Sensors **2012**, *12*, 14489-14507; doi:10.3390/s121114489

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Laser-Based Pedestrian Tracking in Outdoor Environments by Multiple Mobile Robots

Masataka Ozaki [1], Kei Kakimuma [1], Masafumi Hashimoto [2],* and Kazuhiko Takahashi [2]

[1] Graduate School of Doshisha University, Tatara, Kyotanabe, Kyoto 6100321, Japan; E-Mails: mantuuman@gmail.com (M.O.); dtl0726@mail4.doshisha.ac.jp (K.K.)

[2] Faculty of Science and Engineering, Doshisha University, Tatara, Kyotanabe, Kyoto 6100321, Japan; E-Mail: katakaha@mail.doshisha.ac.jp

* Author to whom correspondence should be addressed; E-Mail: mhashimo@mail.doshisha.ac.jp; Tel./Fax: +81-774-65-6410.

Received: 16 August 2012; in revised form: 20 October 2012 / Accepted: 21 October 2012 / Published: 29 October 2012

Abstract: This paper presents an outdoors laser-based pedestrian tracking system using a group of mobile robots located near each other. Each robot detects pedestrians from its own laser scan image using an occupancy-grid-based method, and the robot tracks the detected pedestrians via Kalman filtering and global-nearest-neighbor (GNN)-based data association. The tracking data is broadcast to multiple robots through intercommunication and is combined using the covariance intersection (CI) method. For pedestrian tracking, each robot identifies its own posture using real-time-kinematic GPS (RTK-GPS) and laser scan matching. Using our cooperative tracking method, all the robots share the tracking data with each other; hence, individual robots can always recognize pedestrians that are invisible to any other robot. The simulation and experimental results show that cooperating tracking provides the tracking performance better than conventional individual tracking does. Our tracking system functions in a decentralized manner without any central server, and therefore, this provides a degree of scalability and robustness that cannot be achieved by conventional centralized architectures.

Keywords: pedestrian tracking; multi-mobile robots; laser range scanner; Bayesian filter; decentralized multi-sensor fusion

1. Introduction

Tracking (*i.e.*, estimating the motion) of pedestrians is important to ensure safe navigation of mobile robots and vehicles. There has been much interest in the use of stereo vision or a laser range scanner (LRS) in mobile robotics and vehicle automation [1–5]. We previously presented a pedestrian tracking method using LRS mounted on mobile robots and automobiles [6–8].

Recently, many studies related to multi-robot coordination and cooperation have been conducted [9,10]. When these robots and vehicles are located near each other, they can share their sensing data. This implies that the robots and vehicles are considered to be a multi-sensor system. Therefore, even if pedestrians are located outside the sensing area of any individual robot or vehicle, it can detect pedestrians using the tracking data received from other robots and vehicles in the vicinity, and thus, multiple robots can improve the accuracy and reliability of pedestrian tracking.

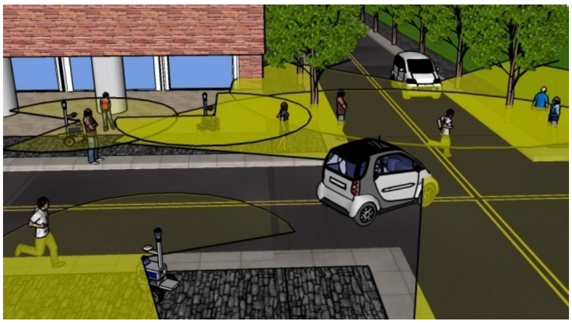
In an intelligent transport system (ITS), if the tracking data is shared with neighboring vehicles through vehicle-to-vehicle communication, each vehicle can detect pedestrians efficiently. This facilitates the construction of an advanced driver-assistance-system. Even if pedestrians suddenly run into roads, the vehicles can detect them, and hence drivers can stop their vehicles to prevent an accident.

This paper presents a pedestrian tracking method employing multiple mobile robots and vehicles. Most studies of cooperative tracking by multiple mobile robots focus on motion planning and controlling issues [11–13]. These studies attempt to keep many moving objects visible to the mobile robots at all times while consuming as little motion energy as possible. In this paper, we address sensor-data fusion, through which pedestrian tracking is achieved by combining the tracking data from multiple mobile robots located in their vicinity.

There has been considerable research in cooperative pedestrian tracking using multiple static sensors located in the environment [14–18] and multi-sensors on robots [19,20]. Our previous work [8] presented a pedestrian tracking method using in-vehicle multi-laser range scanners; pedestrians were tracked by each LRS based on a Kalman filter. In order to enhance the tracking performance, the tracking data were blended based on covariance intersection (CI) method [21].

In this paper, we extend our previous method to pedestrian tracking with multiple mobile robots in the proximity to each other. As illustrated in Figure 1, our method contributes toward building a cooperative pedestrian tracking system using vehicles such as mobile robots, cars, and electric personal assistive mobility devices (EPAMD) in future urban city environments.

Figure 1. Pedestrian tracking system using multiple vehicles such as mobile robots, cars, and EPAMD.

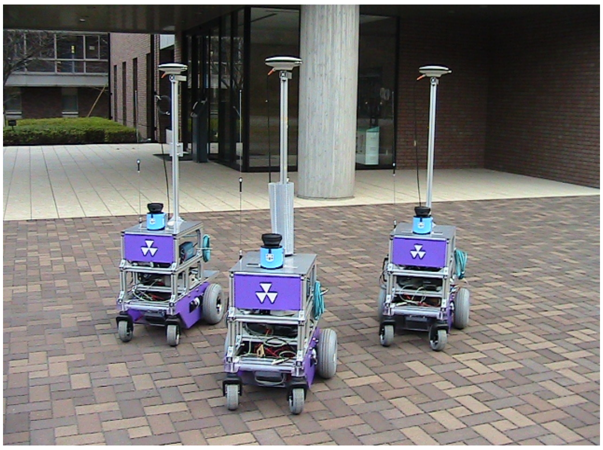


Recent studies [22,23] in cooperative pedestrian tracking by multiple mobile robots require centralized data fusion with a central server; sensing data captured by each robot are sent to a central server for subsequent data fusion. The centralized data fusion reduces system robustness and scalability. Our cooperative tracking system proposed in this paper functions in a decentralized manner without any central server. This paper is organized as follows: in Section 2, we present an overview of our experimental system. In Sections 3 and 4, we present methods of pedestrian tracking and robot localization. In Section 5, we describe simulation and experiment of pedestrian tracking to validate our method, followed by our conclusions.

2. Experimental Mobile Robots

Figure 2 shows our mobile robot system used in the experiments. We use a Okatech Mecrobot wheeled mobile robot platform. Three robots each have two independent drive-wheels. A wheel encoder is attached to each of the drive wheels to measure the wheel velocity. A fiber-optic yaw rate gyro (Tamagawa Seiki, TA7319N3) is attached to the robot's chassis to measure the turn velocity. This information is used to estimate the robot's posture based on dead reckoning. Moreover, each robot is equipped with an RTK-GPS (Novatel ProPak-V3 GPS receiver) to identify its own posture in outdoor environments. The RTK-GPS provides three types of solution: fixed, float, and single solutions. Fixed solution offers range accuracy of less than 0.2 m, and float solution achieves range accuracy of about 0.2 to 1 m. In outdoor environments causing GPS multipath problems and bad weather conditions, we get single solutions with range accuracies of several meters.

Figure 2. Overview of the mobile robot system.



The robot is equipped with a single-layered LRS (Sick LMS100). The LRS captures laser scan images that are represented by a sequence of distance samples in a horizontal plane of 270 deg. The angular resolution of the LRS is 0.5 deg, and the number of distance samples is 541 in one scan image. The onboard computer is a Lenovo ThinkPad R500 with a 2.4 GHz Intel core 2 duo processor, and the operating system used is Microsoft Windows Vista. The sampling frequency of the sensors is 10 Hz.

Broadcast communication via a wireless LAN is used to exchange information among the robots. It takes approximately 40 ms to exchange information between robots. We employ a ring-type network structure in which the robots transmit information in the sequence: robot #1, #2, and #3.

3. Pedestrian Tracking

3.1. Overview

We define two coordinate frames: the world coordinate frame, $\Sigma_w(O_w : X_w Y_w)$ and the i-th robot coordinate frame, $\Sigma_i(O_i : X_i Y_i)$ attached at the robot body, where $i = 1, 2, 3$. Each robot independently detects pedestrians using its own laser image based on an occupancy-grid-based method. Table 1 briefly shows our occupancy grid algorithm in the pseudo-code format. Our detection method is detailed in [6,7].

The detected pedestrians are tracked using the following two tracking modes (Figure 3):

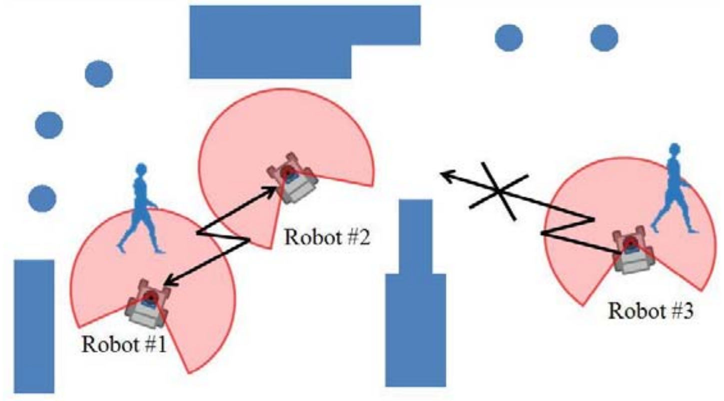
(a) *Individual tracking by a single robot*: Each robot individually tracks pedestrians without any tracking data from other robots. The robot can only track pedestrians inside its LRS sensing area.

(b) *Cooperative tracking by multiple robots*: The robots track pedestrians by sharing their own tracking data so that each robot can track pedestrians both inside and outside its LRS sensing area.

Table 1. Occupancy grid algorithm.

```
1. Let $C\,[X_{\max}, Y_{\max}]$ be a two dimensional array of cells counting the number of observations, where $X_{\max}$
   and $Y_{\max}$ are the maximum $X$ and $Y$ coordinates.
2. Intialize all cells in $C$ to zero.
3. Make an observation with laser range scanner.
4. Determine which cells in $C$ are occupied in the current laser scan image, and increment the occupied cells
   $C\,[X, Y]$.
5. If $C\,[X, Y] == 0$, then we have no information on the cell—free space.
6. If $C\,[X, Y] \geqq 7$, then the cell is “static cell”—static object.
7. If $0 < C\,[X, Y] < 7$, then the cell is “moving cell”—moving object; pedestrian.
8. Repeat from step 3.
```

Figure 3. Tracking mode; robots #1 and #2 track a pedestrian in cooperative tracking mode, and robot #3 tracks a pedestrian in individual tracking mode. The red arc indicates the LRS sensing area.



3.2. Individual Tracking

A pedestrian position in Σ_w is denoted by (x, y). If the pedestrian is assumed to move at almost constant velocity, the rate kinematics is given by:

$$\begin{aligned}\boldsymbol{x}_{(t)} &= \boldsymbol{F}\boldsymbol{x}_{(t-1)} + \boldsymbol{G}\Delta\boldsymbol{x}_{(t-1)} \\ &= \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_{(t-1)} + \begin{pmatrix} \tau^2/2 & 0 \\ \tau & 0 \\ 0 & \tau^2/2 \\ 0 & \tau \end{pmatrix} \Delta\boldsymbol{x}_{(t-1)}\end{aligned} \tag{1}$$

where $\boldsymbol{x} = (x, \dot{x}, y, \dot{y})^T$. $\Delta\boldsymbol{x} = (\Delta\ddot{x}, \Delta\ddot{y})^T$ is an unknown acceleration (plant noise). τ is a sampling period of sensors; in our experimental system, τ is 0.1 s.

The measurement model related to the pedestrian is then:

$$\begin{aligned}\boldsymbol{z}_{(t)} &= \boldsymbol{H}_{i\,(t)}\boldsymbol{x}_{(t)} + \boldsymbol{H}'_{i\,(t)}\boldsymbol{u}_{i\,(t)} + \Delta\boldsymbol{z}_{(t)} \\ &= \begin{pmatrix} \cos\psi_{i\,(t)} & 0 & -\sin\psi_{i\,(t)} & 0 \\ \sin\psi_{i\,(t)} & 0 & \cos\psi_{i\,(t)} & 0 \end{pmatrix} \boldsymbol{x}_{(t)} + \begin{pmatrix} \cos\psi_{i\,(t)} & -\sin\psi_{i\,(t)} \\ \sin\psi_{i\,(t)} & \cos\psi_{i\,(t)} \end{pmatrix} \boldsymbol{u}_{i\,(t)} + \Delta\boldsymbol{z}_{(t)}\end{aligned} \tag{2}$$

where $\boldsymbol{z} = (z_x, z_y)^T$ is the measurement represented in Σ_i. $\Delta\boldsymbol{z}$ is the measurement noise. $\boldsymbol{u}_i = (x_i, y_i)^T$ is the position of the *i*-th robot in Σ_w. ψ_i is the orientation of the *i*-th robot in Σ_w. The posture (position and orientation) $\boldsymbol{x}_i = (x_i, y_i, \psi_i)^T$ is determined using the localization system described in Section 4.

From Equation (1), the pedestrian's posture $\hat{x}$ and its associated error covariance $\boldsymbol{P}$ are predicted using a Kalman filter [24]:

$$\begin{cases} \hat{\boldsymbol{x}}_{(t/t-1)} = \boldsymbol{F}\hat{\boldsymbol{x}}_{(t-1)} \\ \boldsymbol{P}_{(t/t-1)} = \boldsymbol{F}\boldsymbol{P}_{(t-1)}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}_{(t-1)}\boldsymbol{G}^T \end{cases} \tag{3}$$

where $\boldsymbol{Q}$ is the covariance of the plant noise $\Delta\boldsymbol{x}$.

To track multiple pedestrians, as shown in Figure 4(a), a validation region with a constant radius is set around the predicted position $(\hat{x}, \hat{y})$ of each tracked pedestrian. The measurements inside the validation region are considered to be obtained from the tracked pedestrian, and it is applied to the track updated with the Kalman filter. On the other hand, the measurements outside the validation region are considered to be false alarms, and are therefore, discarded. From Equations (2) and (3), the posture of the tracked pedestrian and its associated error covariance are updated by:

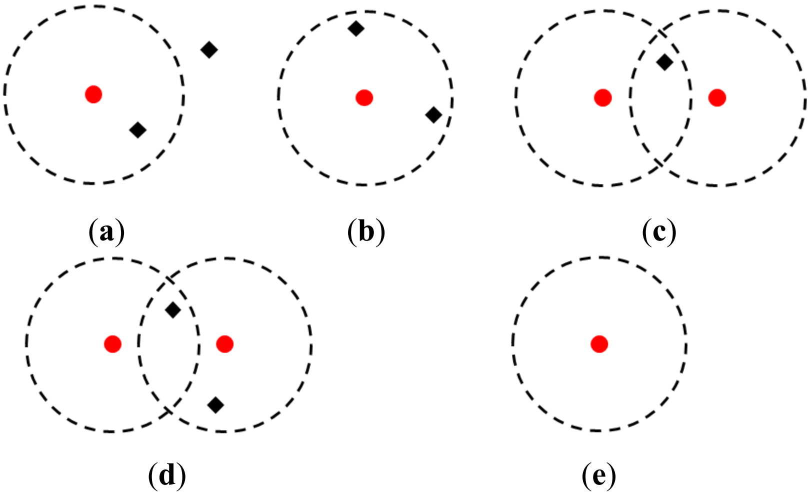
$$\begin{cases} \hat{\boldsymbol{x}}_{(t)} = \hat{\boldsymbol{x}}_{(t/t-1)} + \boldsymbol{K}_{(t)}(\boldsymbol{z}_{(t)} - \boldsymbol{H}_{i\,(t)}\hat{\boldsymbol{x}}_{(t/t-1)} + \boldsymbol{H}'_{i\,(t)}\boldsymbol{u}_{i\,(t)}) \\ \boldsymbol{P}_{(t)} = \boldsymbol{P}_{(t/t-1)} + \boldsymbol{K}_{(t)}\boldsymbol{H}_{i\,(t)}\boldsymbol{P}_{(t/t-1)} \end{cases} \tag{4}$$

where $\boldsymbol{K}_{(t)} = \boldsymbol{P}_{(t/t-1)}\boldsymbol{H}_{i(t)}{}^T\boldsymbol{S}_{(t/t-1)}{}^{-1}$, and $\boldsymbol{S}_{(t/t-1)} = \boldsymbol{H}_{i\,(t)}\boldsymbol{P}_{(t/t-1)}\boldsymbol{H}_{i\,(t)}{}^T + \boldsymbol{R}_{(t)}$. $\boldsymbol{R}$ is the covariance of the measurement noise $\Delta\boldsymbol{z}$.

In our simulation and experiment described in Section 5, the radius of the validation region is set at 1.0 m. The covariances of the plant and measurement noises in Equations (3) and (4) are set at $\boldsymbol{Q}$ = diag (1.0 m^2/s^4, 1.0 m^2/s^4) and $\boldsymbol{R}$ = diag (0.01 m^2, 0.01 m^2), respectively.

In crowded environments, as shown in Figures 4(b–d), multiple measurements exist inside a validation region; multiple tracked pedestrians also compete for measurements. To achieve a reliable data association (matching of tracked pedestrians and measurements), we apply a global-nearest-neighbor (GNN) algorithm [25].

Figure 4. Tracking condition; the red circle and black diamond indicate the tracked pedestrian and the measurement, respectively. The dashed circle indicates the validation region. (**a**) Case 1. (**b**) Case 2. (**c**) Case 3. (**d**) Case 4. (**e**) Case 5.



We consider that, in a validation region, J pedestrians exist and K measurements are received, where J does not necessarily equal K. We then define the distance measure λ_{jk} from the j-th tracked pedestrian to the k-th measurement, where $j = 1, 2, \cdots, J$ and $k = 1, 2, \cdots, K$ as:

$$\lambda_{jk} = \sqrt{(\boldsymbol{z}_{k(t)} - \hat{\boldsymbol{u}}_{j(t/t-1)})^T (\boldsymbol{S}_{j(t/t-1)})^{-1} (\boldsymbol{z}_{k(t)} - \hat{\boldsymbol{u}}_{j(t/t-1)})} \tag{5}$$

where $\boldsymbol{S}_{j(t/t-1)} = \boldsymbol{H}_{j(t)} \boldsymbol{P}_{j(t/t-1)} \boldsymbol{H}_{j(t)}^T + \boldsymbol{R}_{(t)}$, and $\boldsymbol{u}_{j(t/t-1)}$ is the predicted position of the j-th tracked pedestrian.

We then define the following cost matrix Λ:

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1K} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{J1} & \lambda_{J2} & \cdots & \lambda_{JK} \end{pmatrix} \tag{6}$$

We assume that the $a(j)$-th measurement is assigned to the j-th pedestrian. The data association is achieved by finding the $a(j)$ based on the Munkres algorithm [26] so that $\sum_{j=1}^{J} \lambda_{ja(j)}$ can be minimized. It is noted that if the k-th measurement does not exist inside the validation of the j-th tracked person, we set the distance measure at $\lambda_{jk} = \infty$.

Pedestrians always appear in and disappear from the LRS sensing area. They also face interaction and occlusion issues. In order to handle such conditions, we implement a tracking management system based on the following rules:

(a) *Track initiation*: As shown in Figures 4(a–c), the measurements that are not matched with any tracked pedestrians are considered to come from new pedestrians or false alarms, which disappear soon. Therefore, we tentatively initiate tracking of the measurements with Kalman filter. If the measurements are always visible in more than N_1 s, they are considered to come from new pedestrians, and the tracking is continued. If the measurements disappear within N_1 s, they are considered to be the false alarms, and the tentative tracking is terminated.

(b) *Track termination*: When the tracking pedestrians exit the sensing area of the LRS or they meet occlusion, no measurements exist within their validation regions. If no measurements arise from the temporal occlusion, the measurements appear again. We thus predict the positions of the tracking pedestrians with the Kalman filter. If the measurements appear again within N_2 s, we proceed with the tracking. Otherwise (see Figure 4(e)), we terminate the tracking. In our simulation and experiment described in Section 5, we set $N_1 = 1.5$ and $N_2 = 3.0$ by trial and error.

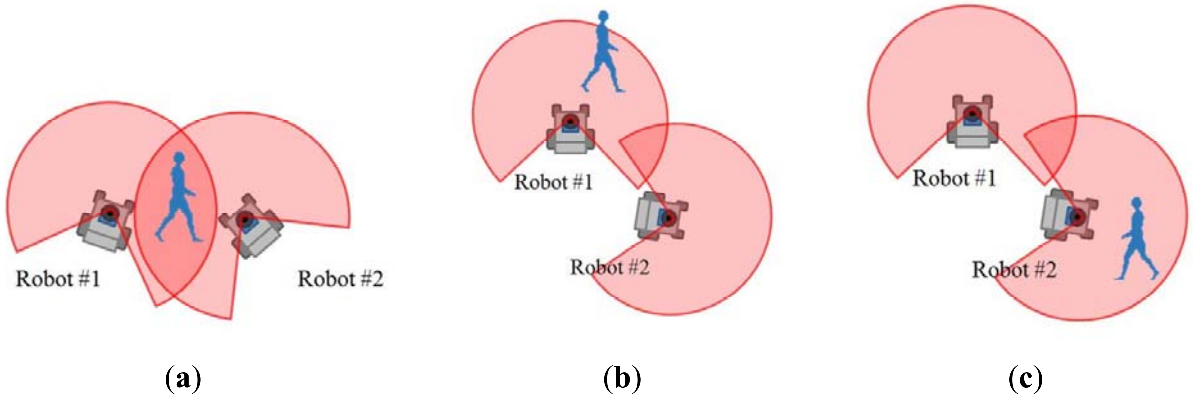
For simplicity, in this paper, pedestrians are assumed to move at an almost constant velocity, and they are tracked using the usual Kalman filter. If the pedestrians move randomly, such as walking, running, going or stopping suddenly, and turning suddenly, using multi-model-based tracking can improve the tracking performance [14,15].

3.3. Cooperative Tracking

When the robots are located near to each other, the tracking mode is switched to cooperative tracking. They communicate with each other and exchange their own tracking data, which consist of estimated positions and velocities of tracked pedestrians and their associated error covariances. Because tracking data are shared, each individual robot constantly tracks pedestrians both inside and outside its own LRS sensing area.

To elucidate cooperative tracking in detail, we consider two robots #1 and #2, as shown in Figure 5. The tracking data for the *m*-th pedestrian tracked by robot #1 is denoted by $\boldsymbol{I}_m^{(1)} = \{\hat{\boldsymbol{x}}_m^{(1)}, \boldsymbol{P}_m^{(1)}\}$, where $m = 1,2,\cdots$. $\hat{\boldsymbol{x}}_m^{(1)}$ denotes the estimate (position estimate $\hat{\boldsymbol{q}}_m^{(1)}$ and velocity estimate $\hat{\dot{\boldsymbol{q}}}_m^{(1)}$); $\boldsymbol{P}_m^{(1)}$ is its associated error covariance. Similarly, the tracking data for the *n*-th pedestrian tracked by robot #2 is denoted by $\boldsymbol{I}_n^{(2)} = \{\hat{\boldsymbol{x}}_n^{(2)}, \boldsymbol{P}_n^{(2)}\}$, where $n = 1,2,\cdots$. We consider that robot #1 combines the tracking data sent from robot #2 with its own tracking data. Combining the tracking data of robot #1 with that of robot #2 can be achieved similarly.

Figure 5. Conditions in cooperative tracking. (**a**) Case 1. (**b**) Case 2. (**c**) Case 3.



(**a**) (**b**) (**c**)

First, we set a validation region with a constant radius around the position estimate $\hat{\boldsymbol{q}}_m^{(1)}$ of the *m*-th pedestrian tracked by robot #1. We consider the position estimate $\hat{\boldsymbol{q}}_n^{(2)}$ of the *n*-th pedestrian tracked by robot #2 as the measurement, and then, we can determine data association (one-to-one matching of pedestrians tracked by robots #1 and #2) using the GNN algorithm. The GNN-based data association in cooperative tracking is similar as one in individual tracking mentioned in Section 3.1. In our simulation and experiment described in Section 5, the radius of the validation region is set at 1.2 m.

As shown in Figure 5(a), when a pedestrian is detected inside the sensing areas of both robots #1 and #2, the two estimates, $\hat{\boldsymbol{q}}^{(1)}$ and $\hat{\boldsymbol{q}}^{(2)}$, of the pedestrian can be matched. For the matched pedestrian, robot #1 updates its own tracking data by the CI method [21]:

$$\begin{cases} \hat{\boldsymbol{x}}_{(t)}^{(1)+} = \boldsymbol{P}_{(t)}^{(1)+}\left\{\omega \boldsymbol{P}_{(t)}^{(1)-1}\hat{\boldsymbol{x}}_{(t)}^{(1)} + (1-\omega)\boldsymbol{P}_{(t)}^{(2)-1}\hat{\boldsymbol{x}}_{(t)}^{(2)}\right\} \\ \boldsymbol{P}_{(t)}^{(1)+} = \left\{\omega \boldsymbol{P}_{(t)}^{(1)-1} + (1-\omega)\boldsymbol{P}_{(t)}^{(2)-1}\right\}^{-1} \end{cases} \tag{7}$$

where $\boldsymbol{I}^{(1)} = \{\hat{\boldsymbol{x}}^{(1)}, \boldsymbol{P}^{(1)}\}$ and $\boldsymbol{I}^{(2)} = \{\hat{\boldsymbol{x}}^{(2)}, \boldsymbol{P}^{(2)}\}$ denote the tracking data of the matched pedestrian. $\hat{\boldsymbol{x}}^{(1)+}$ and $\boldsymbol{P}^{(1)+}$ denote the updated tracking data and its associated error covariance, respectively. The weight ω is selected using the Golden selection search (GSS) method so that the determinant of $\boldsymbol{P}^{(1)+}$ can be minimized under the constraint $0 \le \omega \le 1$. In simulation and experiment described in Section 5, the convergence threshold of the weight ω is set at $1.0*10^{-4}$. In this case, we can determine the appropriate weight ω by iterative calculation of less than twenty times.

As shown in Figure 5(b,c), for non-matched pedestrian, robot #1 updates its own tracking data as follows:

(a) When a pedestrian appears inside the sensing area of robot #1 but outside that of robot #2, as shown in Figure 5(b), robot #1 has the tracking data $\boldsymbol{I}^{(1)}$, but robot #2 does not have $\boldsymbol{I}^{(2)}$. Then, robot #1 sets $\boldsymbol{I}^{(1)+} = \boldsymbol{I}^{(1)}$.

(b) When a pedestrian appears inside the sensing area of robot #2 but outside that of robot #1 as shown in Figure 6(c), robot #2 has the tracking data $\boldsymbol{I}^{(2)}$, but robot #1 does not have $\boldsymbol{I}^{(1)}$. Then, robot #1 sets $\boldsymbol{I}^{(1)+} = \boldsymbol{I}^{(2)}$.

Cooperative tracking with three or more robots can be achieved in a similar manner. Decentralized data fusion provides better system scalability and reliability than centralized data fusion [21]. Therefore, we combine the tracking data in a decentralized manner. Statistically, the tracking data are highly correlated. The conventional Kalman filter-based fusion hampers the development of a decentralized system because it needs to calculate the degree of their correlation. The CI method allows accurate fusion of the tracking data in a decentralized manner without the knowledge of the degree of their correlation. Therefore, we apply the CI algorithm.

Data association is important in pedestrian tracking. In this paper, we apply GNN-based data association to match the current measurement scan to the existing tracks. An alternative effective data association algorithm is multiple hypothesis tracking (MHT) [27,28]. In MHT, the feasible measurement-to-track association hypotheses are enumerated and evaluated up to a certain time depth. The MHT-based data association may outperform the GNN data association in crowded environments; however, in our experience, MHT data association makes real-time tracking difficult in crowded environments because it requires the evaluation of an exponentially increasing number of feasible data association hypotheses. The MHT data association also requires centralized data fusion with a central server [22]. Therefore, we apply GNN data association.

4. Estimation of Robot Posture

To achieve cooperative tracking, each robot must always identify its own posture (position and orientation) with a high degree of accuracy in a world coordinate frame Σ_w and map the tracking data onto Σ_w, for which we apply RTK-GPS. The robot also determines its own posture by a scan matching based localization to improve the accuracy of its posture. If the robot cannot retrieve RTK-GPS information, only the scan matching based localization is applied to determine its own posture.

4.1. RTK-GPS Based Localization

The robot estimates its own velocity (linear/turning velocity) based on dead reckoning using the wheel encoders and gyro. The robot is assumed to move at nearly constant velocity. Motion and measurement models of the *i*-th robot are then given by Equations (8) and (9), respectively:

$$\boldsymbol{V}_i(t) = \boldsymbol{V}_i(t-1) + \Delta\boldsymbol{V}_i(t-1) \tag{8}$$

$$\boldsymbol{z}_i(t) = \begin{pmatrix} 1 & -b/2 \\ 1 & b/2 \\ 0 & 1 \end{pmatrix} \boldsymbol{V}_i(t) + \Delta\boldsymbol{z}_i(t) \tag{9}$$

where $\boldsymbol{V}_i = (v_i, \dot{\psi}_i)^T$; $v_i$is the linear velocity and $\dot{\psi}_i$ is the turning velocity. $\boldsymbol{z}_i = (z_l, z_r, z_\psi)^T$; z_l and z_r are the velocities of the left and right wheels, respectively, measured by the wheel encoders , and z_ψ is the gyro output. $\Delta\boldsymbol{V}_i$ and $\Delta\boldsymbol{z}_i$ are unknown acceleration (disturbance) and the sensor noise, respectively. b is the tread length of the robot.

From Equations (8) and (9), the robot velocity $\boldsymbol{V}_i$ is estimated using Kalman filter. Based on the velocity estimate $\hat{\boldsymbol{V}}_i$, we can determine the posture of the *i*-th robot $\boldsymbol{x}_i = (x_i, y_i, \psi_i)^T$ and its associated covariance by Equations (10) and (11), respectively:

$$\hat{\boldsymbol{x}}_i(t/t-1) = \begin{pmatrix} \hat{x}_i(t-1) + \hat{V}_i(t-1)\tau\cos(\hat{\psi}_i(t-1) + \frac{\hat{\dot{\psi}}_i(t-1)}{2}\tau) \\ \hat{y}_i(t-1) + \hat{V}_i(t-1)\tau\sin(\hat{\psi}_i(t-1) + \frac{\hat{\dot{\psi}}_i(t-1)}{2}\tau) \\ \hat{\psi}_i(t-1) + \hat{\dot{\psi}}_i(t-1)\tau \\ \end{pmatrix} \tag{10}$$

$$\boldsymbol{P}_i(t/t-1) = \nabla\boldsymbol{f}(t-1)\boldsymbol{P}_i(t-1)\nabla\boldsymbol{f}(t-1)^T + \nabla\boldsymbol{f}'(t-1)\boldsymbol{Q}_i(t-1)\nabla\boldsymbol{f}'(t-1)^T \tag{11}$$

where $\hat{\boldsymbol{x}}_i(t/t-1)$ and $\boldsymbol{P}_i(t/t-1)$ are the posture estimate and its error covariance, respectively. $\boldsymbol{Q}_i$ is the error covariance of $\hat{\boldsymbol{V}}_i$. $\nabla\boldsymbol{f}$ and $\nabla\boldsymbol{f}'$ are the Jacobian matrices of Equation (10) at $\hat{\boldsymbol{x}}_i(t/t-1)$ and $\hat{\boldsymbol{V}}_i$, respectively. τ is a sampling period of sensors.

The measurement model related to the RTK-GPS is given by:

$$\begin{aligned} \boldsymbol{z}_{GPS\,(t)} &= \boldsymbol{Hx}_{i\,(t)} + \Delta\boldsymbol{z}_{GPS\,(t)} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \boldsymbol{x}_{i\,(t)} + \Delta\boldsymbol{z}_{GPS\,(t)} \end{aligned} \tag{12}$$

where $\boldsymbol{z}_{GPS}$ is the measurement; position of the *i*-th robot in Σ_w, and $\Delta\boldsymbol{z}_{GPS}$ is the measurement noise.

If the robot obtains posture information from the RTK-GPS, the robot can update its own posture and its associated covariance using Kalman filter as follows:

$$\begin{cases} \hat{\boldsymbol{x}}_{i\,(t)} = \hat{\boldsymbol{x}}_{i\,(t/t-1)} + \boldsymbol{K}_{(t)}[\boldsymbol{z}_{GPS\,(t)} - \boldsymbol{H}\hat{\boldsymbol{x}}_{i\,(t/t-1)}] \\ \boldsymbol{P}_{i\,(t)} = \boldsymbol{P}_{i\,(t/t-1)} - \boldsymbol{K}_{(t)}\boldsymbol{HP}_{i\,(t/t-1)} \end{cases} \tag{13}$$

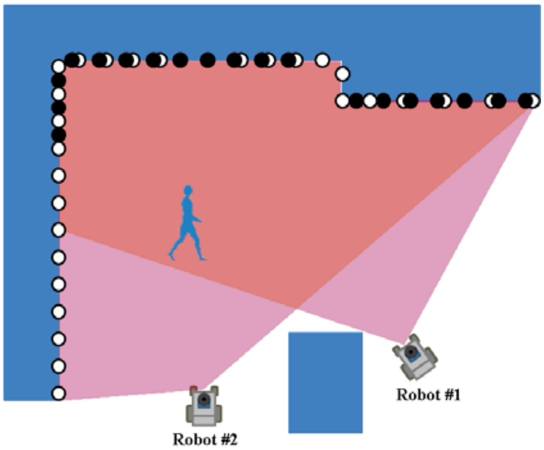
where $\boldsymbol{K}_{(t)} = \boldsymbol{P}_{i\,(t/t-1)}\boldsymbol{H}_{(t)}^{T}(\boldsymbol{HP}_{i\,(t/t-1)}\boldsymbol{H}^{T} + \boldsymbol{R}_{GPS\,(t)})^{-1}$, and $\boldsymbol{R}_{GPS}$ is the covariance of the measurement noise $\Delta\boldsymbol{z}_{GPS}$.

4.2. Scan Matching Based Localization

When multiple robots are located near each other, they have an overlapping sensing area. They improve their own posture accuracy by exchanging their laser-scan images and matching them in their overlapping sensing area.

To elucidate scan matching based localization in detail, we consider two robots #1 and #2, as shown in Figure 6, where the two robots are located near each other and their sensing areas partially overlap. We define the posture of robot #2 relative to robot #1 by ${}^{2}z_{1} = ({}^{2}x_{1}, {}^{2}y_{1}, {}^{2}\psi_{1})^{T}$ in Σ_w. Robot #1 broadcasts its own posture and a laser scan image obtained by its own LRS to robot #2. Robot #2 determines the relative posture, ${}^{2}z_{1}$, by matching its own laser scan image with that sent from robot #1 (Appendix). Hereafter, we call the laser scan matching for estimating relative posture as *relative-scan matching*.

Figure 6. Robot localization by relative-scan matching; the solid and open circles indicate images scanned by robots #1 and #2, respectively.



Measurement model related to the relative-scan matching is given by:

$$
\begin{aligned}
{}^{2}\boldsymbol{z}_{1\,(t)} &= g(\hat{\boldsymbol{x}}_{1\,(t)}, \boldsymbol{x}_{2\,(t)}) + \varDelta^{2}\boldsymbol{z}_{1\,(t)} \\
&= \begin{pmatrix} \cos\psi_{2\,(t)} & -\sin\psi_{2\,(t)} & 0 \\ \sin\psi_{2\,(t)} & \cos\psi_{2\,(t)} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{x}_{1\,(t)} - x_{2\,(t)} \\ \hat{y}_{1\,(t)} - y_{2\,(t)} \\ \hat{\psi}_{1\,(t)} - \psi_{2\,(t)} \end{pmatrix} + \varDelta^{2}\boldsymbol{z}_{1\,(t)}
\end{aligned}
\tag{14}
$$

where $\hat{\boldsymbol{x}}_1$ is the posture of robot #1 estimated by the RTK-GPS-based localization. $\varDelta^{2}\boldsymbol{z}_1$ is the error of the relative posture. From Equations (13) and (14), robot #2 can determine its own posture $\hat{\boldsymbol{x}}_2$ and its associated error covariance $\boldsymbol{P}_2$ using Kalman filter:

$$
\begin{cases}
\hat{\boldsymbol{x}}_{2\,(t)} = \hat{\boldsymbol{x}}_{2\,(t/t-1)} + \boldsymbol{K}_{(t)}[{}^{2}\boldsymbol{z}_{1\,(t)} - \boldsymbol{g}(\hat{\boldsymbol{x}}_{1\,(t)}, \hat{\boldsymbol{x}}_{2\,(t/t-1)})] \\
\boldsymbol{P}_{2\,(t)} = \boldsymbol{P}_{2\,(t/t-1)} - \boldsymbol{K}_{(t)} \nabla\boldsymbol{g}_{(t)} \boldsymbol{P}_{2\,(t/t-1)}
\end{cases}
\tag{15}
$$

where $\boldsymbol{K}_{(t)} = \boldsymbol{P}_{2\,(t/t-1)} \nabla\boldsymbol{g}_{(t)}{}^{T} (\nabla\boldsymbol{g}_{(t)} \boldsymbol{P}_{2\,(t/t-1)} \nabla\boldsymbol{g}_{(t)}{}^{T} + {}^{2}\boldsymbol{R}_{1\,(t)})^{-1}$. $\nabla\boldsymbol{g}$ is the Jacobian matrix of $\boldsymbol{g}$ in Equation (14) at $\hat{\boldsymbol{x}}_{2\,(t/t-1)}$, and ${}^{2}\boldsymbol{R}_1$ is covariance of $\varDelta^{2}\boldsymbol{z}_1$.

5. Simulation and Experimental Results

5.1. Simulation Results

In the experiment described in Section 5.2, it is very difficult to recognize the true positions of the tracked pedestrians. Therefore, we evaluate the performance of the proposed method by simulating four-pedestrian tracking by two robots. As shown in Figure 7, two robots stops at the coordinates $(x, y) = (-8.0, 3.0)$ m and $(7.0, -1.0)$ m, and pedestrians move at the velocity of 0.1–1.7 m/s; pedestrians #1 and #2 move side-by-side at the distance of 0.8 m from start point A, and pedestrians #3 and #4 move side-by-side at the distance of 0.8 m from start point B. Four pedestrians meet each other at point C. In the simulation, the pedestrians are assumed to be always detected correctly, and measurement noise of LRS is assumed to be uniform distribution between −0.05 m and 0.05 m. Simulation tool/software is self-produced using C++ language.

Figure 7. Simulation condition. Red, blue, green and black lines indicate moving path of pedestrians #1, #2, #3 and #4, respectively.

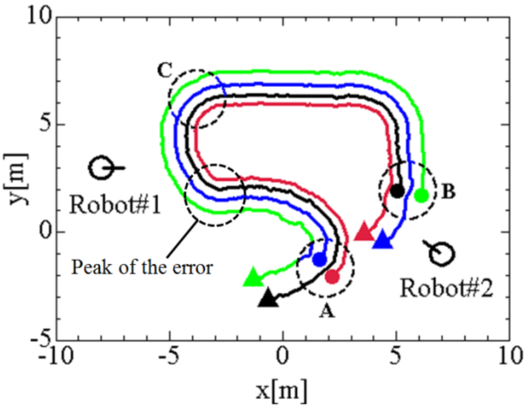
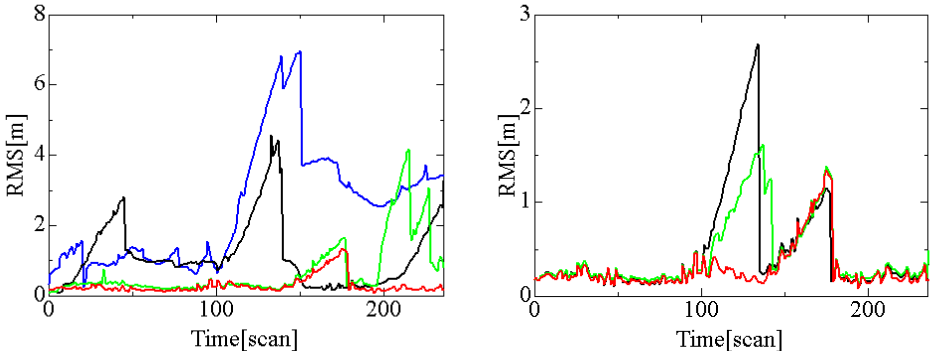


Figure 8(a) shows the effect of tracking mode and data association methods to the tracking error; GNN and conventional nearest neighbor (NN) [24] methods are applied for the data association. Tracking error is evaluated by the following root mean squared error (RMS):

$$RMS_{(t)} = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(\hat{\boldsymbol{u}}_{i\,(t)} - \boldsymbol{u}_{i\,(t)})^T(\hat{\boldsymbol{u}}_{i\,(t)} - \boldsymbol{u}_{i\,(t)})} \tag{16}$$

where $\hat{\boldsymbol{u}}_{i\,(t)} = (\hat{x}_{i\,(t)}, \hat{y}_{i\,(t)})^T$ and $\boldsymbol{u}_{i\,(t)} = (x_{i\,(t)}, y_{i\,(t)})^T$ denote the position estimate and the true position, respectively, of the i-th pedestrian, where i = 1, 2, 3, 4, at the t-th laser scan.

Figure 8. Tracking error. (**a**) Effect of data association method and tracking mode. (**b**) Effect of data fusion method. In (a), red, green, blue and black lines indicate the results by cooperative tracking using GNN, individual tracking using GNN, cooperative tracking using NN, and individual tracking using NN, respectively. In (b), red, green and black lines indicate the results by CI, Kalman filter and averaging method, respectively.



(**a**) (**b**)

First of all, we compare the result of cooperative tracking by GNN data association with that of individual tracking by GNN data association. Both tracking modes have the similar tracking error before 200 scan (20 s); however, individual tracking mode causes large tracking error after 200 scan. Because pedestrian #2 is shadowed by pedestrian #1 around 183 scan, robot #1 loses pedestrian #2 and large tracking error occurs in individual tracking mode after 200scan. However, pedestrian #2 is visible by robot #2 even around 183 scan, and thus cooperative tracking can maintain the accurate tracking after 200scan.

Both tracking modes temporarily cause large tracking error around 160 scan (around (x,y) = (−3.0,0.7)m in Figure 7). This reason is why pedestrian #3 is temporarily shadowed by pedestrian #4 and track lost then occurs.

NN-based data association very often causes incorrect matching of tracked pedestrians and LRS measurements, and this results in the track being lost. On the other side, GNN data association reduces the track lost. Therefore, the tracking error by GNN data association becomes smaller than that by NN data association. As the result, it is clear from Figure 8(a) that cooperative tracking by GNN data association provides better tracking performance than other methods.

Next, we simulate the effect of the data fusion methods for the cooperative tracking to the tracking error. For comparison purpose, we consider three data fusion methods: CI method, Kalman filter, and averaging method. In the Kalman filter, data fusion is achieved by considering the tracking data sent from other robots to be measurements. Based on the averaging method, each robot tracks pedestrians by simply averaging its own tracking data with the tracking data sent from other robots; the averaging method equals CI method by setting the weight ω = 0.5. In this simulation, GNN data association is always applied for the data association.

Figure 8(b) shows the results. The Kalman filter and averaging method cause large tracking errors around 120 scans (around point C in Figure 7). The data fusion method is closely related to the data association method; the performance in data fusion affects that in data association, and vice versa. Compared to Kalman filter and average methods, CI method maintains the accurate tracking performance. From these simulations, we confirmed that cooperative tracking based on CI and GNN methods provides tracking performance better.

5.2. Experimental Results

To evaluate the tracking method, we conducted an experiment in an outdoor environment shown in Figure 9. Three robots and three pedestrians move around in the environment as shown in Figure 10. The moving speed of the robots is less than 0.3 m/s. The walking speeds of pedestrians #1, #2, and #3 are less than 1.5 m/s, 1.5 m/s, and 3.7 m/s, respectively: At first, pedestrian #3 walks at the same speed of pedestrians #1 and #2, and he runs at a speed of 3.7 m/s on the way. The experimental time is 188 scans (18.8 s).

Figure 9. View of experimental environments.

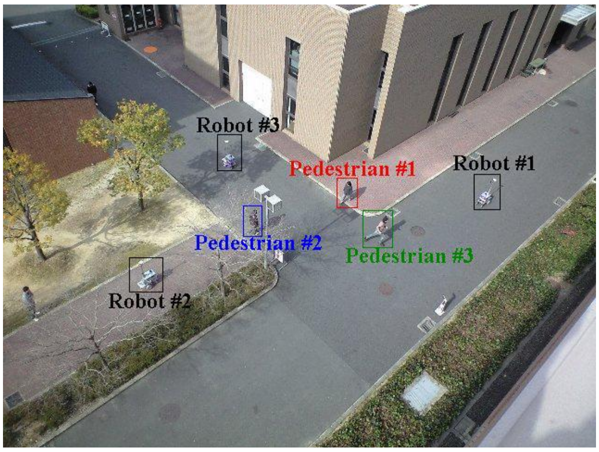


Figure 10. Movement path of robots and pedestrians. (**a**) Robot path. (**b**) Pedestrian path.

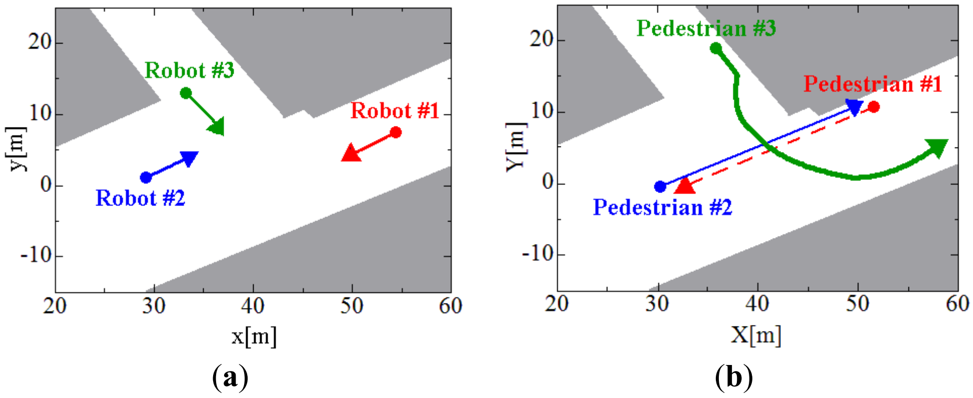


Figure 11 shows the results of pedestrian tracking only by individual tracking; figures (a), (b) and (c) show the tracks of three pedestrians estimated by robot #1, #2, and #3, respectively. Each robot partially tracks pedestrians because the pedestrians exist inside and outside the sensing area of the LRS.

Figure 11. Pedestrian tracks estimated by individual tracking. (**a**) Robot #1. (**b**) Robot #2. (**c**) Robot #3. Red, blue and green lines indicate paths of pedestrians #1, #2 and #3, respectively.

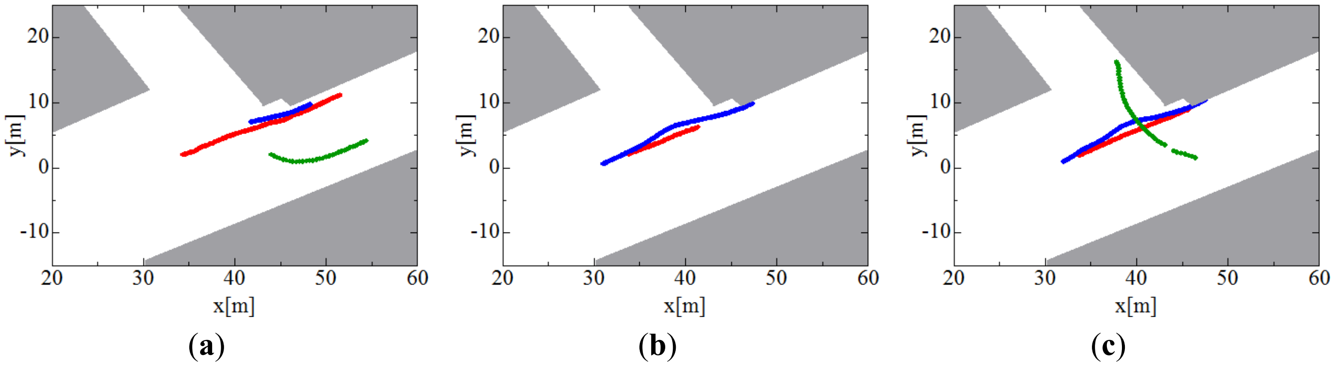


Figure 12 shows the tracks of three pedestrians estimated by individual and cooperative tracking; because the three robots share the tracking data with each other, all three robots can track the three pedestrians for an extended period.

Figure 12. Pedestrian tracks estimated by individual and cooperative tracking. Red, blue and green lines indicate paths of pedestrians #1, #2 and #3, respectively.

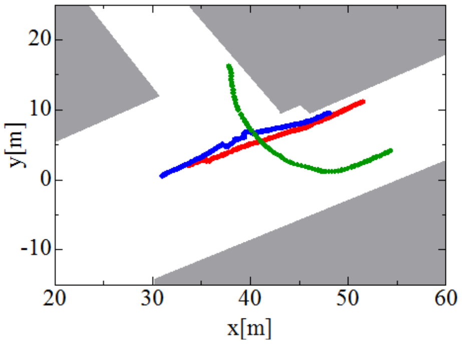
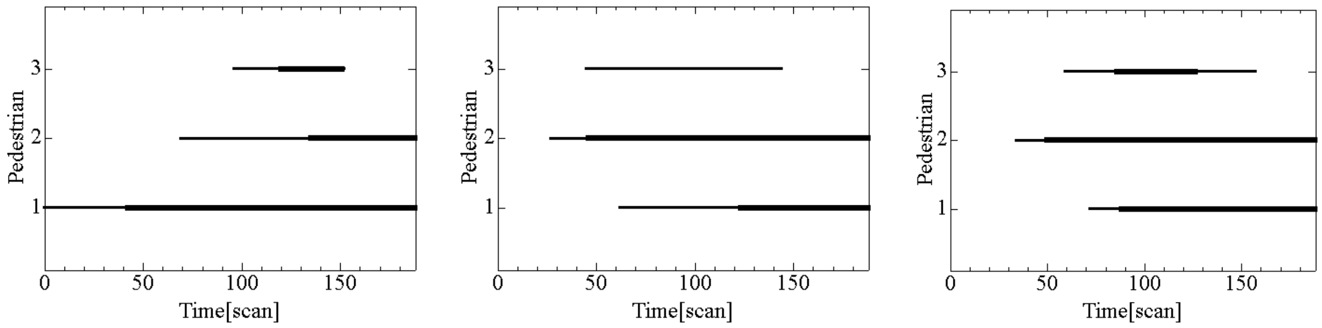


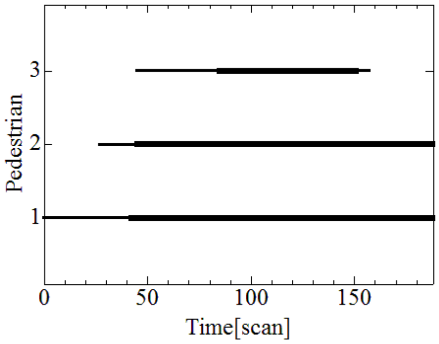
Figure 13 shows the duration of pedestrian tracking; figures (a), (b) and (c) show the times during which robots #1, #2 and #3, respectively, track pedestrians using individual tracking. Figure 14 shows the duration of pedestrian tracking by the individual and cooperative tracking. From these results, cooperative tracking provides a better tracking performance than individual tracking; for example, cooperative tracking detects pedestrian #3 who runs into the road 34 scan (3.4 s) faster than individual tracking. The faster the pedestrians can be detected, the safer becomes robot's navigation.

Figure 13. Duration of individual tracking. (**a**) Robot #1. (**b**) Robot #2. (**c**) Robot #3. The thin line indicates the time during which the pedestrian exits the sensing area of the robot. The bold line indicates the time during which the robot tracks pedestrians using the individual tracking.



(**a**) (**b**) (**c**)

Figure 14. Duration of individual and cooperative tracking. The thin line indicates the time during which the pedestrian exits the sensing area of each robot. The bold line indicates the time during which the robot tracks pedestrians using the individual and cooperative tracking.



6. Conclusions

This paper presents a laser-based pedestrian tracking method using multiple mobile robots. Pedestrians were tracked by each robot using Kalman filter and GNN based data association. The tracking data obtained by each robot was broadcast to others robots and was combined by the CI method. Our method shares the pedestrian tracking data with all robots, and thus, collectively they can always recognize pedestrians that may be invisible to individual robots. The method was validated by simulation and experiment. Our tracking system worked effectively in a decentralized manner without any central server.

In the experiment, three pedestrians were tracked in a sparse environment. We will next conduct pedestrian tracking experiments in crowded environments. To achieve cooperative tracking, the robots must always identify their own postures with a high degree of accuracy in a common coordinate frame, for which, in this paper, we applied two localization methods: RTK-GPS-based and relative-scan matching-based. However, in outdoor environments such as areas surrounded by high buildings and roadside trees, it is difficult for robots to obtain posture information accurately by GPS due to GPS multipath and diffraction problems and so on. To cope with this problem, we will embed a simultaneously localization and mapping (SLAM) method into our tracking system; SLAM-GPS fusion based localization will always maintain a high degree of positioning accuracy, and therefore, it will enhance the robustness of our cooperative pedestrian tracking system in GPS-denied environments such as urban cities.

Acknowledgments

This study was partially supported by Scientific Grant #23560305, Japan Society for the Promotion of Science (JSPS).

References

1. People Detection and Tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop*, Kobe, Japan, 12–17 May 2009.
2. Jia, Z.; Balasuriya, A.; Challa, S. Autonomous Vehicles Navigation with Visual Target Tracking Technical Approaches. *Algorithms* **2008**, *1*, 153–182.
3. Arras, K.O.; Mozos, O.M. Special issue on people detection and tracking. *Int. J. Soc. Robot.* **2010**, *2*, 1–107.
4. Ogawa, T.; Sakai, H.; Suzuki, Y.; Takagi, K.; Morikawa, K. Pedestrian Detection and Tracking using In-Vehicle Lidar for Automotive Application. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, 5–9 June 2011; pp. 734–739.
5. Scholer, F.; Behley, J.; Steinhage, V.; Schulz, D.; Cremers, A.B. Person Tracking in Three-Dimensional Laser Range Data with Explicit Occlusion Adaption. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 9–13 May 2011; pp. 1297–1303.
6. Hashimoto, M.; Ogata, S.; Oba, F.; Murayama, T. A Laser Based Multi-Target Tracking for Mobile Robot. *Intell. Auton. Syst.* **2006**, *9*, 135–144.
7. Sato, S.; Hashimoto, M.; Takita, M.; Takagi, K.; Ogawa; T. Multilayer Lidar-Based Pedestrian Tracking in Urban Environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, 21–24 June 2010; pp. 849–854.
8. Hashimoto, M.; Matsui, Y.; Takahashi, K. Moving-Object Tracking with In-Vehicle Multi-Laser Range Sensors. *J. Robot. Mechatron.* **2008**, *20*, 367–377.
9. Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-Based Multirobot Coordination: A Survey and Analysis. *Proc. IEEE* **2006**, *94*, 1257–1270.
10. Michael, N.; Fink, J.; Kumar, V. Experimental Testbed for Large Multirobot Teams. *IEEE Robot. Autom. Mag.* **2010**, *15*, 53–61.
11. Jung, B.; Sukhatme, G.S. Cooperative Multi-Robot Target Tracking. In *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems*, Minneapolis/St. Paul, MN, USA, 12–14 July 2006; pp. 81–90.
12. Li, Y.; Liu, Y.; Zhang, H.; Wang, H.; Cai, X.; Zhou, D. Distributed Target Tracking with Energy Consideration Using Mobile Sensor Networks. In *Proceeding of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 22–26 September 2008; pp. 3280–3285.

13. La, H.M.; Sheng, W. Adaptive Flocking Control for Dynamic Target Tracking in Mobile Sensor Networks. In *Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 11–15 October 2009; pp. 4843–4848.
14. Hashimoto, M.; Konda, T.; Bai, Z.; Takahashi, K. Laser-Based Tracking of Randomly Moving People in Crowded Environments. In *Proceedings of the IEEE International Conference on Automation and Logistics*, Macau, China, 16–20 August 2010; pp. 31–36.
15. Hashimoto, M.; Bai, Z.; Konda, T.; Takahashi, K. Identification and Tracking Using Laser and Vision of People Maneuvering in Crowded Environments. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Istanbul, Turkey, 10–13 October 2010; pp. 3145–3151.
16. Fod, A.; Howard, A.; Mataric, M.J. A Laser-Based People Tracker. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA, 11–15 May 2002; pp. 3024–3029.
17. Nakamura, K.; Zhao, H.; Shibasaki, R.; Sakamoto, K.; Ooga, T.; Suzukawa, N. Tracking Pedestrians by Using Multiple Laser Range Scanners. In *Proceedings of the ISPRS Congress*, Istanbul, Turkey, 12–23 July 2004; Volume 35, pp.1260–1265.
18. Noguchi, H.; Mori, T.; Matsumoto, T.; Shimosaka, M.; Sato, T. Multiple-Person Tracking by Multiple Cameras and Laser Range Scanners in Indoor Environments. *J. Robot. Mechatron.* **2010**, *22*, 221–229.
19. Bellotto, N.; Hu, H. Multisensor-Based Human Detection and Tracking for Mobile Service Robots. *IEEE Trans. Syst. Man Cybernetics Part B* **2009**, *39*, 167–181.
20. Jung, B.; Sukhatme, G.S. Real-time Motion Tracking from a Mobile Robot. *Int. J. Soc. Robot.* **2010**, *2*, 63–78.
21. Julier, S.; Uhlmann, J.K. *General Decentralized Data Fusion with Covariance Intersection (CI). Handbook of Data Fusion*; Hall, D.L., Llinas, J., Eds.; CRC Press: New York, NY, USA, 2001; pp. 12:1–12:25.
22. Tsokas, N.A.; Kyriakopoulos, K.J. Multi-Robot Multiple Hypothesis Tracking for Pedestrian Tracking. *Auton. Robot.* **2012**, *32*, 63–79.
23. Chou, C.T.; Li, J.Y.; Chang, M.F.; Fu, L.C. Multi-Robot Cooperation Based Human Tracking System Using Laser Range Finder. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 9–13 May 2011; pp. 532–537.
24. Bar-Shalom, Y.; Fortmann, T.E. State Estimation for Linear Systems. In *Tracking and Data Association*; Academic Press, Inc.: San Diego, CA, USA, 1988; pp. 52–122.
25. Konstantinova, P.; Udvarev, A.; Semerdjiev, T. A Study of a Target Tracking Algorithm Using Global Nearest Neighbor Approach. In *Proceedings of the 4th International Conference on Systems and Technologies*, Ruse, Bulgaria, 18–19 June 2003; pp. 290–295.
26. Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q* **1955**, *2*, 83–98.
27. Blackman, S.S. Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE Aerosp. Elect. Syst. Mag.* **2004**, *19*, 6–18.

28. Tsokas, N.A.; Kyriakopoulos, K.J. Multi-Robot Multiple Hypothesis Tracking for Pedestrian Tracking with Detection Uncertainty. In *Proceedings of the 2011 IEEE International Conference on Mediterranean Conference on Control and Automation*, Corfu, Greece, 20–23 June 2011; pp. 315–320.
29. Lu, F.; Milios, E. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *J. Intell. Robot. Syst.* **1997**, *20*, 249–275.

Appendix

Relative-Scan Matching

We determine the posture of robot #2 ${}^2\boldsymbol{z}_1 = ({}^2x_1, {}^2y_1, {}^2\psi_1)^T$ relative to that of robot #1 based on laser-scan matching method. The matching is based on point-to-point scan matching by the iterative closest point (ICP) algorithm [29].

From the laser scan images taken by robots #1 and #2, we compute the relative posture ${}^2\boldsymbol{z}_1$ using the weighted least-squares method; the cost function is given as:

$$J = \sum_{i=1}^{541} w_i \{\boldsymbol{q}_j - (\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{T})\}^2 \tag{A1}$$

where $\boldsymbol{R} = \begin{pmatrix} \cos {}^2\psi_1 & -\sin {}^2\psi_1 \\ \sin {}^2\psi_1 & \cos {}^2\psi_1 \end{pmatrix}$ and $\boldsymbol{T} = \begin{pmatrix} {}^2x_1 \\ {}^2y_1 \end{pmatrix}$.

Here, $\boldsymbol{p}_i = (p_{xi}, p_{yi})^T$, where i = 1, 2, …, 541, denotes the distance sample (scan image) from robot #1; $\boldsymbol{q}_j = (q_{xj}, q_{yj})^T$, where j = 1, 2, … 541, denotes the distance sample from robot #2, as shown in Figure A1.

Each sample $\boldsymbol{p}_i$ corresponds to the minimum distance sample $\boldsymbol{q}_j$ of all samples in the scan by robot #2; w_i denotes the weight. $\boldsymbol{R}$ and $\boldsymbol{T}$ denote the rotational matrix and translational vector, respectively.

To reduce the effects of correspondence errors in the distance samples in the two laser images, we define weight w_i according to the errors between correspondence points $w_i = (1 + d_{ij} / C)^{-1}$, where d_{ij} denotes the distance error between the two laser images and C is a constant. In our experiment described in Section 5, C is set at 0.1.

From Equation (A1), the iterative least-squares method is used to update the relative posture ${}^2\boldsymbol{z}_1^{(m-1)}$ as follows:

$${}^2\boldsymbol{z}_1^{(m)} = {}^2\boldsymbol{z}_1^{(m-1)} + (\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{W}(\boldsymbol{q} - \boldsymbol{p}) \tag{A2}$$

where $\boldsymbol{p} = (\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \ldots, \boldsymbol{p}_{541}^T)^T, \boldsymbol{q} = (\boldsymbol{q}_1^T, \boldsymbol{q}_2^T, \ldots, \boldsymbol{q}_{541}^T)^T, \boldsymbol{W} = \mathrm{diag}(w_1, w_2, \ldots, w_{541})$ and $\boldsymbol{H} = \partial \boldsymbol{p} / \partial {}^2\boldsymbol{z}_1$.

The convergent value of ${}^2\boldsymbol{z}_1^{(m)}$ gives the relative posture ${}^2\boldsymbol{z}_1$.

Figure A1. Relative-scan matching. Solid and open circles indicate scan images taken by robots #1 and #2, respectively.