Calculation of Location Probabilities for Agent-based Target Tracking System

Masaru Shiozuka[†‡], Tappei Yotsumoto[†], Kenichi Takahashi[‡], Takao Kawamura[‡], Kazunori Sugahara[‡]

[†]System Engineering Department,
Melco Power Systems Co. Ltd.
Kobe, Japan
email: {Shiozuka.Masaru@zd, Yotsumoto.Tappei@zb}.MitsubishiElectric.co.jp

[‡]Graduate School of Engineering,
Tottori University
Tottori, Japan
email: {takahashi, kawamura, sugahara }@tottori-u.ac.jp

***Abstract*—Target monitoring systems are widely used in various domains such as companies and schools to prevent crimes. Such systems require operators to monitor the information sent from sensor devices, such as cameras and beacon devices. To reduce the workload on the operators, we proposed an automatic target tracking system. However, issues arose due to target recognition errors caused by the sensor devices. To address this problem, we introduced groups of agent to reduce false tracking. The groups were expanded to avoid losing the target; however, the location of the target in the group was unclear. In this study, we calculated the probabilities of the location of the target in a group and improved tracking efficiency. The validity of this approach was confirmed via simulations.**

Keywords-Agent; Target Tracking; Camera; Monitoring Systems.

I. INTRODUCTION

Various types of systems, such as entrance control systems for monitoring a suspicious person, have been introduced as security measures in companies and other places. However, as the number of cameras and tracking targets increases, it becomes difficult for an operator to track all the targets. Therefore, we proposed an agent-based tracking system applicable to an environment where each sensor is installed in a discrete location [1]-[3]. This system comprises cameras, tracking nodes, agents, and a monitoring terminal. In the proposed system, a node with a camera analyzes the data received from cameras. Agents move among the nodes by detecting the features of the target. An operator can follow the location of the target by checking the location of its corresponding agent.

If cameras could monitor area without any blind spots it would be possible to track targets. However, it is unrealistic to install cameras that cover all areas. A more realistic approach is to install a specific number of cameras at set points, e.g., entrances, rooms, and passages. In this case, there are instances when a target is not caught on any camera. Therefore, we proposed a method to calculate which cameras may detect the target next [2]. This method calculates the neighbor relation nodes of each camera based on the value of each camera's shooting area and the floor map.

The system extracts the features of a target from a picture taken by the cameras. However, the features are not always extracted accurately; for example, when a camera tracks a person with brown hair color, their hair color may be recognized as black, depending on the intensity of the light. This results in target recognition errors. Therefore, a person who is not a target may be recognized as the target, and vice versa. This results in false tracking. To address such cases, we introduced groups of agent with two thresholds to reduce false tracking [1]. We defined a group as a set of agents that track the same target. Additionally, two thresholds were introduced (1) a decision threshold to determine a person as the target, and (2) a re-evaluation threshold to postpone the decision. If only one threshold is used, an agent must determine whether a person is its target or not. Hence, false tracking increases if the threshold value is low and non-detection increases if the threshold value is high. By introducing the second threshold, the decision is postponed when the evaluation value is between these two thresholds, and the group is temporarily expanded to cover the nodes where the target may appear next. By expanding the group, it was possible to prevent the target from leaving the group undetected. This method mitigated the cases of false tracking and non-detection.

In this study, group-based tracking was improved. Previous group-based tracking studies [1]-[3] did not consider where the target was in the group. However, in this study, the probabilities of each agent in a group were calculated. Thus, the most probable location of where a target is known by checking the probability of each agent in the group.

This paper is organized as follows: Section II reviews several studies on target tracking systems. Section III provides an overview of the agent-based target tracking system. Section IV explains the derived equations used to calculate probabilities. Section V evaluates the method and Section VI concludes the paper.

 ISBN: 978-1-61208-786-3

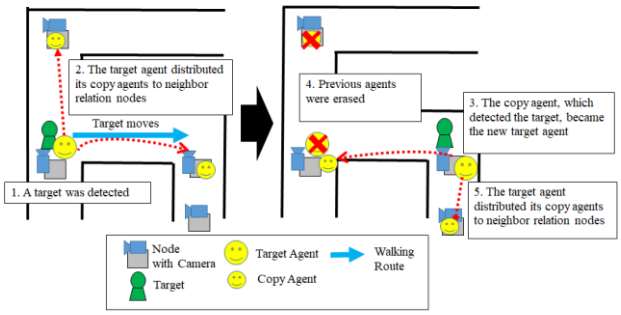

Figure 1. Overview of the proposed system.

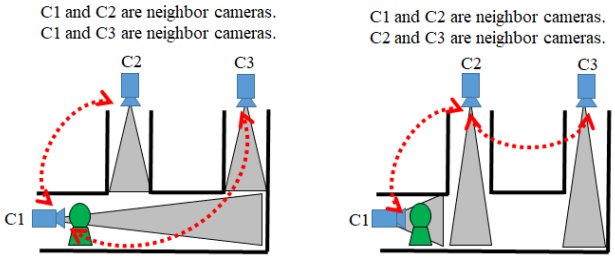

Figure 2. Neighbor relations of neighboring cameras.

II. Related Works

Several studies on target tracking systems using cameras or other devices have been proposed previously.

Wenxi et al. [4] proposed a method to predict the migration route of a target in a crowd by using a high-order particle filter and online-learning. Jin and Bhanu [5] proposed a group structure to improve tracking accuracy when the shooting ranges of cameras overlapped. These studies are not applicable to situations where cameras are installed discretely.

Babenko et al. [6] and Zhang and Maaten [7] proposed an online classifier to improve the tracking accuracy of a single object. Cho et al. [8] proposed a method to automatically create neighbor relationships between cameras; however, this method requires a central server to collect and manage data from cameras. As the number of cameras increases, the system requires expensive machines to manage the increased computational cost.

Alejandro et al. [9] and Bocca et al. [10] proposed a method that tracks a target by analyzing the Received Signal Strength Indication (RSSI) value. Komai et al. [11] also proposed a method that tracks a target using the RSSI values of Bluetooth low energy. The system sends RSSI values to a database server and estimates the location of the target. These methods require the ability to measure the signal strength in advance; thus, it is difficult to expand the tracking area dynamically.

The system in this study assumed a dynamic network, instead of a static network. Therefore, the network could be easily rebuilt when nodes were dynamically added or deleted, or when the shooting range of a camera was changed. Furthermore, this system did not require a central server; thus, when nodes malfunctioned, other nodes continued tracking the targets.

III. Agent Based Target Tracking System

We developed an automatic target tracking system [2]. In this system, one group took charge of one target. Each target was tracked by a single group automatically. An operator could follow the location of each target via its corresponding group.

A. System Overview

Figure 1 shows an overview of the system, which was comprised of targets, cameras, nodes, agents, and a monitoring terminal. A target was defined as a person tracked by a group, which was a program used to identify and track a person using information from cameras. The node connected to a camera included a data analysis function and an execution environment for agents and collected pictures from the camera. Agents moved across nodes along their target. The location of a target was displayed on the monitoring terminal using the location of the agent.

B. Tracking Flow

When a person moved within the shooting range of a camera, the corresponding node took the target's picture. Each agent on the node checked to see if the person was their target. When an agent judged a person was its target, the agent became the "target agent." The target agent sent its copies, called "copy agents," to its neighbor relation nodes, which were the nodes where the target may appear next. Neighbor relation nodes were calculated by the method proposed in [2]. An example of neighbor relation nodes is shown in section III-C. When a copy agent detected its target, the copy agent became the new target agent. The target was then tracked by this new target agent. The original target and copy agents were subsequently erased. The new target agent sent its copy agents to its neighbor relation nodes. Following these steps, an agent tracked a target.

C. Neighbor Relations

Neighbor relation was used to calculate on which cameras a target may be caught next [2]. It calculated the neighbor relation nodes of each camera based on the value of each camera's shooting range and a map of the floor. Figure 2 shows an example of neighbor relations. Red arrows indicate the neighbor relations of cameras. On the left-side of the figure, C1 & C2 and C1 & C3, have neighbor relations; thus, C2 & C3 are the neighbor relation nodes of C1. On the right-side of the figure, because C1 & C2 have a neighbor relation, only C2 is the neighbor relation node of C1. By calculating neighbor relations, the camera on which a target may be caught next can be determined.

 ISBN: 978-1-61208-786-3

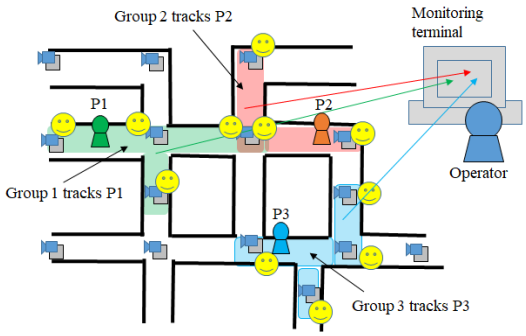

Figure 3. Example of groups.

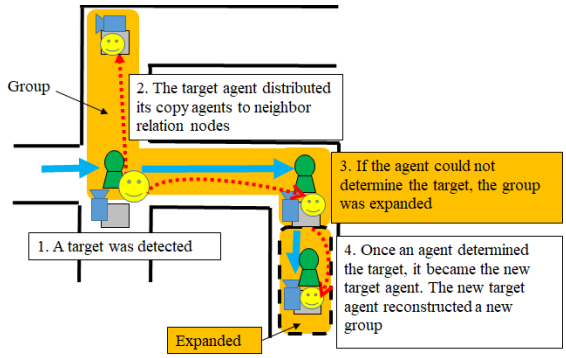

Figure 4. Group expansion.

D. Group

Figure 3 shows examples of groups. Operators can knows the approximate location of a target determined by checking its corresponding group. One group tracked one target. If the system tracked two targets, then there were two groups. If there were any overlaps between the groups, there were multiple agents tracking different targets on nodes where the groups overlapped. Group 1 and Group 2 are shown to overlap in Figure 3.

E. Group Expansion

When a target goes out of the group, the target can no longer be tracked. Therefore, we proposed a group expansion mechanism using the two thresholds previously mentioned [1]. The two thresholds were: (1) a decision threshold, used to determine if a person was a target, and (2) a re-evaluation threshold, used to postpone the decision regarding a target. If the result of a person's evaluation was between these two thresholds, the decision was postponed, and the group was expanded. Figure 4 shows an example of group expansion. If an agent evaluated a person and its evaluation value was between the two thresholds, then the agent sent copy agents to its neighbor relation nodes. This meant the group was expanded to include nodes where the person may appear next. An (copy) agent surrounded by dashed line in Figure 4 joins the group. Because the copy agent stayed on the expanded node, the person could be evaluated again when appearing on the expanded node. Thus, it was possible to continuously track

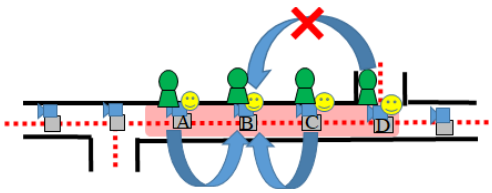

Figure 5. Target move patterns.

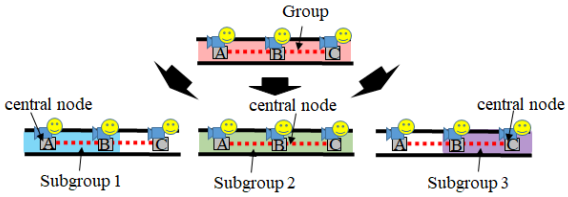

Figure 6. Example of subgroups.

the person. When the evaluation value of the person exceeded the decision threshold, the detected person was determined to be a target. Then, the agent became the new target agent, and other agents in the group were erased (the group was collapsed.) The new target agent sent its copy agents to its neighbor relation nodes for the creation of a new group. By repeating these steps, the agents tracked a target.

IV. Calculation of Probabilities in Group

A group expansion mechanism with two thresholds enabled the system to mitigate false tracking. Thus, the target is known to be somewhere in the group; however, its exact location is unknown. Furthermore, if group expansions are performed frequently, the group will expand endlessly. Thus, we proposed a method to calculate the probabilities of where a target is within the group. Using probability calculations, operators could determine the most probable location of a target (by checking the probability of each agent in the group.)

Considering that the target was moving between nodes, the target should be caught by neighboring nodes. A target could not skip a neighbor relation node. Figure 5 shows a target movement patterns. When a target exists at node B, the target cannot reach node D, without passing node C. Therefore, we first calculated the probabilities that the target will move from a node to its neighbor relation nodes. For this calculation, we divided the group into subgroups, which consisted of a node (hereafter, called the central node) and its neighbor relation nodes. Subsequently, the probabilities of each node in a group were calculated by integrating the probabilities of each node in the subgroups.

A. Probabilities in SubGroup

We defined a subgroup as a set of nodes comprised of a central node and its neighbor relation nodes. Figure 6 shows an example in which a group is divided into subgroups. There are three nodes in the group. The group is divided into three subgroups, each with a different central node; Subgroup 1 is comprised of central node A and its neighbor relation node B;

 ISBN: 978-1-61208-786-3

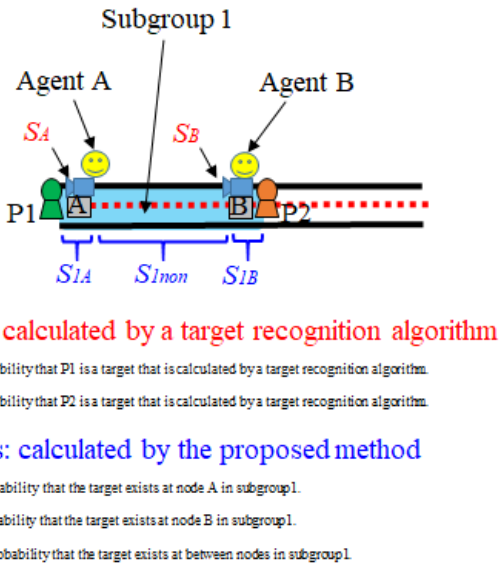

Figure 7. Calculating probabilities in subgroup.

Subgroup 2 is comprised of central node B and its neighbor relation nodes, A and C; Subgroup 3 is comprised of central node C and its neighbor relation node B.

The purpose of breaking a group into subgroups was to calculate the probability of where the target would move next. This was important considering that a target might be along the edges between two nodes after the central node identified the target for the last time.

1) Calculating Probabilities within each SubGroup

To calculate the probability that the target was in each node in a subgroup, it was assumed that there were two people in subgroup 1, P1 was at node A and P2 was at node B illustrated in Figure 7. The probability of that P1 was a target was s_A, and that of P2 was s_B. There were three cases identified where P1 was a target, P2 was a target, and neither P1 nor P2 were a target. Because P1 was a target only if P2 was not a target, the probability of P1 as a target can be represented by (1).

$$s_A \times (1 - s_B) \tag{1}$$

Because P2 was a target only if P1 was not a target, the probability of P2 as a target can be represented by (2).

$$(1 - s_A) \times s_B \tag{2}$$

The probability that neither P1 nor P2 was a target is represented by (3).

$$(1 - s_A) \times (1 - s_B) \tag{3}$$

These equations were generalized to represent *n* nodes in a subgroup, where a target was detected by an agent, with a probability of s_i at node *m*. The probability, that a person detected at node *m* was a target, is represented by (4).

$$s_m \times \prod_{\substack{i=1 \\ i \neq m}}^{n} (1 - s_i) \tag{4}$$

The probability that the target was not observed by any nodes in the subgroup is also generalized by (5).

$$\prod_{i=1}^{n} (1 - s_i) \tag{5}$$

2) Observability and Normalization

The probability that a target exists at node *m* in a subgroup was calculated. However, the possibility of observing the target decreased, if the distance between the nodes was significant. Conversely, the possibility of observing the target increased if the distance was small. For example, if two cameras were installed at both ends of a long passage, then, a wide area would not be covered; therefore, the possibility of being identified by either of two cameras would be small. Thus, we introduced a probability α that a target can be observed. Then, the probabilities in (4) and (5) were reformed as follows.

$$\left\{ s_m \times \prod_{\substack{i=1 \\ i \neq m}}^{n} (1 - s_i) \right\} \times \alpha \tag{6}$$

$$\left\{ \prod_{i=1}^{n} (1 - s_i) \right\} \times (1 - \alpha) \tag{7}$$

These probabilities were normalized by their ratio. Then, the probability S_m that a target exists at node *m* was represented by (8).

$$\begin{aligned} S_m &= \frac{(6)}{\{\sum_{i=1}^{n}(6)\} + (7)} \\ &= \frac{\left\{ s_m \times \prod_{\substack{i=1 \\ i \neq m}}^{n} (1 - s_i) \right\} \times \alpha}{\left\{ \sum_{i=1}^{n} \left(\left\{ s_i \times \prod_{\substack{k=1 \\ k \neq m}}^{n} (1 - s_k) \right\} \times \alpha \right) \right\} + \left\{ \{\prod_{i=1}^{n}(1 - s_i)\} \times (1 - \alpha) \right\}} \end{aligned} \tag{8}$$

The probability S_{non} that a target is not observed was represented by (9).

$$\begin{aligned} S_{non} &= \frac{(7)}{\{\sum_{i=1}^{n}(6)\} + (7)} \\ &= \frac{\{\prod_{i=1}^{n}(1 - s_i)\} \times (1 - \alpha)\}}{\left\{ \sum_{i=1}^{n} \left(\left\{ s_i \times \prod_{\substack{k=1 \\ k \neq m}}^{n} (1 - s_k) \right\} \times \alpha \right) \right\} + \left\{ \{\prod_{i=1}^{n}(1 - s_i)\} \times (1 - \alpha) \right\}} \end{aligned} \tag{9}$$

To summarize briefly, s_A and s_B are probabilities calculated by a target recognition algorithm. Then, the probability S_{1A} and S_{1B} that a target exists at node A and node B in subgroup1 are calculated by equation (8). Then, the probability S_{1non} that a target exists between nodes is calculated by equation (9).

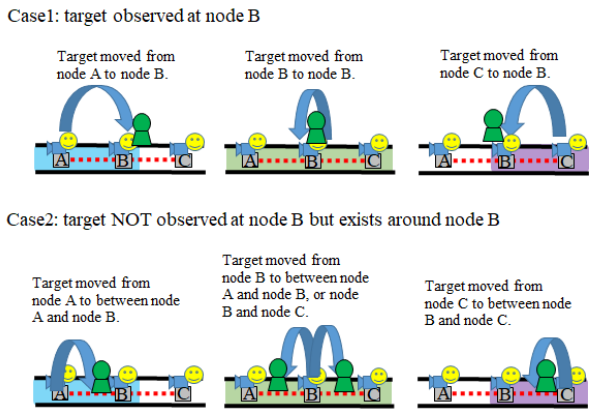

Figure 8. Target move cases in a group.

B. Probabilities in Group

To calculate probabilities in a group, the assumption was made that there was a group illustrated in Figure 6. The probability that the target exists at node B, or around node B, was calculated by the sum of the following two cases.

- Case 1. A target moved from the neighbor relation nodes A or C, to node B. Additionally, a target moved from node B to B (that is, the target stayed at node B.)
- Case 2. A target existed around node B; however, the target was not observed.

Figure 8 shows the details of above two cases. We calculated the probabilities of the above cases, 1 and 2. Supposing that $S1_A, S1_B, S1_{non}$ were the probabilities calculated in (8) and (9) for subgroup 1; $S2_A, S2_B, S2_C, S2_{non}$ were the probabilities for subgroup 2; and $S3_B, S3_C, S3_{non}$ were the probabilities for subgroup 3. ${G_A}', {G_B}', {G_C}'$ were the probabilities that a target is identified at nodes A, B, and C for the last time, respectively.

Case 1: Occurred when a target was at node A for the last time and subsequently the target was at node B in subgroup 1. A target was at node C for the last time and then the target was at node B in subgroup 3, or when a target was at node B for the last time, and stayed at node B in subgroup 2. Thus, the probability of Case 1 is represented by (10).

$$ {G_A}' \times S1_B + {G_B}' \times S2_B + {G_C}' \times S3_B \quad (10) $$

Case 1 was further generalized by (11), where a node *m* has *n* neighbor relation nodes.

$$ \sum_{i=1}^{n} ({G_i}' \times Si_m) \quad (11) $$

Case 2: Occurred when a target existed around node B; however, the target was not observed. This was calculated using the sum of the probabilities of the cases where a target exists around a node; however, the target is not observed in each subgroup. Hence, the probability of Case 2 is represented by (12).

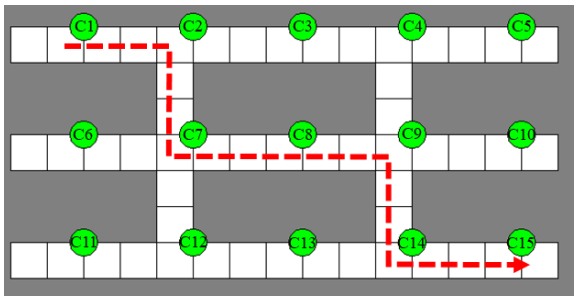

Figure 9. Floor map.

$$ ({G_A}' \times S1_{non} + {G_B}' \times S2_{non} + {G_C}' \times S3_{non}) \times {G_B}' \quad (12) $$

Case 2 was further generalized by (13), where there are *n* subgroups.

$$ \left\{\sum_{i=1}^{n} ({G_i}' \times Si_{non})\right\} \times {G_m}' \quad (13) $$

Since the probability G_m that the target was at node *m* in a group is the sum of (11) and (13), it is represented by (14).

$$ \begin{aligned} G_m &= (11) + (13) \\ &= \sum_{i=1}^{n} ({G_i}' \times Si_m) \\ &+ \left\{\sum_{i=1}^{n} ({G_i}' \times Si_{non})\right\} \times {G_m}' \end{aligned} \quad (14) $$

Thus, the probabilities of each node in a group can be calculated.

V. EXPERIMENTS

A simulation environment was implemented to evaluate the proposed method. For this simulation, we evaluated whether the probabilities could be obtained correctly for situations where target recognition errors occurred. The purpose of the simulation was to verify the validity of the proposed method.

A. Simulation Settings

1) Floor Map

Figure 9 shows a floor map used for the simulation. The green circle represents a camera. There are 15 cameras on the floor. The white square blocks represent a passage.

The walking route of a target is denoted by the red line in Figure 9. The target P1 moves between the cameras in the order of C1→C2→C7→C8→C9→C14→C15. Table I shows the walking routes of P1 to P8. A maximum of eight targets are assumed to be walking at the same time.

TABLE I. WALKING ROUTE

TargetID	Walking Route
P1, P5	C1→C2→C7→C8→C9→C14→C15
P2, P6	C11→C12→C7→C8→C9→C4→C5
P3, P7	C5→C4→C9→C8→C7→C12→C11
P4, P8	C15→C14→C9→C8→C7→C2→C1

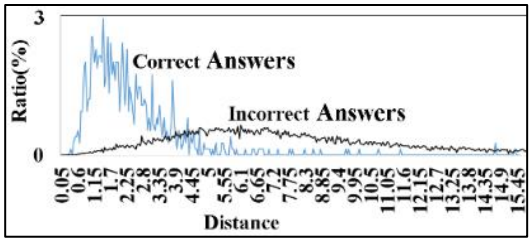

Figure 10. Distribution of correct/incorrect answers.

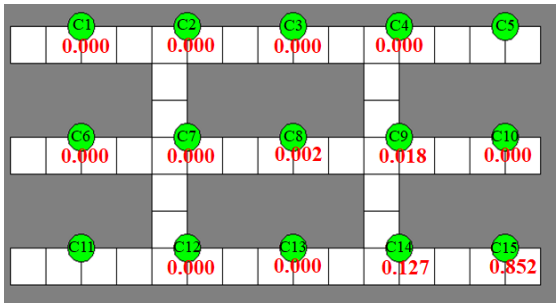

Figure 11. Tracking result of P1.

2) Simulation Data

For this simulation, we used the proposed target recognition method [12]. This method calculates the distance between persons, using several features, such as the clothes and height of the target. Several distances were obtained by applying the target recognition method to the public pictures' dataset (SARC3D [13] in the PETA dataset [14]). SARC3D consisted of 50 different people. Each person had four pictures taken, from four different directions. Thus, SARC3D had 200 pictures. Figure 10 shows a distribution of the distances of the correct answers and incorrect answers. A correct answer was the distance between two pictures of the same person. An incorrect answer was the distance between two pictures of different people. When a target approached a node with a camera, the node utilized the distance from the correct answers randomly. The agent used a ratio of the correct answers on the distance, to determine the probability of a target.

B. Results

1) Tracking Result of P1

Figure 11 shows the snapshot of when P1 reaches the goal, C15. The numbers in Figure 11 indicate the existence probabilities of the target at each node. The highest probability is 0.852, for the last camera, C15. The next highest probability is 0.127, for camera C14, which the target passed immediately before C15. Table II shows the probabilities of each node in each time period. The cells in which the target existed are shaded. For example, when 8 seconds passed (t=8), the target was at C7, and the probability of C7 was 0.924. This was the highest probability at t=8. The experimental results show that the probabilities changed according to the movement of the target, and the tracking was successful.

TABLE II. TRACKING RESULT OF P1 DETAILS

Camera No.	t=0	t=4	t=8	t=12	t=16	t=20	t=24
C1	*1.000*	*0.072*	*0.005*	*0.001*	0.000	0.000	0.000
C2	0.000	*0.928*	*0.071*	*0.008*	*0.001*	0.000	0.000
C7	0.000	0.000	*0.924*	*0.101*	*0.012*	*0.002*	0.000
C8	0.000	0.000	0.000	*0.890*	*0.106*	*0.015*	*0.002*
C9	0.000	0.000	0.000	0.000	*0.881*	*0.122*	*0.018*
C14	0.000	0.000	0.000	0.000	0.000	*0.862*	*0.127*
C15	0.000	0.000	0.000	0.000	0.000	0.000	*0.852*

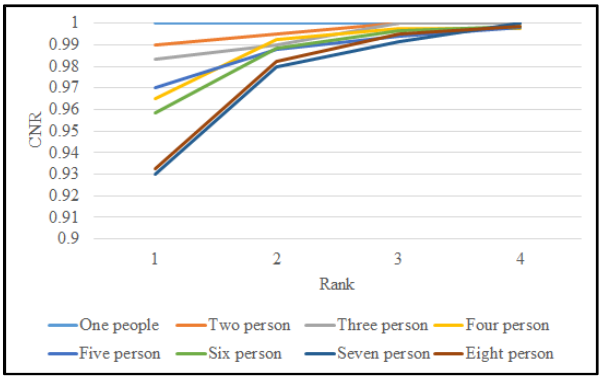

Figure 12. Rank of tracking results

The probabilities of each node gradually decreased after the target moved to other nodes. For example, the probability of C7 was 0.924 at t=8; however, it gradually decreased to 0.101, 0.012, and 0.002 with the elapse of time. Even if the target was not detected, the probability did not immediately become zero. This was because we considered that there were cases when the target was not observed, despite its existence around the node C7. If the probability became zero immediately, the target would never be caught when the target moved to C2 (from the unobserved situation around C7.) It would have resulted in the loss of the target. Because we consider unobserved situations (14), the probabilities gradually decrease and finally become zero.

2) Tracking Result of P1 to P8

Figure 12 shows the rank of *n* cumulative accuracy rates (CNRs) when the simulation was conducted 100 times. The CNR indicated the rank orders of a node targets existence. Figure 12 shows 93% of targets existed on the node of 1st rank, and 99% exited at the 4th rank.

For comparison, a comparative system that regards a person with the highest probability as the target was created. Figure 13 shows the proposed method tracks with a higher accuracy rate than the comparative system.

 ISBN: 978-1-61208-786-3

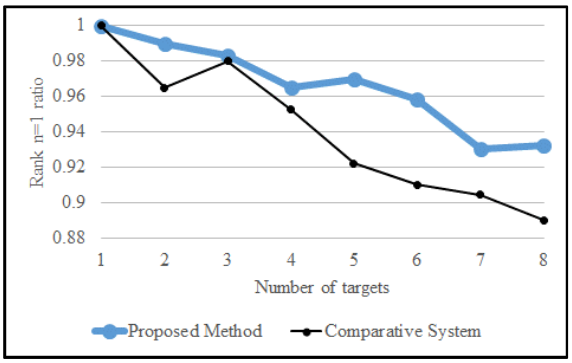

Figure 13. Comparison with the comparative system

C. Discussion

This section discusses the limitations of the proposed method. The proposed method requires a condition that a distance of a target tends to be closer than that of a distance of a non-target as shown in Figure 10. The proposed method calculates a probability of a node based on its neighbor relation nodes. In other words, when the probability of a certain node is high, the probabilities of its neighbor relation nodes calculated tends to be higher. Therefore, if the probability of a non-target is accidentally high, the probability of its neighbor relation nodes tends to be calculated higher. These probabilities are unexpected results. However, these probabilities are temporary. These probabilities will gradually become proper values through the evaluation of several nodes because the probability of a target tends to be higher than that of the non-tracking target as shown in Figure 10.

VI. Conclusion and Future Work

In this study, we proposed a method to calculate the probabilities of the location of a target in a group of agents. In the future, we plan to evaluate the validity of the proposed method in an actual environment. In an actual environment, walking routes of targets will be more complicated, such as walking the same routes repeatedly or turning back. In these cases, the probabilities will be updated frequently and the probabilities may become too high or low. We have to handle such cases by adjusting the results.

References

[1] M. Shiozuka et al., “Countermeasure to Human Recognition Error for Agent-based Human Tracking System,” 12th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM2018), pp. 65-70, 2018.

[2] T. Yotsumoto et al., “Automatic Human Tracking System using Localized Neighbor Node Calculation,” Sensors & Transducers, Vol. 194, No. 11, pp. 54-61, 2015.

[3] T. Yotsumoto, M. Shiozuka, K. Takahashi, T. Kawamura, and K. Sugahara, “Hidden neighbor relations to tackle the uncertainness of sensors for an automatic human tracking,” 2017 Second IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT 2017),. Coimbatore, India, pp. 690-696, 2017.

[4] L. Wenxi, C. Antoni, L. Rynson, and M. Dinesh, ”Leveraging long-term predictions and online learning in agent-based multiple person tracking,” IEEE Transactions on Circuits and Systems for Video Technology, Vol.25, No.3, pp. 399-410, 2015.

[5] Z. Jin and B. Bhanu, “Multi-camera Pedestrian Tracking using Group Structure,” International Conference on Distributed Smart Cameras, Article No. 2, pp. 1-6, 2014.

[6] B. Babenko, M.-H. Yang, and S. Belongie, “Robust Object Tracking with Online Multiple Instance Learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No.8, pp. 1619-1632, 2011.

[7] L. Zhang and L. van der Maaten, “Preserving Structure in Model-Free Tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, No. 4, pp. 756-769, 2014.

[8] Y. J. Cho, S. A. Kim, J. H. Park, K. Lee, and K. J. Yoon, “Joint Person Re-identification and Camera Network Topology Inference in Multiple Camera,” arXiv:1710.00983, 2017.

[9] C. Alejandro, M. Antoni, B. Marc, and V. Jose, ”Navigation system for elderly care applications based on wireless sensor networks,” Signal Processing Conference (EUSIPCO 2012), Proceedings of the 20th European. IEEE, pp. 210-214, 2012.

[10] M. Bocca, O. Kaltiokallio, and N. Patwari, “Multiple Target Tracking with RF Sensor Networks,” IEEE Transactions on Mobile Computing, Vol. 13, No. 8, pp. 1787-1800, August, 2014.

[11] K. Komai et al., “Elderly Person Monitoring in Day Care Center using Bluetooth Low Energy,” 10th International Symposium on Medical Information and Communication Technology (ISMICT 2016), Worcester, MA, USA, pp. 140-144, 2016.

[12] M. Nishiyama et al., “Person Re-identification using Co-occurrence Attributes of Physical and Adhered Human Characteristics,” 23rd International Conference of Pattern Recognition (ICPR), pp. 2086-2091, 2016.

[13] SARC3D, http://www.openvisor.org/sarc3d.asp, April, 2020.

[14] Y. Deng, P. Luo, C. Loy, and X. Tang, “Pedestrian attribute recognition at far distance,” ACM Multimedia, pp. 3-7, 2014.

 ISBN: 978-1-61208-786-3