PAPER

SHOT: Scenario-Type Hypothesis Object Tracking with Indoor Sensor Networks**

Masakazu MURATA[†*a)], ***Nonmember***, **Yoshiaki TANIGUCHI**[††b)], **Go HASEGAWA**[††c)], ***and*** **Hirotaka NAKANO**[††d)], ***Members***

SUMMARY In the present paper, we propose an object tracking method called scenario-type hypothesis object tracking. In the proposed method, an indoor monitoring region is divided into multiple closed micro-cells using sensor nodes that can detect objects and their moving directions. Sensor information is accumulated in a tracking server through wireless multihop networks, and object tracking is performed at the tracking server. In order to estimate the trajectory of objects from sensor information, we introduce a novel concept of the virtual world, which consists of virtual micro-cells and virtual objects. Virtual objects are generated, transferred, and deleted in virtual micro-cells according to sensor information. In order to handle specific movements of objects in micro-cells, such as slowdown of passing objects in a narrow passageway, we also consider the generation of virtual objects according to interactions among virtual objects. In addition, virtual objects are generated when the tracking server estimates loss of sensor information in order to decrease the number of object tracking failures. Through simulations, we confirm that the ratio of successful tracking is improved by up to 29% by considering interactions among virtual objects. Furthermore, the tracking performance is improved up to 6% by considering loss of sensor information.

key words: *object tracking, sensor network, region segmentation, virtual object*

1. Introduction

In recent years, wireless sensor network technology has attracted a great deal of attention. Object tracking is a major wireless sensor network application and it can be used in numerous applications, including accident prevention in care facilities, detection of suspicious individuals in buildings, and flow line analyses in commercial facilities [2].

Binary sensors, such as infrared sensors, are among the simplest sensors and can detect only the presence or absence of objects within sensing range. Binary sensors generate only one-bit information. Although binary sensors can neither detect the number of objects nor identify objects individually within sensing range, they have advantages in terms of cost, simplicity, and energy-efficiency. Therefore, a number of object tracking methods using wireless sensor networks composed of binary sensors have been proposed [3]–[6].

However, these methods assume that sensor nodes are deployed uniformly so that the sensing ranges efficiently cover the monitored region. Therefore, a large number of sensor nodes are required to cover a wide monitoring region. In addition, they assume that objects move at constant velocity in the monitoring region. However, in general, the movement characteristics of objects change dynamically depending on interactions among objects and surrounding environments. For example, the speeds of objects may decrease when the objects pass each other in a narrow passageway, and the objects double back at dead end passageways or speed up when riding on moving sidewalks. By handling such specific movement scenarios in object tracking methods, the number of successful instances of tracking can be increased.

In the present paper, we propose an object tracking method called scenario-type hypothesis object tracking (SHOT) for a wide indoor monitoring region, e.g., all floors in a building. In SHOT, sensor nodes are placed only at specific points in the monitoring region, so that the monitoring region is divided into multiple regions called micro-cells, as shown in Fig. 1. Hereinafter, we refer to the location at which a sensor node is placed as a gate. We assume that a sensor node is composed of a pair of infrared sensors with a wireless communication device and can detect transit events of objects along with their moving directions [7], [8]. Sensor

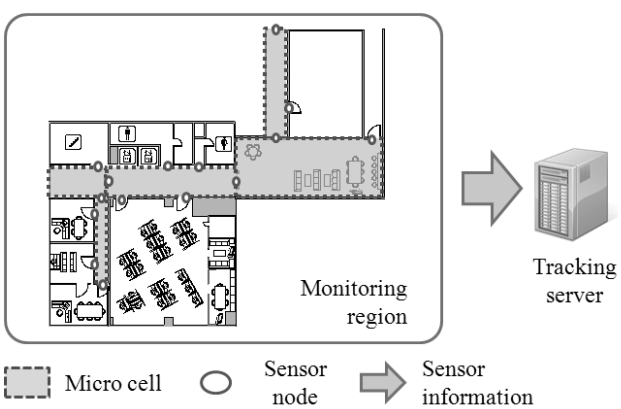


Fig. 1 Object tracking with indoor sensor networks.

Manuscript received August 25, 2010.
Manuscript revised January 15, 2011.

[†]The author is with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.
[††]The authors are with the Cybermedia Center, Osaka University, Toyonaka-shi, 560–0043 Japan.
[*]Presently, with Fujitsu Limited.

[**]This paper is an extended version of the work originally presented at ICNS 2010 [1].

a) E-mail: m-murata@ist.osaka-u.ac.jp
b) E-mail: y-tanigu@cmc.osaka-u.ac.jp
c) E-mail: hasegawa@cmc.osaka-u.ac.jp
d) E-mail: nakano@cmc.osaka-u.ac.jp

DOI: 10.1587/transinf.E94.D.1035

information is collected in a tracking server through wireless multihop networks, and the tracking server estimates object trajectories based on sensor information.

In SHOT, in order to estimate the trajectories of objects, the tracking server maintains a virtual world, which consists of virtual micro-cells, virtual gates, and virtual objects[†]. Virtual objects are generated when the tracking server receives the arrival event information of a real object from outside the monitoring region. The tracking server transfers virtual objects in virtual micro-cells according to sensor information and records their trajectories. In addition, the tracking server duplicates and updates virtual objects according to sensor information, so that the tracking server maintains a number of hypotheses of object trajectories for each real object. Furthermore, the tracking server deletes virtual objects when the possibility that their trajectories correspond to the trajectories of real objects decreases. When the tracking server receives departure event information of a real object departing from the monitoring region, the trajectory of the most appropriate virtual object is selected as the trajectory of the real object.

A number of object tracking methods, such as multi-hypothesis tracking (MHT), which maintain hypotheses of object trajectories, have been proposed [9]–[11]. Whereas MHT-based methods generate hypotheses according to sensor information, SHOT also generates hypotheses, i.e., virtual objects, according to interactions among virtual objects by introducing the virtual world. As a result, SHOT can handle specific movement scenarios of objects in micro-cells by generating virtual objects for all candidate trajectories in virtual micro-cells. In the present paper, we consider the slowdown of objects that pass each other in a narrow passageway as one scenario example in order to demonstrate the effectiveness of introducing the virtual world. In addition, SHOT handles loss of sensor information by generating additional virtual objects when the tracking server estimates loss of sensor information. As a result, SHOT can decrease the number of object tracking failures due to loss of sensor information. The performance of SHOT is evaluated through simulation experiments in comparison with an MHT-based method.

The remainder of the present paper is organized as follows. In Sect. 2, research related to object tracking is described. Then, in Sect. 3, we propose SHOT. In Sect. 4, we evaluate the performance of SHOT through simulation experiments. Finally, we conclude the present paper with a discussion of future research in Sect. 5.

2. Related Research

Object tracking has attracted a great deal of attention from researchers in various fields, such as computer vision and sensor networks. Although SHOT assumes binary sensors, several object tracking methods with a variety of sensors have been proposed.

There have been a number of object tracking methods that do not assume special devices, such as radio frequency identification (RFID), for objects. Most research has been in the field of computer vision and assumes cameras as the sensors [2]. Although cameras can detect the number of objects and can identify individual objects using image processing technologies, cameras generate a large amount of data, require high processor power, and have higher initial costs than other simple sensors. Multi-hypothesis tracking (MHT) [9], which comes from research in the area of computer vision, is a method for tracking multiple objects, and a number of MHT-based tracking methods have been proposed [10], [11]. In MHT, all possible situations, i.e., hypotheses, are evaluated based on sensor information, and the hypothesis with highest event probability is selected as the estimated object trajectory. A hypothesis is described by the number of objects in a monitoring region and the estimated trajectory of each object. Since the number of hypotheses of MHT increases exponentially as the number of objects increases, a Markov-chain-Monte-Carlo-based (MCMC-based) method was proposed in [12] for the solution of NP-hardness.

Recently, a number of researchers have considered object tracking with binary sensor networks, which are composed of binary sensors such as infrared sensors [3]–[6]. In [3], the authors first analyzed the fundamental performance limits of object tracking when sensor nodes are uniformly deployed in the monitoring region. They also proposed an object tracking method that achieves these limits for binary sensor networks. In a previous study [4], the authors applied the particle filter algorithm for tracking multiple objects with binary sensor networks. In this method, the hypothesis in the near future is estimated from current sensor information. Object tracking is then conducted based on the likelihood of estimated hypothesis, which is calculated from sensor information. However, these methods assume that sensor nodes are deployed uniformly, so that the sensing ranges efficiently cover the monitoring region, which requires a large number of sensor nodes in order to cover a wide monitoring region in actual environments.

In SHOT, we consider that sensor nodes are placed only at certain points in a monitoring region. Therefore, the number of sensor nodes in SHOT is less than that in other methods. SHOT is hypothesis-based tracking method that is similar to MHT-based methods. However, SHOT can also handle specific movement scenarios of objects by introducing the novel concept of a virtual world. SHOT is described in detail in what follows.

3. Scenario-Type Hypothesis Object Tracking

3.1 Overview

In SHOT, a monitoring region is divided into multiple micro-cells by placing sensor nodes as shown in Fig. 1. We refer to the location at which a sensor node is placed as a

[†]Hereinafter, we denote objects in the real world as real objects.

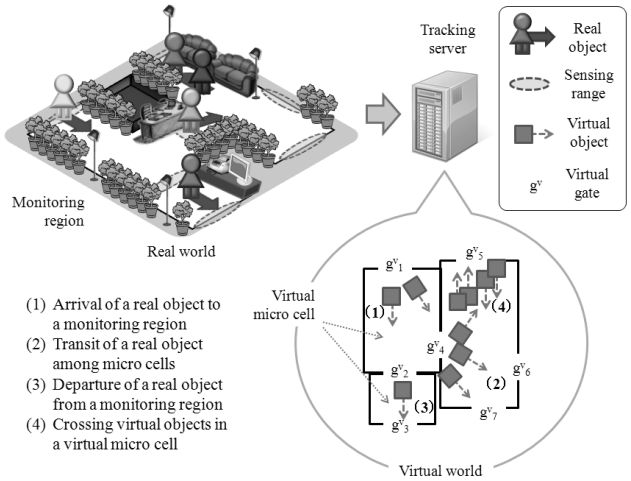


Fig. 2 Overview of SHOT.

gate. We assume that a sensor node is composed of a pair of infrared sensors with a wireless communication device and that the sensor node can detect transit events of objects along with their moving directions. Sensor information is collected in a tracking server through wireless multihop networks. Sensor information consists of the information of transit events of an object along with its moving direction and the time of the event. Object tracking is performed at the tracking server according to the sensor information.

In SHOT, in order to estimate the trajectories of objects, the tracking server maintains a virtual world, which consists of virtual micro-cells, virtual gates, and virtual objects, as shown in Fig. 2. Hereinafter, in order to clearly distinguish virtual objects and real objects, we denote objects in the real world as real objects. The tracking server generates virtual objects when it receives arrival event information of a real object from outside the monitoring region. The tracking server transfers virtual objects in virtual micro-cells according to sensor information and records their trajectories. In addition, the tracking server duplicates and updates virtual objects according to sensor information so that it maintains a number of hypotheses of object trajectories for each real object. Furthermore, the tracking server deletes virtual objects when the possibility that their trajectories correspond to the trajectories of real objects decreases. When the tracking server receives departure event information of a real object from the monitoring region, the trajectory of the most appropriate virtual object is selected as the trajectory of the real object.

In the present paper, we denote a set of micro-cells as $\mathcal{M}$, a set of gates as $\mathcal{G}$, the c-th micro-cell as $m_c \in \mathcal{M}$, the a-th gate as $g_a \in \mathcal{G}$, and the sensor node at gate $g_a \in \mathcal{G}$ as s_a. In addition, we denote a set of virtual micro-cells as $\mathcal{M}^v$, a set of virtual gates as $\mathcal{G}^v$, the c-th virtual micro-cell as $m_c^v \in \mathcal{M}^v$, and the a-th virtual gate as $g_a^v \in \mathcal{G}^v$. Furthermore, we denote the distance between gate g_a and gate g_b as $l(g_a, g_b)$, and a set of virtual objects, which is maintained by the tracking server, as $\mathcal{B}$. The information maintained

Table 1 Information maintained by virtual object b_i.

Parameter	Details
k_i	Object identifier
m_i^v	Virtual micro-cell
$t_{in:i}$	Arrival time
$g_{in:i}^v$	Arrival virtual gate
$t_{out:i}$	Estimated departure time
$g_{out:i}^v$	Estimated departure virtual gate
v_i	Estimated velocity
$t_{ex:i}$	Maximum existing time
(x_i, y_i)	Estimated location

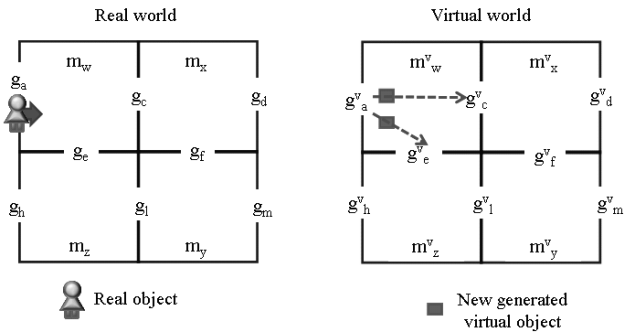


Fig. 3 Arrival of a real object to a monitoring region.

by virtual object $b_i \in \mathcal{B}$ in the tracking server is shown in Table 1. An object identifier is generated for each real object when a real object arrives at the monitoring region. The maximum existing time is used for the decision to remove a virtual object.

In the following, we assume an ideal situation in which the clocks of all sensor nodes are synchronized. In addition, we assume that the tracking server knows the locations of all gates, the distances among all gates, and the mean value $\overline{v_c}$ of the distribution of real object velocities in micro-cells $m_c \in \mathcal{M}$. The following subsections describe how virtual objects are maintained according to sensor information, interactions among virtual objects, and estimation of loss of sensor information.

3.2 Management of Virtual Objects According to Sensor Information

In this subsection, we describe how the tracking server generates, updates, and deletes virtual objects.

3.2.1 Arrival of a Real Object to the Monitoring Region

Consider a real object that arrives at the monitoring region through gate g_a, which is placed on the boundary between the outside of the monitoring region and micro-cell $m_w \in \mathcal{M}$ at time t_1. When the tracking server receives the arrival event information from sensor node s_a on gate g_a, the tracking server generates new object identifier k and generates virtual objects for all virtual gates in virtual micro-cell m_w. We denote a set of newly generated virtual objects as $\mathcal{B}_{arr}$. Figure 3 shows an example of the behavior of SHOT, where

there are four micro-cells and eight gates in the monitoring region. In this example, when a real object arrives at gate g_a, two virtual objects are generated for virtual gates g_c^v and g_e^v in virtual micro-cell m_w^v.

The information of each virtual object $b_i \in \mathcal{B}_{arr}$ is set as follows: Object identifier k_i, virtual micro-cell m_i^v, arrival time $t_{in:i}$, and arrival virtual gate $g_{in:i}^v$ are set as k, m_a, t_1, and g_a, respectively. The estimated velocity v_i and the estimated location (x_i, y_i) are set to $\overline{v_w}$ and the location of virtual gate g_a^v, respectively. The estimated departure time $t_{out:i}$ is calculated as follows:

$$t_{out:i} = t_{in:i} + \frac{l(g_{in:i}^v, g_{out:i}^v)}{v_i}. \tag{1}$$

Maximum existing time $t_{ex:i}$ is calculated as follows.

$$t_{ex:i} = t_{in:i} + \frac{l_{max}(m_i^v, g_{in:i}^v)}{v_i} + \mu_w, \tag{2}$$

where $l_{max}(m_i^v, g_{in:i}^v)$ is the distance between virtual gate $g_{in:i}^v$ and the most distant virtual gate from virtual gate $g_{in:i}^v$ in virtual micro-cell m_i^v. Parameter μ_w (> 0) is a margin in order to consider the estimation error of the velocity of a virtual object and the delay for receiving the sensor information at virtual micro-cell m_w.

The virtual objects in $\mathcal{B}_{arr}$ are then added to $\mathcal{B}$. The tracking server transfers virtual object $b_i \in \mathcal{B}$ toward the estimated departure virtual gate $g_{out:i}$ at velocity v_i in virtual micro-cell m_i^v by updating its location (x_i, y_i).

3.2.2 Transit of a Real Object among Micro-Cells

We next consider the situation in which a real object transits gate g_f from micro-cell m_x to micro-cell m_y at time t_2. When the tracking server receives the transit event information from sensor node s_f on gate g_f, the tracking server first selects virtual object candidates in $\mathcal{B}$, which have possibility to transit virtual gate g_f from virtual micro-cell m_x^v to virtual micro-cell m_y^v. We denote a set of candidates of transit virtual objects as $\mathcal{B}_{cand}$. Figure 4 shows an example of the behavior of SHOT in this situation. In this example, there are three virtual objects in virtual micro-cell m_x^v, and two of these virtual objects, which are indicated by circles, are selected as candidates of the transit virtual object.

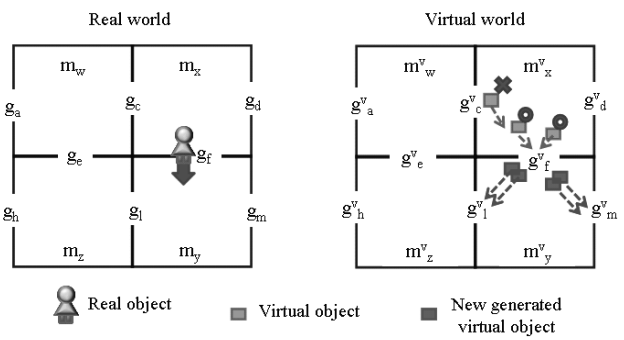


Fig. 4 Transit of a real object among micro-cells.

In the present paper, we propose two selection methods, namely, *velocity-based selection* and *time-based selection*, for selecting a set of candidates of transit virtual objects $\mathcal{B}_{cand}$.

- Velocity-based selection:
  In this method, the estimated velocity square error is used to select $\mathcal{B}_{cand}$. The estimated velocity square error ε_i^2 of virtual object b_i is defined as follows:

$$\varepsilon_i^2 = \left(\frac{v_i^2 \, (t_2 - t_{out:i})}{v_i \, (t_2 - t_{out:i}) - l(g_{in:i}^v, g_{out:i}^v)} \right)^2. \tag{3}$$

  The detailed derivation of the above equation is shown in Appendix.
  Then, $\mathcal{B}_{cand}$ is selected as follows:

$$\mathcal{B}_{cand} = \{b_i \in \mathcal{B} \mid \varepsilon_{min}^2 / \varepsilon_i^2 > \varphi_x, \\ g_{out:i}^v = g_f^v, m_i^v = m_x^v\}. \tag{4}$$

  Here, ε_{min}^2 is the minimum value of the estimated velocity square error among virtual objects for which the estimated departure virtual gate is g_f^v in virtual micro-cell m_c^v.

$$\varepsilon_{min}^2 = \min_{b_i \in \mathcal{B},\ g_{out:i}^v = g_f^v,\ m_i^v = m_x^v} \varepsilon_i^2 \tag{5}$$

  where φ_x ($0 \le \varphi_x \le 1$) is a threshold. By using a smaller value of φ_x, the tracking server may contain a virtual object indicating the true trajectory of a real object with a higher probability. At the same time, a larger number of candidates are selected, and the tracking server should maintain a larger number of virtual objects.
  Additional information can be used to improve the efficiency for selecting $\mathcal{B}_{cand}$. For example, by recording the history of the velocity of a virtual object in that virtual object, threshold φ_x can be adjusted based on the velocity distribution. In addition, the velocity distribution of real objects in the micro-cell can also be used for selecting candidates. A detailed discussion is beyond the scope of the present paper and will be the subject of a future study.
  In velocity-based selection, virtual objects are evaluated by assuming that they transit the virtual micro-cell with constant velocity. However, some micro-cells may have specific movement scenarios for real objects, and the velocities of real objects change dynamically depending on interactions among the real objects and the surrounding environments. In this case, another selection method should be used to select $\mathcal{B}_{cand}$. Therefore, in the following, we propose another selection method.

- Time-based selection:
  In this method, $\mathcal{B}_{cand}$ is selected according to the estimated departure time of virtual objects and the time of the transit event as follows:

$$\mathcal{B}_{cand} = \{b_i \in \mathcal{B} \mid t_{out:i} - \delta_x \le t_2 \le t_{out:i} + \delta_x, \\ g^v_{out:i} = g^v_f, m^v_i = m^v_x\}, \quad (6)$$

where δ_x (> 0) is a threshold in order to consider the estimation error of the departure time in virtual micro-cell m_x. By using a larger value for δ_x, the tracking server may contain a virtual object indicating the true trajectory of a real object with a higher probability, although a larger number of candidates is selected and the tracking server should maintain a larger number of virtual objects.

The tracking server then generates new virtual objects in virtual micro-cell m_y for all virtual gates in virtual micro-cell m_y, from each virtual object $b_i \in \mathcal{B}_{cand}$. In Fig. 4, four virtual objects are generated in virtual micro-cell m^v_y based on the two candidates in virtual micro-cell m^v_x. We denote newly generated virtual objects as $\mathcal{B}_{tran}$. Information of newly generated virtual object $b_j \in \mathcal{B}_{tran}$ is set as follows. Object identifier k_j, virtual micro-cell m^v_j, arrival time $t_{in:j}$, and arrival virtual gate $g^v_{in:j}$ are set as k_i, m_y, t_2, and g^v_f, respectively. Estimated velocity v_j is calculated from observation in virtual micro-cell m^v_x as follows.

$$v_j = \frac{l(g^v_{in:i}, g^v_{out:i})}{t_2 - t_{in:i}}. \quad (7)$$

Estimated departure time $t_{out:j}$ and the maximum existing time $t_{ex:i}$ are calculated from Eqs. (1) and (2), respectively. Estimated location (x_j, y_j) is set to the location of virtual gate g^v_f. Finally, the virtual objects in $\mathcal{B}_{tran}$ are added to B.

3.2.3 Departure of a Real Object from the Monitoring Region

We then consider the situation in which a real object departs from the monitoring region through gate g_h, which is placed on a boundary between the outside of the monitoring region and micro-cell $m_z \in \mathcal{M}$ at time t_4. When the tracking server receives the departure event information from sensor node s_h on gate g_h, the tracking server calculates the minimum value of the estimated velocity square error ε^2_{min} using Eqs. (3) and (5) by considering t_2, g^v_f, and m^v_x as t_4, g^v_h, and m^v_z, respectively. Virtual object b_p, the estimated velocity square error of which becomes ε^2_{min}, is selected as the corresponding real object k_p, and the trajectory of virtual object b_p is chosen as the trajectory of real object k_p. Figure 5 shows an example of the behavior of SHOT in this situation.

3.2.4 Removal of Virtual Objects from the Virtual World

In SHOT, the tracking server deletes virtual objects in the following two cases. In the first case, the tracking server deletes virtual object $b_i \in \mathcal{B}$ at time t when $t_{ex:i} < t$ is satisfied. At the same time, the tracking server also estimates sensor information loss, as described in detail in Sect. 3.4. In the second case, when virtual object b_p is selected as the departure real object, as described in Sect. 3.2.3, the tracking server deletes virtual objects for which the object identifier is k_p from $\mathcal{B}$.

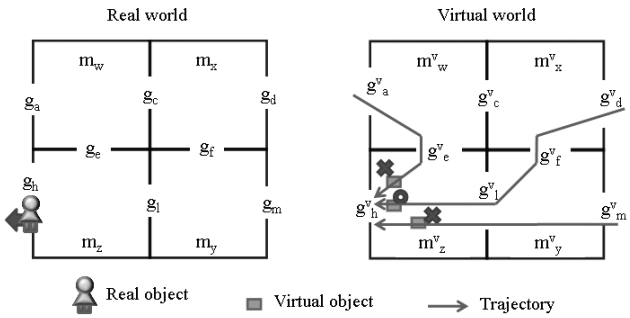


Fig. 5 Departure of a real object from the monitoring region.

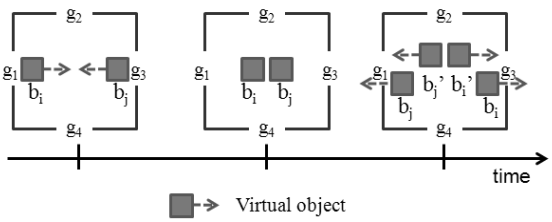


Fig. 6 Passing virtual objects in a virtual micro-cell.

3.3 Management of Virtual Objects According to Interactions among Virtual Objects

In SHOT, by introducing the virtual world, it is possible to handle specific movement scenarios of real objects. For example, the speed of real objects may decrease when real objects pass each other in a narrow passageway. In the present paper, we consider the slowdown of passing objects in a narrow passageway as one example scenario, although a variety of scenarios can be considered, as described in Sect. 1.

We first introduce additional staying time d_i for the information maintained by virtual object b_i. The additional staying time is used to consider the delay of a real object due to specific movement scenarios. Additional staying time d_i is initially set to zero when virtual object b_i is generated. Note that, by introducing additional staying time d_i, Eqs. (3) and (7) are combined to obtained the following equation:

$$\varepsilon^2_i = \left(\frac{v_i^2 \, (t_2 - t_{out:i} + d_i)}{v_i \, (t_2 - t_{out:i} + d_i) - l(g^v_{in:i}, g^v_{out:i})} \right)^2. \quad (8)$$

$$v_j = \frac{l(g^v_{in:i}, g^v_{out:i})}{t_2 - t_{in:i} - d_i}. \quad (9)$$

We then consider the situation in which virtual objects $b_i \in \mathcal{B}$ and $b_j \in \mathcal{B}$ pass each other in a narrow passageway. In this situation, the tracking server generates virtual objects b'_i and b'_j by duplicating b_i and b_j as shown in Fig. 6. Virtual objects b'_i and b'_j indicate hypotheses in which real objects pass each other and decrease in velocity. In the present paper, the decrease in the object velocities is realized by virtual objects stopping for τ in the virtual micro-cell when they pass each other. Here, τ is the delay time for the case of passing real objects in a narrow passageway. Estimated

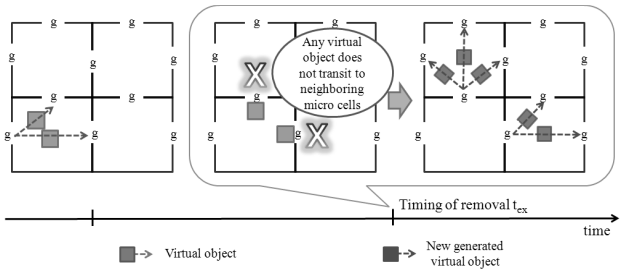


Fig. 7 Loss of sensor information.

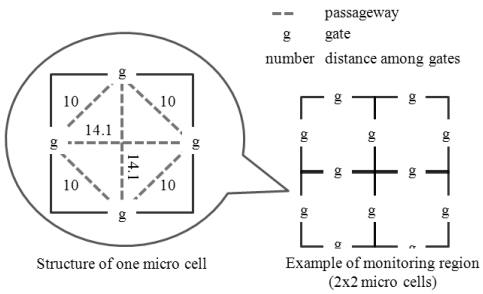


Fig. 8 Structure of a micro-cell.

departure time $t'_{out:i}$, $t'_{out:j}$ and additional staying time d'_i, d'_j of newly generated virtual objects are updated by adding τ. Finally, virtual objects b'_i and b'_j are added to $\mathcal{B}$.

3.4 Management of Virtual Objects According to Estimation of Loss of Sensor Information

In actual environments, information loss occurs due to packet loss, sensing failures, and functional limitations of sensors. In SHOT, when the tracking server estimates the loss of sensor information, the tracking server generates virtual objects for all possible object trajectories to handle the loss of sensor information.

We first introduce transit flag f_i for the information maintained by virtual object b_i in order to estimate the loss of sensor information. Transit flag f_i is first set to *false* when virtual object b_i is generated. Transit flag f_i is set to *true* when virtual object b_i is selected one of candidates, i.e., $b_i \in \mathcal{B}_{cand}$, as described in Sect. 3.2.2.

By setting the maximum existing time according to Eq. (2) in Sect. 3.2.1, the virtual objects generated from the same event and the same virtual object are deleted simultaneously. The estimation of the loss of sensor information is achieved before the deletion process. We denote a set of virtual objects generated from the same event and the same virtual object as $\mathcal{B}_d$ and their maximum existing time as $t_{ex:d}$. At time $t_{ex:d}$, if there is no virtual object $b_i \in \mathcal{B}_d$, the transit flag f_i of which is *true*, the tracking server considers loss of sensor information to have occurred. The tracking server then generates virtual objects in neighboring virtual micro-cells by considering that virtual object $b_i \in \mathcal{B}_d$ transited gate $g_{out:i}$ at time $t_{out:i}$, as shown Fig. 7. The transit flag of newly generated virtual objects is set to *true* in order to prevent virtual objects from being generated again in neighboring virtual micro-cells.

4. Performance Evaluations

4.1 Simulation Settings and Evaluation Metrics

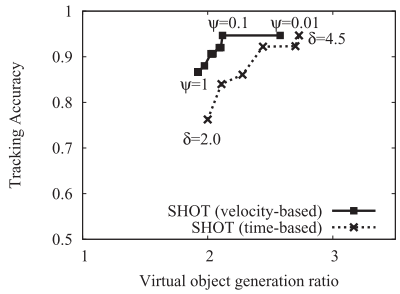
In the simulation experiments, we use a combination of a square micro-cell, which has four gates, as shown in Fig. 8. For example, a monitoring region consists of 2×2 micro-cells, eight gates that face outward from the monitoring region, and four gates inside the monitoring region, as shown in Fig. 8. We introduce passageways among the gates in a micro-cell. A real object moves along the passageway. The distance of the passageways among neighboring gates is 10.0, and the distance of passageways among opposite gates is 14.1. A sensor node is located at each gate. At the sensor node, which is inside the monitoring region, information-loss probability p is introduced, which means that the tracking server fails to retrieve sensor information from the sensor node with probability p.

We consider two mobility models in this paper which are refereed as *constant velocity mobility model* and *random velocity mobility model*. In the constant velocity mobility model, real objects arrive at the monitoring region, following the Poisson arrival process with the arrival rate of λ objects/sec. An arrival gate is randomly selected from among the gates that face outward from the monitoring region. The velocity of real objects follows a Gaussian distribution with an average of 1.31 m/sec and a deviation of 0.272, which is the measurement distribution of pedestrian velocities in an open space area [13]. Real object moves at a constant velocity in the whole monitoring region. When a real object arrives at a micro-cell, the object randomly selects a departure gate from among all gates, excluding the arrival gate in the micro-cell. We should note here that a real object never changes its velocity in the constant velocity mobility model. On the other hand, in the random velocity mobility model, a real object changes its velocity when it passes a gate, in a similar fashion of other mobility models such as random way-point mobility model and the random walk mobility model [14]. The new velocity of real objects follow the same Gaussian distribution as the constant velocity mobility model.

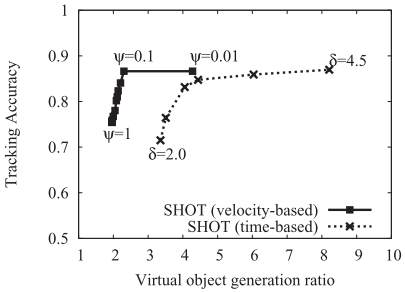
In velocity-based selection and time-based selection, we use same threshold $\varphi_x = \varphi$ and $\delta_x = \delta$, respectively, for all micro-cells $m_x \in \mathcal{M}$. In Eq. (2), we use $\mu_w = 3$ sec for all micro-cells $m_w \in \mathcal{M}$.

We define the tracking accuracy as n_{suc}/n_{obj}, where n_{suc} is the number of real objects for which the trajectories are correctly estimated. Here, n_{obj} is the number of real objects arriving at the monitoring region. In addition, we define the virtual object generation ratio as n_{vobj}/n_{obj}, where n_{vobj} is the number of generated virtual objects in the tracking server. The virtual object generation ratio implicitly indicates the processing overhead of SHOT.

In order to evaluate the effectiveness of the introduc-

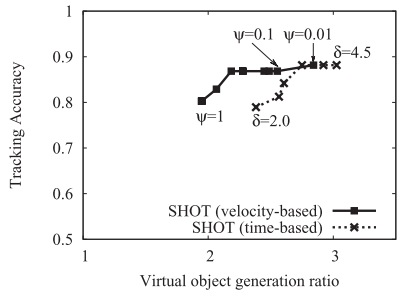


(a) λ = 0.08

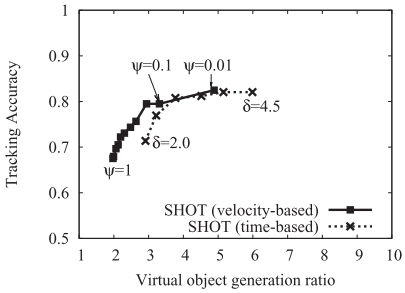


(b) λ = 0.24

Fig. 9 Effect of selection method (constant velocity mobility model).

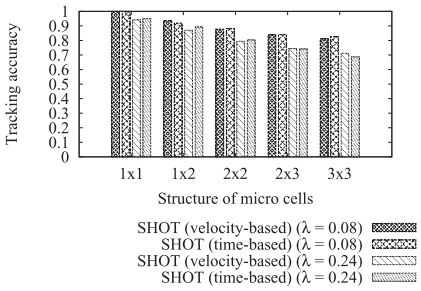


(a) λ = 0.08

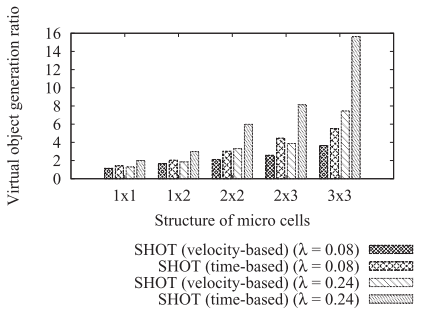


(b) λ = 0.24

Fig. 10 Effect of selection method (random velocity mobility model).



(a) Tracking accuracy.



(b) Virtual object generation ratio.

Fig. 11 Effect of micro-cell structure.

tion of the virtual world, we also conduct simulation experiments using a method based on MHT [9]. In the method for comparison, virtual objects are not deleted when a real object departs from the monitoring region. In addition, virtual objects are not generated when they pass each other in a narrow passageway. Furthermore, the method for comparison does not have an effective mechanism against tracking information loss.

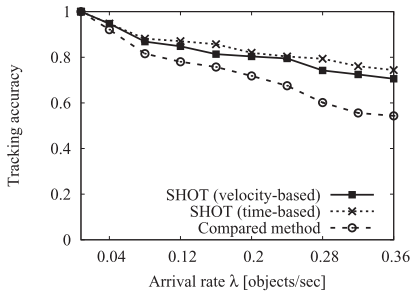
4.2 Effect of Virtual Object Selection Method, Thresholds, and Mobility Model

First, we evaluate the effect of threshold φ in velocity-based selection and threshold δ in time-based selection. Threshold φ is set to be from 0.01 to 1.0, and threshold δ is set to be from 2.0 to 4.5. Arrival rate λ is set to be from 0.08 and 0.24. For the micro-cell structure, 2×2 micro-cells are used. In order to evaluate the fundamental performance of SHOT, we assume that there is no information loss, i.e., p = 0, and that no narrow passageway exists in the simulation experiments. The simulation duration is set to 1,000 sec, which is a sufficient duration for steady results.

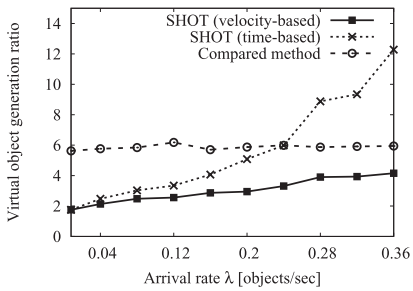
Figure 9 shows the tracking accuracy with respect to the virtual object generation ratio, when we use the constant velocity mobility model. In velocity-based selection, there is the optimal threshold φ in both arrival rate cases to achieve the highest tracking accuracy while maintaining a smaller virtual object generation ratio, as shown in Figs. 9 (a) and 9 (b). Therefore, thresholds should be carefully configured. In time-based selection, by using a larger value of δ, the tracking accuracy becomes higher. However, the virtual object generation ratio also becomes higher. Therefore, a trade-off exists between the tracking accuracy and the virtual object generation ratio.

When we compare results of velocity-based selection and that of time-based selection, the virtual object generation ratios of velocity-based selection is smaller than that of time-based selection to achieve same tracking accuracy. For example, to achieve the tracking accuracy of 0.94 in the constant velocity mobility model, the virtual object generation ratio of velocity-based selection is 22% smaller than that of time-based selection, as shown in Fig. 9 (a). Since the constant velocity model assumes that a real object moves at a constant velocity in the monitoring region, the settings of the simulation are suitable for velocity-based selection.

Figure 10 shows the tracking accuracy with respect to the virtual object generation ratio, when we use the random velocity mobility model. As shown in Fig. 10, both velocity-based selection and time-based selection need similar virtual object generation ratio to achieve the highest tracking accuracy. For example, to achieve the tracking accuracy of 0.88, velocity-based selection needs the virtual object generation ratio of 2.85 and time-based selection needs that of 2.75, as shown in Fig. 10 (a). Since a real object in the random velocity mobility model changes its velocity when it transits a gate, velocity-based selection is less effective compared to the case of the constant velocity mobility model. On the other hand, time-based selection has benefit in terms of simplicity to calculate the set of candidates of transit virtual objects $\mathcal{B}_{cand}$ compared to velocity-based selection. Therefore, time-based selection is suitable for the case of the random velocity mobility model. The selection method should be determined considering mobility behavior of real objects in the monitoring region.

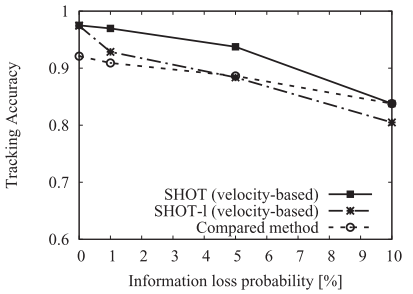


(a) Tracking accuracy.

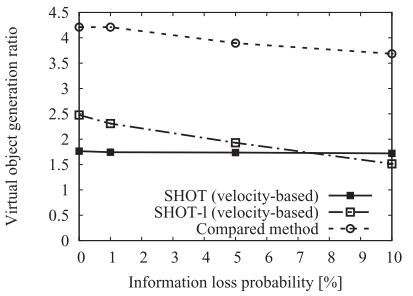


(b) Virtual object generation ratio.

Fig. 12 Effect of passing real objects.

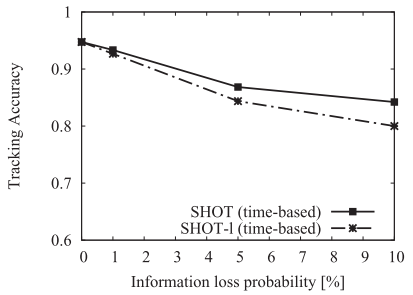


(a) Tracking accuracy.

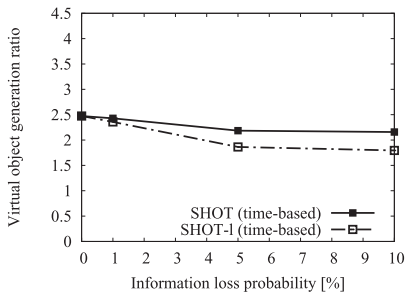


(b) Virtual object generation ratio.

Fig. 13 Effect of sensor information loss (velocity-based selection method).



(a) Tracking accuracy.



(b) Virtual object generation ratio.

Fig. 14 Effect of sensor information loss (time-based selection method).

4.3 Effect of Micro-Cell Structure

Next, we conduct simulation experiments by varying the micro-cell structure as 1×1, 1×2, 2×2, 2×3, and 3×3. The arrival rate λ is set to 0.08 and 0.24. For the mobility model, constant velocity mobility model is used. For thresholds, $\delta = 4.5$ and $\varphi = 0.1$ are used for time-based selection and velocity-based selection, respectively, since each parameter achieves the highest tracking accuracy. The other settings of the simulation experiments are as described in the previous subsection.

Figures 11 (a) and 11 (b) show the tracking accuracy and the virtual object generation ratio, respectively, with respect to structure of micro-cells. As shown in Fig. 11 (a), the tracking accuracy decreases slightly when the number of micro-cells increases. This is because the number of transit micro-cells of real objects increases when the number of micro-cells increases. This means that tracking of a real object becomes more complicated. In addition, the tracking server generates virtual objects according to the transit event of real objects. As a result, the number of generated virtual objects increases when the number of micro-cells increases, as shown in Fig. 11 (b).

Figures 11 also shows that the performance of SHOT depends on the arrival rate of real objects. When the arrival rate increases, the number of transit events of real objects increases and the number of virtual objects in a virtual micro-cell also increases. Therefore, more virtual objects are selected as a set of candidates $\mathcal{B}_{cand}$ in Sect. 3.2.2, and more virtual objects are generated when the tracking server receives transit event information of real objects. As a result, the number of generated virtual objects increases, and the tracking accuracy decreases when the arrival rate increases, as shown in Fig. 11.

When we compare the results of velocity-based selection and that of time-based selection, the tracking accuracy of both selection methods are almost same. However, the virtual object generation ratio of time-based selection is much larger than that of velocity-based selection, since larger number of virtual objects are generated in time-based selection in the constant velocity mobility model as described in Sect. 4.2

4.4 Effect of the Generation of Virtual Objects According to Interactions among Virtual Objects

We evaluate the effect of the generation of virtual objects according to interactions among virtual objects. In the simulation experiments, 2×2 micro-cells in which all passageway are narrow passageway, are deployed. In narrow passageway, the speeds of real objects decrease when the objects pass each other. We use $\tau = 3$ sec and $p = 0$. Arrival rate λ is set to be from 0.008 to 0.36. The other settings are the same as in the previous subsection.

Figures 12 (a) and 12 (b) show the tracking accuracy and the virtual object generation ratio, respectively, with respect to the arrival rate λ. As shown in Fig. 12 (a), in both selection methods, the tracking accuracy of SHOT is higher than that of the method for comparison, since SHOT handles the scenario of the slowdown of passing real objects in a narrow passageway by generating virtual objects when the real objects pass each other. In addition, it is shown that the tracking accuracy decreases as the arrival rate increases because the number of real objects in a micro-cell grows. However, the rate of decrease of the tracking accuracy of

SHOT is smaller than that of the method for comparison. As a result, the tracking accuracy of SHOT with velocity-based selection is improved by 29% and that with time-based selection is improved by 37% when $\lambda = 0.36$.

On the other hand, as shown in Fig. 12 (b), the virtual object generation ratio of SHOT increases as the arrival rate increases in both selection methods, although that of the method for comparison is approximately the same. This is because SHOT generates virtual objects when virtual objects pass each other in a narrow passageway, and the number of passes increases as the number of virtual objects increases.

4.5 Effect of the Generation of Virtual Objects According to the Estimation of Loss of Sensor Information

Finally, we conduct simulation experiments considering the loss of sensor information. In these simulation experiments, arrival rate λ is set to 0.04. In order to evaluate effect of the generation of virtual objects according to the estimation of loss of sensor information, we assume that no narrow passageway exists in the simulation experiments. Information-loss probability p is set to be from 0 to 0.1. The other settings of the simulation experiments are the same as in previous subsection. For the purpose of comparison, we also conduct simulation experiments in which the tracking server does not generate virtual objects when it estimates loss of sensor information. Hereinafter, we denote this method as SHOT-l.

Figures 13 (a) and 13 (b) show the tracking accuracy and the virtual object generation ratio, respectively, with respect to the information-loss probability when we use velocity-based selection. As shown in Fig. 13 (a), the tracking accuracy of all methods decreases the as information-loss probability increases. However, the tracking accuracy of SHOT is higher than that of both SHOT-l and the method for comparison. For example, the tracking accuracy of SHOT is improved by approximately 6% compared to that of SHOT-l when the information-loss probability is $p = 10\%$. These simulation experiments confirmed the effectiveness of the generation of virtual objects according to the estimation of sensor information loss.

As shown in Fig. 13 (b), the virtual object generation ratio of SHOT remains similar, whereas those of SHOT-l and the method for comparison decrease as the information-loss probability increases. This is because SHOT generates virtual objects according to not only sensor information but also the estimation of sensor information loss.

Figure 14 shows the tracking accuracy and the virtual object generation ratio when we use time-based selection. The results and discussions are similar to that of velocity-based selection as described above paragraphs. In time-based selection, the tracking accuracy of SHOT is improved by 5% compared to that of SHOT-l when the information-loss probability is $p = 10\%$.

5. Conclusion and Future Research

In the present paper, an object tracking method with indoor sensor networks called SHOT was proposed. In SHOT, the novel concept of a virtual world that consists of virtual objects and virtual micro-cells is introduced in order to handle specific movement scenarios of objects. Virtual objects are generated, transferred, and deleted according to sensor information, interactions among virtual objects, and the estimation of loss of sensor information. Through simulation evaluations, we confirmed that the ratio of successful tracking is improved by up to 29% by considering the interaction among virtual objects. In addition, the tracking performance was improved by up to 6% by considering the loss of sensor information. Although in the present study we considered only one simple scenario, i.e., slowdown of passing objects in a narrow passageway, the effectiveness of the introduction of the virtual world was demonstrated.

In the future, we intend to take into account various scenarios, such as doubling back of objects and objects on moving sidewalks. Furthermore, we intend to implement SHOT using off-the-shelf sensor nodes and evaluate SHOT through actual experiments.

Acknowledgments

The authors would like to thank anonymous reviewers for their helpful comments.

References

[1] M. Murata, Y. Taniguchi, G. Hasegawa, and H. Nakano, "An object tracking method based on scenario-type hypothesis tracking in segmented multiple regions," Proc. ICNS 2010, pp.200–205, March 2010.
[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Comput. Surv., vol.38, no.13, pp.1–45, Dec. 2006.
[3] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms," Proc. ACM SenSys 2006, pp.251–264, Nov. 2006.
[4] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, "Tracking multiple targets using binary proximity sensors," Proc. IPSN 2007, pp.529–538, April 2007.
[5] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," Proc. ACM SenSys 2003, pp.150–161, Nov. 2003.
[6] W. Kim, K. Mechitov, J.Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," Proc. IPSN 2005, pp.251–264, April 2005.
[7] B.R. Son, S.C. Shin, J.G. Kim, and Y.S. Her, "Implementaiton of the real-time people counting system using wireless sensor networks," Internaitonal Journal of Multimedia and Ubiquitous Engineering, vol.2, no.3, pp.63–80, July 2007.
[8] S.W. Lee, "A room-level indoor location system for smart houses," Proc. SeNAmI 2009, Dec. 2009.
[9] D. Reid, "An algorithm for tracking multiple targets," IEEE Trans. Automatics Control, vol.24, no.6, pp.843–854, Dec. 1979.
[10] T.J. Cham and J.M. Rehg, "A multiple hypothesis approach to figure tracking," Proc. IEEE CVPR 1999, pp.239–245, June 1999.

[11] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," IEEE Aerosp. Electron. Syst. Mag., vol.19, pp.5–18, Jan. 2004.
[12] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," Proc. IEEE CDC 2004, pp.735–742, Dec. 2004.
[13] S. Aihara, M. Sasabe, and H. Nakano, "Mobility model based on incoming and outgoing nodes to an area," Proc. MMT 2007, May 2007.
[14] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," Wireless Communications and Mobile Computing, vol.2, no.5, pp.483–502, 2002.

Appendix: Process of Deriving Estimated Velocity Square Error

Consider a real object that departs from gate g_f at time t_2, and assume that the estimated departure time of virtual object $b_i \in \mathcal{B}$ satisfies $t_{out:i} < t_2$. We then introduce the estimated velocity error ε_i, so that virtual object b_i departs the gate g_f at time t_2 if the velocity of virtual object b_i is $v_i - \varepsilon_i$. Then, the time difference $\Delta t_i = t_2 - t_{out:i}$ becomes

$$\begin{aligned}\Delta t_i &= \frac{l(g^v_{in:i}, g^v_{out:i})}{v_i - \varepsilon_i} - \frac{l(g^v_{in:i}, g^v_{out:i})}{v_i} \\ &= -\frac{\varepsilon_i l(g^v_{in:i}, g^v_{out:i})}{v_i\,(v_i - \varepsilon_i)}. \end{aligned} \tag{A·1}$$

Then, we have

$$\varepsilon_i = \frac{v_i^2\,\Delta t_i}{v_i\,\Delta t_i - l(g_{in:i}, g_{out:i})}. \tag{A·2}$$

We use the square of the estimated velocity error ε_i^2 because ε_i becomes negative when $t_{out:i} > t_2$.

$$\varepsilon_i^2 = \left(\frac{v_i^2\,(t_2 - t_{out:i})}{v_i\,(t_2 - t_{out:i}) - l(g_{in:i}, g_{out:i})}\right)^2. \tag{A·3}$$

Masakazu Murata received his B.E. degree from Kwansei Gakuin University, Japan, in 2008, and his M.E. degree from Osaka University, Japan, in 2010. He is currently an engineer at Fujitsu Limited, Japan. His research interests include ubiquitous networking.

Yoshiaki Taniguchi received his B.E., M.E., and Ph.D. degrees from Osaka University, Japan, in 2002, 2004, and 2008, respectively. Since 2008, he has been an Assistant Professor at the Cybermedia Center, Osaka University. His research interests include wireless sensor networks, wireless mesh networks, and object tracking systems. He is a member of IEEE.

Go Hasegawa received his M.E. and Ph.D. degrees from Osaka University, Japan, in 1997 and 2000, respectively. From 1997 to 2000, he was a Research Assistant at the Graduate School of Economics, Osaka University. He is currently an Associate Professor at the Cybermedia Center, Osaka University. His research is in the area of transport architecture for future high-speed networks. He is a member of IEEE.

Hirotaka Nakano received his B.E., M.E., and D.E. degrees in Electrical Engineering from The University of Tokyo, Japan, in 1972, 1974 and 1977, respectively. He joined NTT Laboratories in 1977 and has been engaged in research and development of videotex systems and multimedia-on-demand systems. He was an executive manager of the Multimedia Systems Laboratory at the NTT Human Interface Laboratories from 1995 to 1999. He was the head scientist of the Multimedia Laboratory at NTT DOCOMO until 2004 and is now a Professor at the Cybermedia Center, Osaka University. His research is in the area of ubiquitous networks. He is a member of IEEE and ITE.