Paper:

An RT Component for Simulating People Movement in Public Space and its Application to Robot Motion Planner Development

Atsushi Shigemura, Yuki Ishikawa, Jun Miura, and Junji Satake

Department of Computer Science and Engineering, Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan
E-mail: shigemura@aisl.cs.tut.ac.jp
[Received April 28, 2011; accepted August 24, 2011]

This paper describes a software module for simulating "people movement" in public space such as shopping centers and cafeterias. We decompose people movement into global and local, and make a model of each of them. Global movement corresponds to following a route from a current position to a destination. In local movement, a person moves toward the next subgoal while avoiding surrounding persons and obstacles. We also model behavior specific to a cafeteria, such as queuing and searching for unoccupied seats. We implement these simulation algorithms in a simulator RT component, that can be used easily for development of robot motion planners, which are also realized as RT components. Various simulation experiments show the effectiveness of the simulation algorithms and the simulator RT component.

Keywords: people movement simulator, people behavior modeling, path planner, RT component

1. Introduction

There is an increasing demand for robots that can help persons in daily life. Such personal service robots are one of the promising areas in which robotic technologies can be applied.

One possible task of service robots is to move safely while guiding persons or carrying items in public space such as shopping centers and cafeterias. Realizing the safe movement of a robot is, however, a difficult mission, especially when the robot moves among many persons. Such a difficulty comes, for example, from the low reliability of environmental sensing. Since the motion planning function of the robot relies heavily on the quality of environment recognition, it is still difficult to test a robot system in real environments.

We therefore developed a "people movement" simulator that models and simulates person's movement in a public space. Using the simulator, we can easily and safely test our planning algorithms in various simulated environments.

Crowd simulation is an approach to analyzing the collective behavior of multi-agent systems. Raynolds [1] proposed a model of coordinated animal motions such as that of bird flocks, which has three relatively simple rules for coordination: separation, alignment, and cohesion. He used this model to create computer animation of a flock of birds (boids). Helbing et al. [2] developed pedestrian models for normal situations and for evacuation in panic situations based on a generalized force model of pedestrian dynamics.

Akuzawa and Taguchi [3] developed a simulator of people movement in the concourse of a station. They decompose people movement into global and local movement and simulated movement by combining a potential method and spatial network-based path planning. We take a similar hierarchical approach but use a different route search method and more elaborate expressions for representing local people movement.

Most of the previous work on people movement simulation is for simulating the flow of people in a public space. For this purpose, the researchers model people movement toward a destination. In addition to modeling such goal-directed movement, we model behavior specific to public space, such as queuing and searching for unoccupied seats.

A simulator has been implemented as an RT component, which is a software module running in an RT-middleware environment [4].[1] Since our motion planning module has also been realized as an RT component, the motion planning module can easily be connected to the simulator for development, and then used without any modification for controlling a real robot.

This paper is organized as follows: Section 2 describes a model of local movement based on a potential method. Section 3 describes a model of global movement based on a spatial network. Section 4 describes simulation results, simulation implementation as an RT component, and applications to path planner development. Section 5 concludes this paper and discusses future work.

2. Modeling of Local Movement

Local movement is defined here as the action of a person moving toward a goal position while avoiding colli-

1. RT-middleware is a specification on a component model and infrastructure services applicable to the domain of robotics software development, authorized by the OMG (Object Management Group).

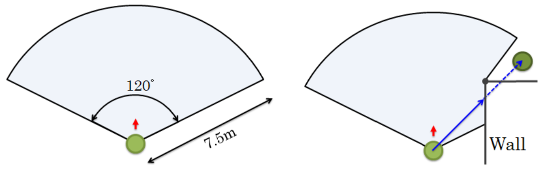


Fig. 1. Field of view.

sion with surrounding persons and obstacles. A person's local movement is modeled by using a potential method. A person is defined as a circle. A wall is defined as a segment that connects two points.

A person has a limited field of view. Hall [5] classified social distance into four types; public, social, personal, and intimate. In this paper, a person's view is limited to a sector with a 7.5 m radius, which corresponds to close phase of public distance, and 120° in central angle (at left in **Fig. 1**). Individuals generally need a viewing angle of at least 110° in order to understand their surroundings, and the viewing angle of persons walking is 110°–160°. In this paper, we set the angle by referring to Akuzawa's research [3].

Each person recognizes other persons and objects that come within their field of view as long as they are not occluded by other people and objects. When a wall exists that intersects with a line connecting two persons, we judge that the two persons cannot see each other (at right in **Fig. 1**).

2.1. A Person's Walking Speed

People do not walk at a fixed velocity. Their walking speed changes depending on their surroundings. People slow down to avoid collision in a crowd. We assume that only those persons in front of a person and within short distance influence the person. The influence region is defined as a sector with a 2 m radius and a central angle of 60°. This distance is kept between persons when they are walking at normal speed in a crowd, and this angle is based on the largest field of view, i.e., 60°–70°, which a person can see other persons comfortably [6]. We calculate people density in a region and use it to control a walking person's speed.

Several research studies have discussed the relationship between walking speed and congestion (people density). Walking speed V of a person is determined by Eq. (1):

$$V = a - b \cdot K, \quad \quad (1)$$

where a is the walking speed when influence from the surroundings is not imposed and b is a constant that shows the relationship between congestion K and the walking speed. Various values of a and b have been proposed in existing research, which are shown in **Table 1**. It is desirable to use these values based on the particular situations under consideration. We, however, use a simpler model given by

$$V = V_s - b \cdot K, \quad \quad (2)$$

Table 1. Relation between walking speed and congestion [7].

Researcher	Situation	a	b
Yoshioka	Commuting	1.61	0.33
Yoshioka	Event	1.349	0.376
Yoshioka	Shopping	1.13	0.28
Mouri and Tukaguchi	Commuting	1.48	0.204
Takeuchi	In residential area	1.50	0.38
Fruin	Commuting	1.356	0.341
Older	Shopping	1.311	0.394
Oeding	Composite	1.50	0.394
Navin and Wheeler	In university	1.63	0.60

where V_s is the walking speed when no influence from the surrounding is imposed and the walking speed does not change in any situations.

2.2. Determining a Person's Direction of Movement by Potential Function

In this simulator, the direction of movement of a specific person is decided by using the potential method as follows:

- A positive charge is given to persons, walls, and other objects.
- A negative charge is given to the destination.
- Attractive and repulsive forces between charges are calculated.
- The direction of movement is determined to be the composition of attractive and repulsive forces.

Repulsive force F_{wall} from the wall is defined by

$$F_{wall} = -\frac{Q_{wall} \cdot e_{wall}}{|e_{wall}|^4}, \quad \quad (3)$$

where e_{wall} is the vector to the nearest point on the wall in the environment and Q_{wall} is a constant (the amount of the charge) that determines the influence of the wall.

Repulsive force from other persons F_{person} is defined by

$$F_{person} = -Q_{person} \cdot \sum_{i=1}^{n_L} \frac{e_{person_i} \cdot \cos(\theta_i)}{|e_{person_i}|^2}, \quad \quad (4)$$

where n_L is the number of persons in the field of view, e_{person_i} is the vector to person i, Q_{person} is a constant that determines the influence of a person, and θ_i is the relative angle between the direction of movement of the subject and person i. **Fig. 2** shows vectors to the wall and person.

Similarly, F_{robot} from robots is given by

$$F_{robot} = -Q_{robot} \cdot \sum_{i=1}^{n_R} \frac{e_{robot_i}}{|e_{robot_i}|^2}, \quad \quad (5)$$

where n_R is the number of robots in the subject's field of view.

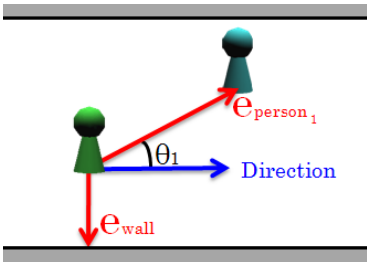


Fig. 2. Vector to wall and person.

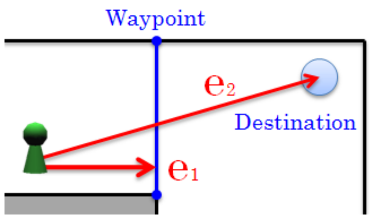


Fig. 3. Vector to destination.

Next, we consider the attractive force of the destination. The attraction from a position is given by

$$F_i = \frac{e_i}{|e_i|} \quad \ldots \ldots \ldots \ldots \quad (6)$$

The attraction is normalized so that a certain amount of the force can be applied even for a distant destination (or waypoint).

People often cannot take a straight path toward the destination due to obstacles on the way. In such a case, they select a path passing several waypoints.[2] To simulate smooth movement, we combine two attractive forces, one from the nearest waypoint and the other from the second nearest one, as shown in **Fig. 3**. A waypoint can be the destination when a person is near to the final destination.

In **Fig. 3**, e_1 is the vector to the next waypoint and e_2 is the vector to the destination. If there are no waypoints, e_1 becomes the vector to the destination and e_2 is set to 0.

The combined attractive forces $F_{forward}$ from the forces, F_1 and F_2, from two waypoints is calculated by:

$$F_{forward} = \frac{F_1 + \frac{1}{2} \cdot F_2}{\left| F_1 + \frac{1}{2} \cdot F_2 \right|} \quad \ldots \ldots \ldots \quad (7)$$

This expression does not have any constant because relative importance factors of the other forces are defined relative to this attractive force.

The direction of resultant force F_{all} decided by surrounding persons and robots, walls, and the destination indicates that the person's direction of movement is given by

$$F_{all} = F_{forward} + F_{wall} + F_{person} + F_{robot} \quad . \quad (8)$$

2. Waypoints are defined in Section 3.1.

2.3. Calculating a Person's Position at Each Time Step

Position p_{next} after movement is calculated from current position p, the direction of F_{all} obtained by Eq. (8), the distance of movement given by velocity V (see Eq. (2)), and time step Δt.

To avoid large acceleration/deceleration of a person, we use as the current velocity the average of the velocity calculated from the current and the previous situation given by:

$$v = \frac{1}{2} \cdot \left(V \cdot \frac{F_{all}}{|F_{all}|} + v_p \right) \quad \ldots \ldots \ldots \quad (9)$$

The position after current movement is then given by

$$p_{next} = p + v \cdot \Delta t \quad \ldots \ldots \ldots \ldots \quad (10)$$

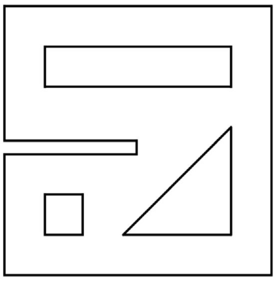
3. Modeling of Global Movement

People movement simulation in a simple environment such as a straight corridor can be done only by using the local movement method. People movement in a complex environment where multiple pathways exist, however, cannot be simulated by local movement alone. This section explains a route search method that is necessary in such cases. A cafeteria is given as an example of public space. Destination selection behavior specific to the environment is also described.

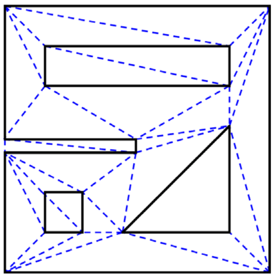
3.1. Building a Spatial Network

To calculate a person's route, we divide the free space into subareas and a spatial network with the subareas being nodes is constructed. We can calculate the route from one subarea to another by using this spatial network. A subarea is defined as a convex polygon. We divide the environment into convex polygons using the method suggested by Lamarch et al. [8]. **Fig. 4** shows the process of space division. First, for the map of the environment (**Fig. 4(a)**), constrained Delaunay triangulation of the segments representing walls is computed (**Fig. 4(b)**). Next, bottlenecks that characterize the minimal distance between corners and walls are computed (**Fig. 4(c)**), and triangulation is locally recomputed in order to take these modifications into account (**Fig. 4(d)**). In order to simplify this subdivision and to minimize the number of cells, triangles are merged into convex polygons while locally conserving bottleneck information (**Fig. 4(e)**).

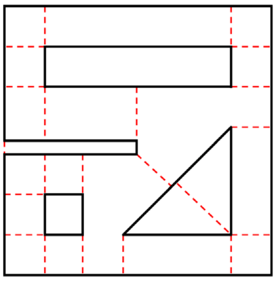
Borderlines of a subarea are either walls or those with adjacent subareas. For each person, we calculate the route from the current subarea to the subarea of destination, and move the person to the destination by tracing subareas on the route. Waypoints are set on borderlines between subareas. When a person moves to a borderline, the nearest point on the borderline to the current position is selected as the waypoint to approach.



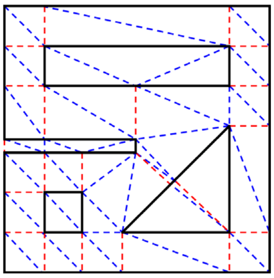
(a) 2D map of environment



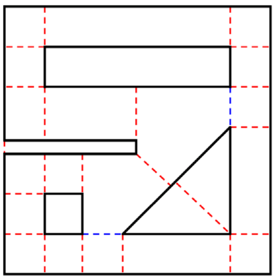
(b) Original constrained Delaunay triangulation



(c) Computed shortest distances between corners and walls



(d) Constrained Delaunay triangulation with shortest distances



(e) Merged convex polygons

Fig. 4. Division of environment.

3.2. Route Finding Considering Congestion

A route is calculated by using the A* algorithm. We use distance and congestion (people density) for determining the cost between nodes. As a result, we do select the minimum time route rather than the shortest route.

Speed V_a of persons in a subarea is calculated by using Eq. (1). The density in the subarea is used for density K. The time required for passing through the subarea is calculated by

$$Cost_a = \frac{d_a}{V_a}, \qquad (11)$$

where d_a is the travel distance inside the subarea.

The distance between subareas is given by (see **Fig. 5**):

- **Current subarea:**
  Distance from the current position to the center of the border to the next subarea.

- **Subarea on the way:**
  Distance between centers of borders.

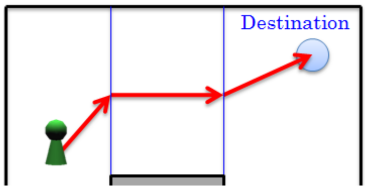


Fig. 5. Distance for route cost calculation.

- **Subarea in destination:**
  Distance from the center of the border with the previous subarea to the destination.

The total cost is the summation of the time required for passing the subareas on the route, given by:

$$Cost = \sum_{a=1}^{n_A} Cost_a, \qquad (12)$$

where n_A indicates the number of subareas on the route. The minimum time route is selected as the best route.

3.3. Classification of Destinations and People Movement Modeling at Individual Destinations

This paper deals with destination-directed people movement. People behave differently depending on the destination to move to. In this simulator, the destination is classified into the following four categories: Exit, Seat, Queue, and Point.

Exit: This is a destination for going outside of the simulation environment. It is defined as a segment, and a person moves toward the nearest point on the segment. The person who reaches the exit disappears from the simulation environment.

Seat: This is a place occupied by a person, e.g., a bench in a passage or a chair in a cafeteria. People cannot select a seat as a destination if someone is already sitting there.

Queue: This is where people form a queue, e.g., to a ticket vending machine or a ticket gate. This is defined by the position of the person at the head of the queue and the direction of queuing. A person who wants to enter the queue moves towards the end of it.

Point: This is the place a person chooses as a destination but is not classified as other categories. This is defined by a circle (the position of the center and the radius), and a person is judged to have reached the destination when entering the circle.

3.4. Selection of a Destination

A person may sometimes select one destination from multiple ones of the same kind, such as the case where a person selects one of multiple exits or one of multiple ticket vending machines. To simulate such a behavior, we calculate the evaluation value of each destination and have a person select the optimum one. Evaluation function $E(d)$ is defined for each destination d.

The evaluation function for an exit is defined as:

$$E_{Exit}(d) = Cost(d) \cdot N(1.0, 0.2^2), \quad \ldots \quad (13)$$

where $Cost(d)$ is the route cost to move to destination d, i.e. the time necessary for movement, and calculated by Eq. (12). The cost function is multiplied by a normal distribution to introduce.

The evaluation function for a point is defined as:

$$E_{Point}(d) = Cost(d) \cdot N(1.0, 0.2^2) \quad \ldots \quad (14)$$

This is the same as the expression for exits.

The evaluation function for a seat is defined as:

$$E_{Seat}(d) = (Cost(d) \cdot (1.0 + \rho)) \cdot N(1.0, 0.2^2), \quad (15)$$

where ρ is the density of persons around the seat, given by the density of the subarea including the seat. We multiply the cost function by $1.0 + \rho$ to simulate how persons work to avoid crowded areas. This eventually avoids the situation in which all persons select the seats nearest.

The evaluation function of a queue is defined as:

$$E_{Queue}(d) = (Cost(d) + N_q \cdot t_q) \cdot N(1.0, 0.2^2), \quad (16)$$

where N_q is the number of people in the queue and t_q is the time for receiving service at the head of the queue per person. This expression estimates the time until the person leaves the queue, calculated as the sum of the time for reaching the queue, that for queuing, and that for receiving service. Each person selects the queue for which service will finish soonest.

In all of the above evaluations, the smaller the value, the better.

3.5. People Action Sequence

Human behavior does not consist of a single action but of a sequence of actions with each persons executing a sequence of movements from the entrance to the simulated environment, and then from it to the exit. A person may, for example, take the following sequence: queues up to the vending machine to buy a meal ticket, receives a dish at the counter for the ticker, moves to an occupied seat, and starts eating. The action of not moving, such as eating, is also modeled and described for simulation. Each person is given a sequence of actions and executes one action after another until all actions are executed.

4. Simulator RT Component

We developed a simulator based on the above people movement models, and performed simulation experiments. The subsections that follow describe people movement simulation results, simulator implementation as an RT component, and applications to path planner development.

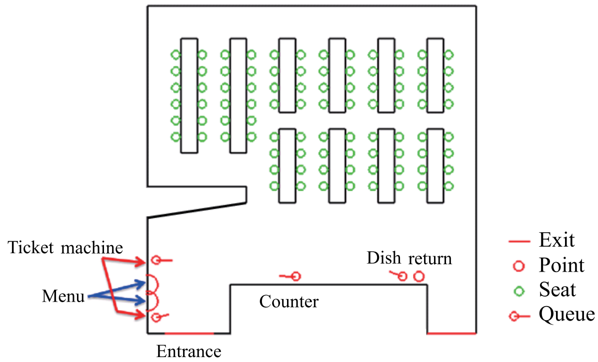


Fig. 6. Modeled environment.

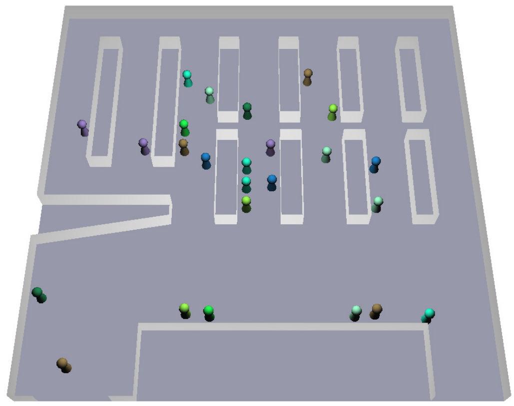


Fig. 7. Simulation.

4.1. People Movement Simulation

We have developed a simulator that models people movement in the cafeteria at our university, as shown in **Fig. 6**. **Fig. 7** shows a simulated scene. Cones with balls on top denote people. We can simulate behavior that checks the menu, queues for a ticket vending machine, receives a dish at the counter, eats while seated, returns tableware, and leaves via the exit. We use OpenGL for graphics simulation. **Fig. 8** shows a scene simulated for queuing and **Fig. 9** shows a person going between persons.

When the number of people in the environment is about 30, the execution time of one cycle of simulation is below one millisecond. Since many people are stopping to sit or queue in our environment, i.e., a cafeteria, and motion planning is not necessary for them, we can simulate more persons simultaneously in real time.

In our subjective evaluation, the developed algorithm is considered to simulate behavior in the cafeteria well. It is necessary, however, to quantitatively evaluate results in comparison with actual people movement data.

4.2. Simulator RT Component Implementation

One important issue in the development of software modules using RT-Middleware is the selection of interfaces, which determine what information is transferred in what format. The main function of our simulator RT

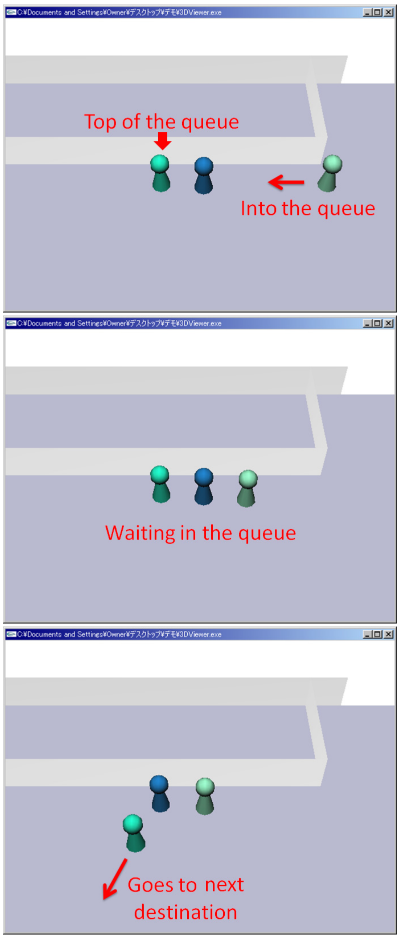


Fig. 8. Simulation scenes – queuing.

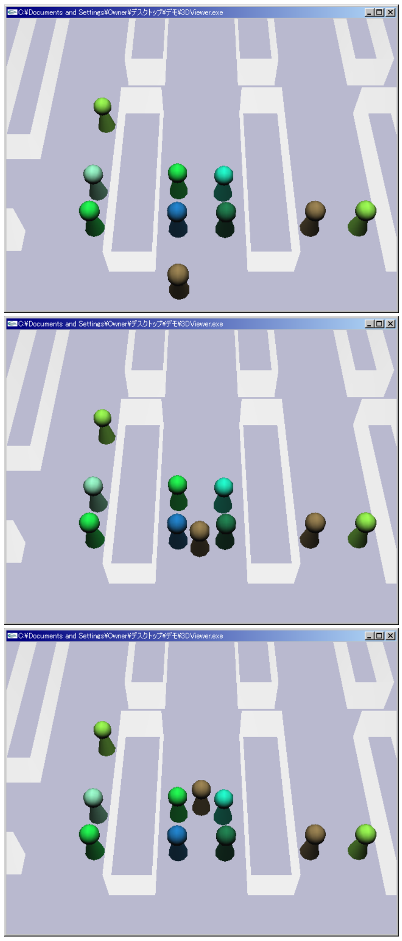


Fig. 9. Simulation scenes – going between persons.

Component (RTC) is to provide a sufficient information channel for mobile robot motion planning and control. We therefore define the following interfaces:

- Input: Robot control command.
- Output: Robot pose and velocity, people position, global map in the world coordinates, and local map in the robot coordinates.

The simulator RTC has the following parameter set given in a configuration file:

- Name of the map data file. The file has information on walls, entrances including the rate at which persons show up, destination, and behavioral patterns of persons specified by a sequence of actions and their transition timing.
- Specifications of simulated persons such as the walk rate and the target for a person-following task.
- Robot specifications such as tread, maximum acceleration, and maximum deceleration. These parameters must be identical to those of the actual robot to be used.
- Robot initial pose (x, y, θ). We can start the simulation of robot movement anywhere in the simulated environment.

4.3. Application to Path Planner Development

Testing a path planner in various environments is important for fast and reliable planner development. It is, however, difficult to test the planner in an actual environment because environment recognition is not always reliable. We thus connect the path planner to the simulator for tests in various situations.

We have been developing a randomized path planner for a mobile robot and implemented it as an RT-component [9]. By also implementing the simulator as an RT-component, we can connect these components directly as shown in **Fig. 10**. This configuration is for testing and refining a local path planner. After development, the path planner component can be applied directly to an actual robot.

Figure 11 shows the results of path planning. The robot is instructed to follow a person. In this figure, circles and triangles indicate stopping and moving people, respectively. Lines passing the robot indicate the robot trajectory and a randomized path search result.

It is also possible to construct a more complex simulation system using this simulator. **Fig. 12(a)** shows a system for an autonomous robot maneuver, including both global-level route planning and local-level motion planning, constructed by using RT components. **Fig. 12(b)**

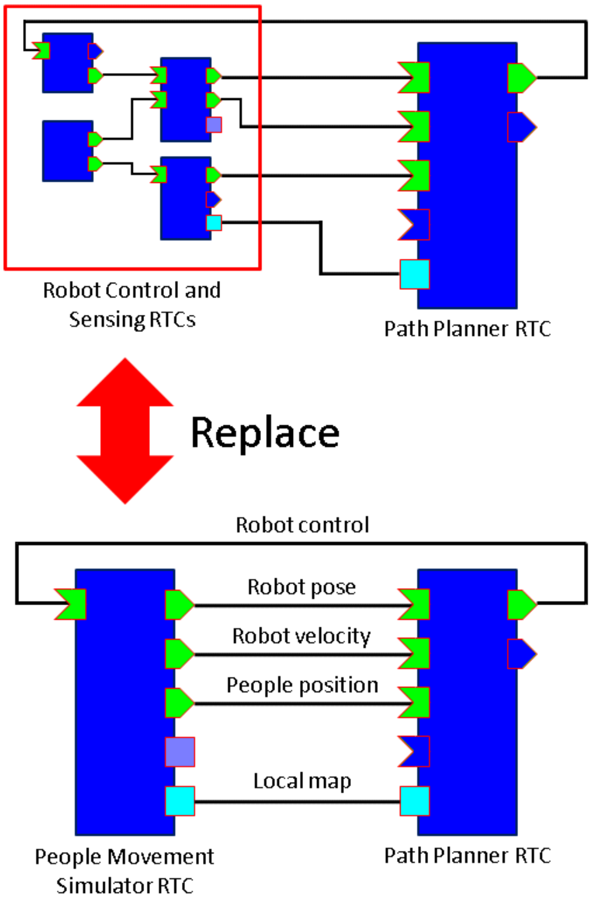


Fig. 10. Path planner development environment constructed by using RT components.

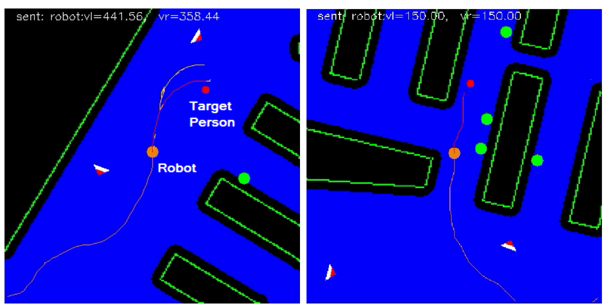
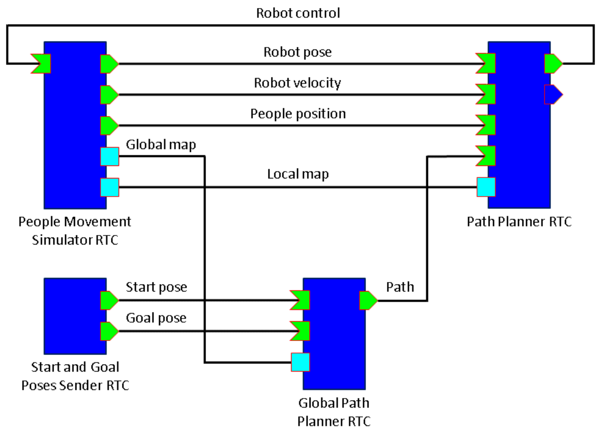


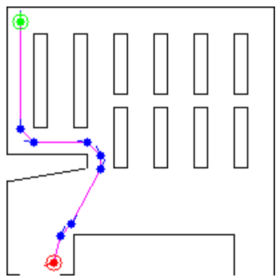
Fig. 11. Simulation with path planning algorithm.

shows route planning and **Fig. 12(c)** shows motion planning. The robot moves to a destination by tracing waypoints generated by the global path planner RTC.

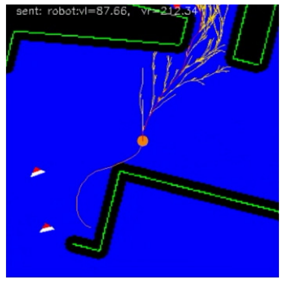
We can also simulate people movement using map data generated by sensing an actual environment. **Fig. 13(a)** shows a map generated by our SLAM algorithm [10]. On the map, free space where a robot can move is shown in black and obstacle regions are shown in white. **Fig. 13(b)** shows the result of space division into subareas, with boundaries between subareas represented as thin lines. **Fig. 13(c)** shows a snapshot of people movement simulation and robot motion planning for person-following behavior in this environment.



(a) Robot autonomous maneuver system



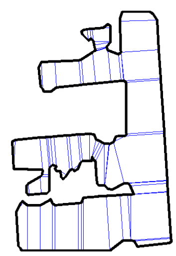
(b) Generated waypoints



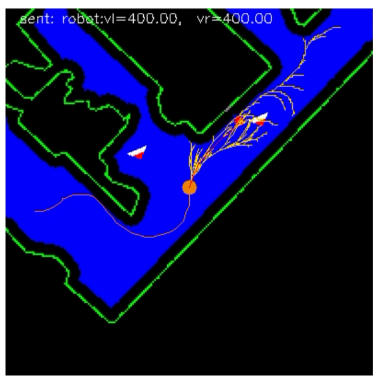
(c) Robot motion planning

Fig. 12. Global planning using the simulator RTC.

(a) Original map data



(b) Vectorized environment



(c) Robot motion planning

Fig. 13. People movement simulation and robot motion planning on a map generated by a SLAM algorithm.

As the above results show, it is easy to test robot motion planning in various environments by using the Simulator RTC.

5. Conclusions

This paper has described a simulator for people movement in public space and its application to robot path planner development. The simulator models not only the simple goal-directed movement of people but also behavior in specific scenes in a cafeteria. The experimental results show that the people's behavior in cafeteria is well simulated. The simulator has been implemented as an RT component and can easily be applied to testing path planner RT components in various environments.

Several directions are possible in future work. One is to apply the simulator to the development of more advanced path planning algorithms, such as the one that considers people's flow and/or behavior in specific situations. Other future work is to improve the simulation model so that it can simulate more natural human behavior such as smooth passing by in a narrow space and movement of avoiding queues. It is also necessary to evaluate how realistic people movement simulation is. A quantitative comparison of statistics on people movement in an actual and a simulated environment would lead to a possible evaluation.

Acknowledgements

This work was supported by the NEDO (New Energy and Industrial Technology Development Organization, Japan) Intelligent RT Software Project.

References:

[1] C. W. Raynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," Computer Graphics, Vol.21, No.4, pp. 25-34, 1987.

[2] D. Helbing, I. J. Harkas, P. Molnar, and T. Vicsek, "Simulation of Pedestrian Crowds, in Normal and Evacuation Situations," Proc. of Pedestrian and Evacuation Dynamics 2000, pp. 21-58, 2000.

[3] A. Akuzawa and A. Taguchi, "A Study of Simulation Model for Pedestrian Movement in a Station Yard," Abstracts of 2005 Autumn Forum of the Operations Research Society of Japan, pp. 192-193, 2005. (in Japanese)

[4] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "RT-middleware: Distributed component middleware for RT (robot technology)," 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3555-3560, 2005.

[5] E. T. Hall, "The hidden dimension," Doubleday & Company, 1966.

[6] J. J. Fruin, "PEDESTRIAN – Planning and Design –," Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[7] K. Okada, T. Wada, and Y. Takahashi, "Walking model and people flow simulation based on person's behavior," Abstracts of 2003 Spring Forum of the Operations Research Society of Japan, pp. 102-103, 2003. (in Japanese)

[8] F. Lamarche and S. Donikian, "Crowds of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environments," Computer Graphics Forum, Proc. of Eurographics 2004, Vol.3, No.23, pp. 509-518, 2004.

[9] I. Ardiyanto and J. Miura, "Heuristically Arrival Time Field-Biased (HeAT) Random Tree: An Online Path Planning Algorithm for Mobile Robot Consdering Kinodynamic Constraints," Proc. of 2011 IEEE Int. Conf. on Robotics and Biomimetics, pp. 360-365, 2011.

[10] K. Kitajima, J. Miura, and J. Satake, "Map generation and localization for mobile robot using appearance and range information," JSME Robotics and Mechatronics Conderence 2011, 2011. (in Japanese)

Name:
Atsushi Shigemura

Affiliation:
Department of Computer Science and Engineering, Toyohashi University of Technology

Address:
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)

Name:
Yuki Ishikawa

Affiliation:
Department of Computer Science and Engineering, Toyohashi University of Technology

Address:
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan

Brief Biographical History:
2011 M.Eng. in computer science at Toyohashi University of Technology

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)

Name:
Jun Miura

Affiliation:
Professor, Department of Computer Science and Engineering, Toyohashi University of Technology

Address:
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan

Brief Biographical History:
1989 Dr.Eng. in Information Engineering, the University of Tokyo
1989-2007 Department of Mechanical Engineering, Osaka University
1994-1995 Visiting Scientist, Computer Science Department, Carnegie Mellon University
2007- Professor, Department of Computer Science and Engineering, Toyohashi University of Technology

Main Works:
• intelligent robotics, computer vision, and artificial intelligence

Membership in Academic Societies:
• Institute of Electric and Electronic Engineers (IEEE)
• Association for the Advancement of Artificial Intelligence (AAAI)
• Robotics Society of Japan (RSJ)
• Japanese Society of Artificial Intelligence (AIA)
• Information Processing Society of Japan (IPSJ)
• Institute of Electronics, Information, and Communication Engineers (IEICE)
• Japanese Society of Mechanical Engineers (JSME)
• Institute of Systems, Control, and Information Engineers (ISCIE)

Name:
Junji Satake

Affiliation:
Assistant Professor, Department of Computer Science and Engineering, Toyohashi University of Technology

Address:
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi 441-8580, Japan

Brief Biographical History:
2004-2008 The National Institute of Information and Communications Technology (NICT)
2008- Toyohashi University of Technology

Main Works:
• "Stereo-Based Multi-Person Tracking using Overlapping Silhouette Templates," Int. Conf. on Pattern Recognition, pp. 4304-4307, Aug. 2010.
• "Multiple Human Tracking on Image Sequence under Hierarchical Attention Control," Systems and Computers in Japan, Vol.37, No.13, pp. 78-88, Nov. 2006.

Membership in Academic Societies:
• The Robotics Society of Japan (RSJ)
• Information Processing Society of Japan (IPSJ)
• Institute of Electronics, Information, and Communication Engineers (IEICE)