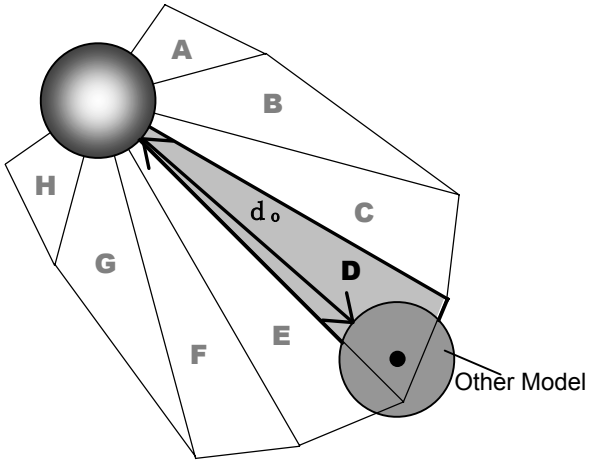


Figure6 Example of sensor judgment

b) Units in the output layer

*Next velocity

A continuous value output from the neural network between 0 and 1 is multiplied by 0.1 and the next velocity in every 0.1 second is obtained.

*Next direction

Maximal rotation angle $|r|max$ is defined as shown in Figure 7. As another output from the neural network, a value between 0 and 1 is obtained. This value is transformed into the next direction 'r' is determined by equation (2).

$$r=2*|r|max*(OutputValue-0.5) \quad (2)$$

Then, a next direction angle of an agent in the absolute coordinate is calculated by equation (3). In equation (3), 't' is time from the start.

$$\theta(t+1) = \theta(t) + r \quad (3)$$

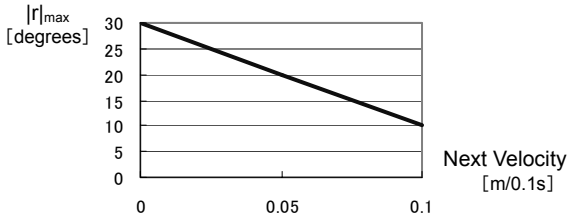


Figure7 Relation between $|r|_{max}$ and the velocity

Figure 8 shows the possible range of changing direction regarding to the traveling direction.

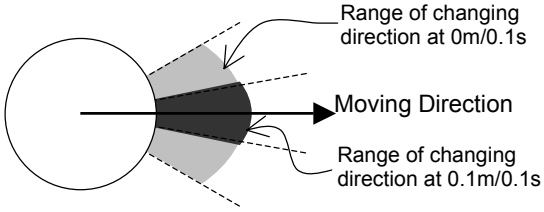


Figure8 The possible range of changing direction

D. Learning method of walking behavior

In this paper, learning of the walking behavior of the agent means training of neural network. As for the training method of neural network, supervised learning method such as error back-propagating method is employed in many cases. However, in this paper, a kind of reinforcement learning method without training data is introduced by using PfGAs.

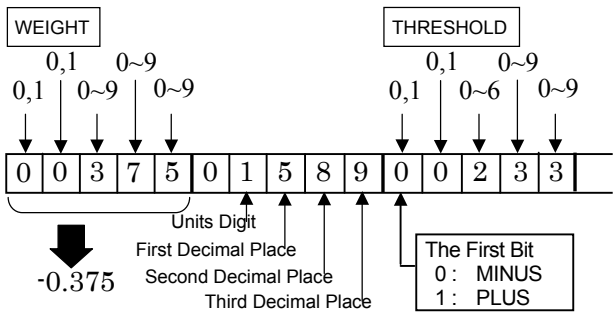


Figure9 Converting genotype to phenotype

Chromosomes of PfGAs consist of all weights and threshold values of the neural network. Every 5 bits of a

chromosome corresponds to a value of a weight or a threshold value as shown in Figure 9. Decoding method of a weight or a threshold value also shown in Figure 9.

To perform GA operations, it is necessary to assume proper evaluation functions. In this paper, to learn the walking rules how to arrive the destination as fast as possible and how to avoid collision with the other agents as much as it can, the below evaluation value is configured.

The shortest time '*Ts'* from the start to the destination is obtained by division of the maximal speed '*Vmax*' to its straight distance '*Dist*'. In this paper, an allowed time '*Ta'* to the start to the destination is assumed to be 2.5 times *Ts*, *Ta=2.5*Ts=2.5*Dist/Vmax*. In case that all agents can be reached to the destination within *Ta*, evaluation value '*e*' is estimated by equation (4) and otherwise by equation (5).

$$e = t + c*5 \quad (4)$$

$$e = a + d'/Vmax + c*5 \quad (5)$$

In equation (4), the time '*t*' from the start to the destination and the numbers of collision '*c*' are considered. To average each evaluation value, '*c*' is divided by 5. In equation (5), the time required to move remaining distance '*d'*' with maximal speed '*Vmax*' is added to the evaluation. This time is summed up regarding to agents not reached to the destination. In Case1 mentioned in next chapter, that operation repeats 5 times changing initial direction of the agent, and evaluation value '*e*' is summed up.

In this paper, a chromosome with minimal evaluation values is assumed to be best one and weights and threshold values are saved as optimal walking rules of the agent.

IV. Implementation

A. Learning

Agents are placed in the assumed space, and learn their walking rules. In Case1, one agent model faces several another models which have different rules to move linearly only from the left side to the right. In Case2, agents face each other 5 to 5, and the distances from the start to the destination of each agent are equal. It is assumed that an agent learning in Case1 is called AGENT1, and an agent in Case2 is called AGENT2.

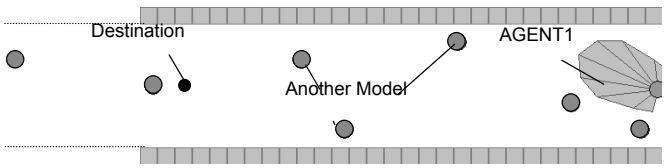


Figure10 Case1: Training by using one agent and another models

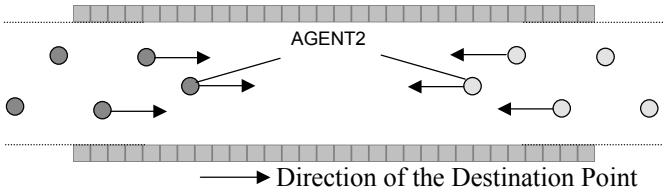
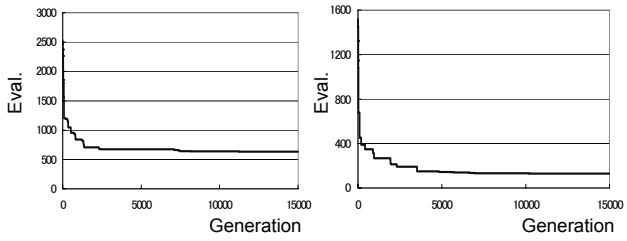


Figure11 Case2: Training by using 10 agents (5 to 5)

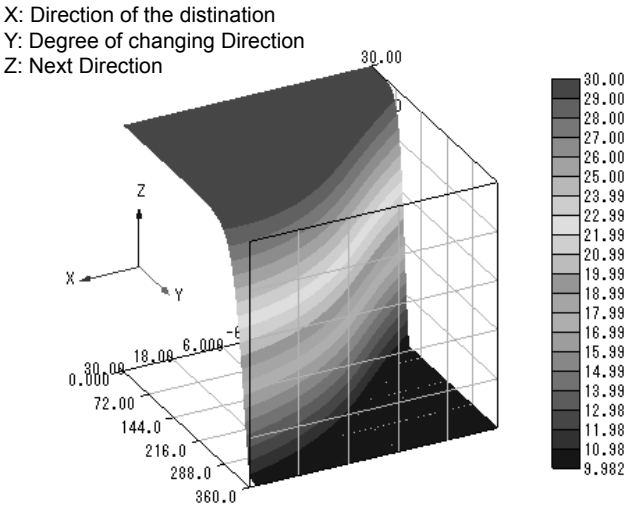
B. Result

Figure12 shows the convergence of the evaluation value of AGENT1 and AGENT2. Figure13 shows the output distribution charts of the neural networks of AGENT1 and AGENT2. 0 is input for "Each sensor's judgment" and 1 is input for "Current velocity". It is clarified that "Next direction" is calculated according to "Degree of changing direction" and "Direction of the destination".

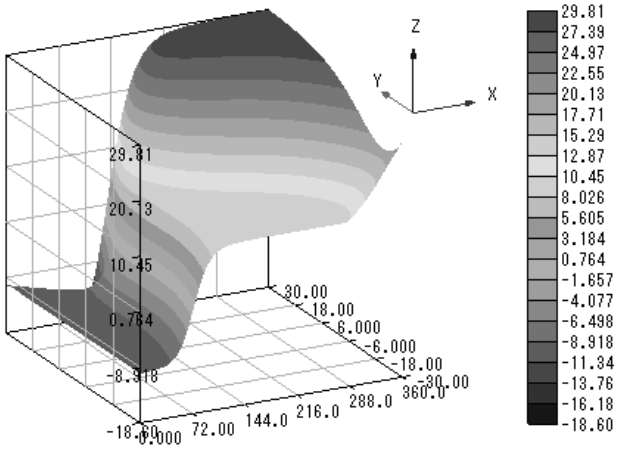


(a) AGENT1 (b) AGENT2

Figure12 Transition of the evaluation value



(a) AGENT1



(b) AGENT2

Figure13 Output distribution charts of neural networks

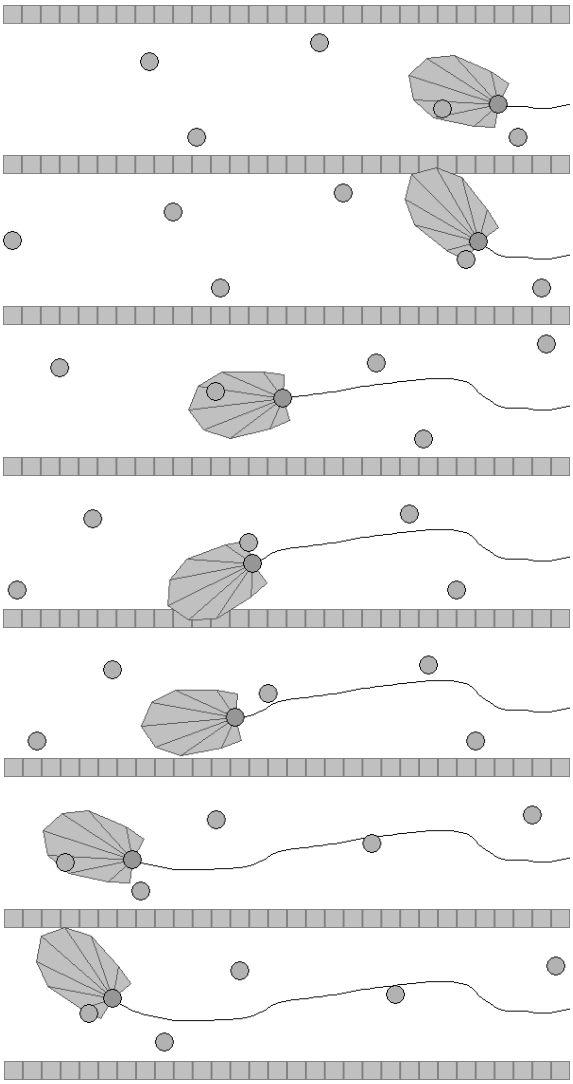


Figure14 The movement trajectory of AGENT1 (with another models)

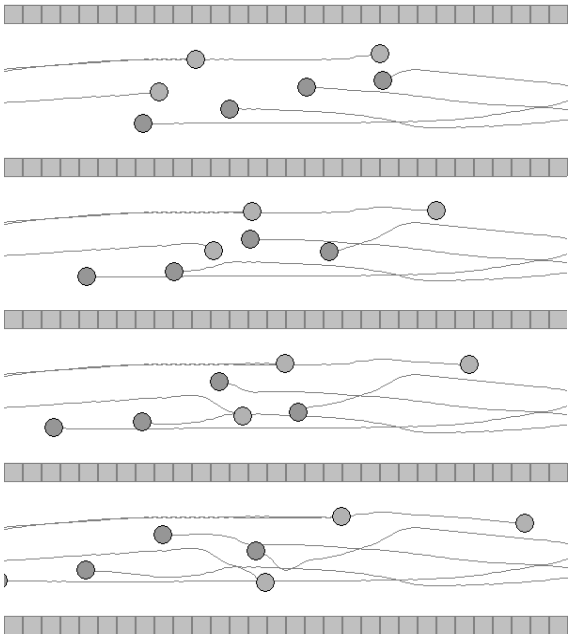


Figure15 The movement trajectory of multiple AGENT1s

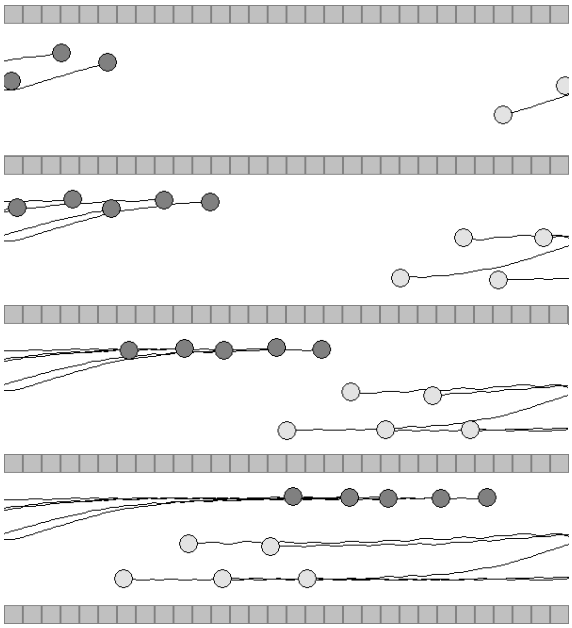


Figure16 The movement trajectory of multiple AGENT2s

Figures from 14 to 16 show the movement trajectory of AGENT1 and AGENT2 already learned their walking rules. In Figure14, AGENT1 is moving in the space under the same condition in learning it's walking rules. To distinguish between AGENT1 and another models, the sensor of AGENT1 is only shown. In Figure15, several AGENT1s are moving concurrently from both sides. In Figure16, AGENT2s are also moving from both sides.

V. DISCUSSION

Figure 12 shows that agents are able to learn accurately.

AGENT2 learned more complicated walking rules as for moving direction for the direction of the destination without close-by obstacles as shown in Figure13.

Figures 14 and 15 show that AGENT1 can move to the destination, avoiding another models or the other AGENT1s. It tries to move as straight as possible to the destination.

Figure16 shows that AGENT2 learned the walking rule to keep to the left side. It is assumed that this is an efficient rule as a whole. Rules to follow to one who walks ahead are also obtained.

VI. CONCLUSION

This study made an attempt about agents' avoidance to collision by using multiple agents with the discrimination sensor.

The results show that an agent is able to learn moving rules including avoidance to collision with dynamic obstacles by neural network and PfGAs.

By the learning of one agent, the rule that agents walk straight to destinations and avoid close-by obstacles is obtained. Agents tend to keep to the right or left side with

walking rules learned by using multiple agents. These results show a possibility that an agent gets well-adapted moving rules in other spaces.

To advance performances of an agent and to compare the results of simulation using agents proposed here with observation results of actual human walking behaviors are collaterally assumed as future issues.

ACKNOWLEDGMENT

This study was supported by the Japan Society for the Promotion of Science (2004 Grants-in-Aid for Scientific Research, Scientific Research (A)(1), No.14205087: System Design for Architecture, Urban and Social Systems Adapting to Human and Environment by Complexity Science), and supported in part by the Twenty-First Century Center of Excellence (COE) Program "Design Strategy towards Safety and Symbiosis of Urban Space" awarded to Graduate School of Science and Technology, Kobe University. The Ministry of Education, Culture, Sports, Science and Technology of Japan sponsors the Program.

REFERENCES

[1]Morishita S., Yamamoto H., Ohtaka Y., Nakano T., “Simulation of Purchase in a Store by Cellular Automata ”, Trans. of the Japan Society for Computational Engineering and Science., 149 – 154, 1999 (in Japanese)

[2]Sano T., Takayanagi H., Watanabe H., “The Evaluation of Congestion in Pedestrian Space with Spase-Time Diagram Model”, J. Archit. Plann. Environ. Eng., AIJ, No. 555, 191 – 197, 2002. (in Japanese)

[3]Oda M., Takizawa A., Kawamura H., Tani A., “Construction of Simulator of Human Behavior in Continuously Defined Space and Application for Planning of Building Space Using Agent Model”, J. Archit. Plann. Eng., AIJ, No. 558, 315 – 332, 2002. (in Japanese)

[4]Katsuyama M., Takizawa A., Kawamura H., Tani A. “Human Model by Agent System –Behavior of Appreciation in a Museum-”, AIJ. Kinki Brunch, 305 – 308, 2003. (in Japanese)

[5]Shimada Y., Takizawa A., Kawamura H., Tani A. “Pedestrian Walking Model with Discrimination Sensor Trained by Multilayered Neural Network –In Consideration of Moving Velocity of Agent-”, AIJ. Kinki Brunch, 309 – 312, 2004. (in Japanese)

[6]A. J. F. van Rooij, L. C. Jain, R. P. Johnson, *Neural Network Training Using Genetic Algorithms,* World Scientific Publishing, 1 – 38, 56 – 74, 1996.

[7]Kitano H., *Genetic Algorithm 4,* Sangyo Tosyo Co., Ltd., 77 – 111, 2000. (in Japanese)