Cluster-Based SJPDAFs for Classification and Tracking of Multiple Moving Objects

Naotaka Hatao, and Satoshi Kagami

Abstract This paper describes a method for classifying and tracking multiple moving objects with a laser range finder (LRF). As moving objects are tracked in the framework of sample-based joint probabilistic data association filters (SJPDAFs), the proposed method is robust against occlusions or false segmentation of LRF scans. It divides tracking targets and corresponding LRF segments into clusters and able to classify each cluster as a car or a group of pedestrians. In addition, it can correct false segmentation of LRF scans. We implemented the proposed method and obtained experimental results demonstrating its effectiveness in outdoor environments and crowded indoor environments.

1 Introduction

This paper describes a method using a horizontal laser range finder (LRF) to track and classify multiple moving objects. Mobile robots need to detect and track moving objects, and moving object tracking can also be used in traffic stream measuring, security systems, and so on.

It is generally difficult to identify individual moving objects from horizontal LRF scans in crowded areas because the shapes of objects are not static, and some moving objects are occluded by other moving objects. Multiple hypothesis approaches are often efficient at coping with these uncertainties. Joint probabilistic data association filters (JPDAFs) [1] and multiple hypothesis tracking (MHT) are well-known tracking algorithms using multiple hypothesis methods [2], and several modifications of these algorithms have been proposed to improve tracking performance in cluttered environments [3][4][5]. These algorithms enumerate the correspondences between features extracted from sensor data and moving objects and build hypotheses using

Naotaka Hatao
National Inst. of AIST, 2-3-26 Aomi, Koto-ku, Tokyo, Japan e-mail: n.hatao@aist.go.jp

sets of these correspondences. As these algorithms can handle false positives and negatives, they are able to track moving objects reliably even in noisy environments.

Particle filters are robust methods for tracking moving objects, but normal particle filters are unsuitable for tracking multiple targets because particles easily become gathered around a single target. Although several methods have been proposed to overcome this problem [6][7], they do not have a mechanism to treat false positives or negatives. Schulz et al. proposed the use of sample-based JPDAFs (SJPDAFs), which use particle filters instead of Kalman filters used in normal JPDAFs [8].

It is important to identify categories of moving objects. Because the shapes of LRF scans corresponding to moving objects are not stable, many researchers have used time-series LRF scans to classify such objects [9][10][11]. Their methods, however, assume that pedestrians in a group have a same velocity vector and do not explicitly treat the merging and separation of groups of moving objects.

Candidates of moving objects are extracted from LRF scans by dividing the scan points into segments, and segmenting LRF scans correctly is also a difficult problem. If the gap between adjacent LRF scan points exceeds a threshold distance, it becomes a boundary of segments.Segments, however, do not always correspond one-to-one with moving objects. For example, pedestrians often walk so close to each other that an LRF cannot measure the space between them. And LRF scans corresponding to a car are often divided into several parts.

In this paper, we propose an classification and tracking method based on SJPDAFs. SJPDAFs are robust against false positives and negatives, and they make it possible to flexibly design individual trackers using particle filters. The proposed method has two advantages to track moving objects effectively in the real world. The first is that it generates additional hypotheses to cope with the above-mentioned false segmentation. The second is that it divides tracking targets and corresponding LRF segments into clusters and classify each cluster as a car or a group of pedestrians. The numbers of pedestrians in clusters are also estimated, and each pedestrian is tracked individually. Individual tracking enables the merging or splitting of clusters.

This paper is organized as follows. An overview of the proposed method is presented in Section 2, and the methods for detecting moving object candidates are described in Section 3. The proposed method is described in greater detail in Section 4, experimental results are presented in Section 5, and a conclusion is given in Section 6.

2 Multiple Moving Object Tracking using SJPDAFs

2.1 Overview of SJPDAFs

This section briefly introduces the concept of SJPDAFs. Suppose that n_k moving objects are being tracked and m_k features are measured at time k. For convenience we let i denote the ID number of a moving object and let j denote the ID number of a

Table 1 Example hypotheses: the case of two moving objects and three features

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	θ_{11}	θ_{12}	θ_{13}
$\boldsymbol{x}_1^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_2(k)$	$\boldsymbol{z}_2(k)$	$\boldsymbol{z}_2(k)$	$\boldsymbol{z}_3(k)$	$\boldsymbol{z}_3(k)$	$\boldsymbol{z}_3(k)$
$\boldsymbol{x}_2^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_2(k)$	$\boldsymbol{z}_3(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_2(k)$	$\boldsymbol{z}_3(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_3(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_2(k)$
$m_k - \lvert\theta\rvert$	3	2	2	2	2	1	1	2	1	1	2	1	1

feature. $\boldsymbol{X}^k = \{\boldsymbol{x}_1^k, \cdots, \boldsymbol{x}_i^k, \cdots, \boldsymbol{x}_{n_k}^k\}$ and $\boldsymbol{Z}(k) = \{\boldsymbol{z}_1(k), \cdots, \boldsymbol{z}_j(k), \cdots, \boldsymbol{z}_{m_k}(k)\}$ denote the state vectors of moving objects and the features, respectively. In addition, $\boldsymbol{Z}^k$ denotes the sequence of all features up to time k.

As mentioned above, SJPDAFs use particle filters to estimate the states of moving objects. SJPDAFs deploy N particles per moving object. The prediction and resampling steps of SJPDAFs are the same as those of normal particle filters, but the likelihoods of particles are calculated by the following expressions.

$$\omega_{i,n}^k = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\boldsymbol{z}_j(k) | \boldsymbol{x}_{i,n}^k) \tag{1}$$

$\boldsymbol{x}_{i,n}^k$ and $\omega_{i,n}^k$ denote the state vector and the likelihood of the nth particle in ith particle set at time k, respectively. β_{ji} is calculated as follows:

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} [P(\theta | \boldsymbol{Z}^k)] \tag{2}$$

where

$$P(\theta | \boldsymbol{Z}^k) = \alpha \gamma^{(m_k - |\theta|)} \prod_{(j,i) \in \theta} \frac{1}{N} \sum_{n=1}^{N} p(\boldsymbol{z}_j(k) | \boldsymbol{x}_{i,n}^k). \tag{3}$$

θ and $P(\theta | \boldsymbol{Z}^k)$ respectively denote a hypothesis and its likelihood. γ denotes the probability that an observed feature is a false alarm (false positive), and $m_k - |\theta|$ denotes the number of false alarms in θ.Θ_{ji} denotes the set of all hypotheses for which $\boldsymbol{z}_j$ corresponds to $\boldsymbol{x}_i$, and α represents a normalizer.

We assume that two objects are being tracked and three features are measured. All possible hypotheses are listed in Table 1, where $\boldsymbol{z}_0(k)$ indicates a false negative. For example, in θ_5, $\boldsymbol{x}_1$ corresponds to $\boldsymbol{z}_1(k)$ and no feature that corresponds to $\boldsymbol{x}_2$ is found. Also, in this example, Θ_{12} equals $\{\theta_2, \theta_9, \theta_{12}\}$.

The original implementation of SJPDAFs uses local minima of LRF scans as the features. It uses occupancy grid maps to calculate $p(\boldsymbol{z}_j(k) | \boldsymbol{x}_{i,n}^k)$, whereas the proposed method uses the shapes of contours of moving objects. See Sections 4.2 and 4.3 for details.

A major weakness of (S)JPDAFs is that they cannot estimate the number of moving objects, and the number of moving objects is estimated in a separate process. In the original implementation of SJPDAFs, it is assumed that the change in the number of moving objects follows a Poisson process, and the number of moving objects is calculated as follows:

$$
\begin{aligned}
P(N^k|\boldsymbol{M}^k) &= \alpha \cdot P(m_k|N^k,\boldsymbol{M}^k) \cdot P(N^k|\boldsymbol{M}^{k-1}) \\
&= \alpha \cdot P(m_k|N^k) \cdot \sum_n [P(N^k|N^{k-1}=n) \cdot P(N^{k-1}=n|\boldsymbol{M}^{k-1})] \qquad (4)
\end{aligned}
$$

N^k denotes the number of moving objects at time k, and $\boldsymbol{M}^k$ denotes the sequence of the number of features up to time k.

2.2 Cluster-Based SJPDAFs with Classification

This section describes the overview of the proposed method. The proposed method works both on LRFs fixed on the ground and on LRFs mounted on mobile robots.

In the original work on SJPDAFs, LRFs were mounted at a height of 40 cm and detected the legs of persons. In the proposed method, in contrast, LRFs are placed at about the height of a person's chest. This is because the detection of small objects such as legs at a large distance is unstable. In addition, if the LRFs on outdoor mobile robots are mounted low, the laser beams often hit the ground.

The original implementation of SJPDAFs estimates the number of people within the entire sensor area by assuming a Poisson process. In the real world, however, the areal density of moving objects varies considerably, thus a Poisson process is not suitable for representing the change in the number of moving objects. The proposed method therefore uses cluster-based SJPDAFs. A "cluster" in this paper means a set of SJPDAF particle filter components and corresponding LRF segments. "Segment" means a set of LRF scan points, and the boundaries of segments are placed where the gaps between adjacent scan points are large. Estimation of the number of pedestrians and classification are performed for each cluster.

The proposed tracking method is performed as follows:

1. Moving object candidates are extracted from the latest LRF scan. Each candidate is an LRF segment. If the "Extended Trajectories" method described in Section 3 is used, initial grouping of segments is performed.
2. Particles in existing clusters are updated according to $p(\boldsymbol{x}_{i,n}^k|\boldsymbol{x}_{i,n}^{k-1})$, and corresponding segments are enumerated.
3. Merging and splitting of clusters are performed. If two particle filters belonging to different clusters share a corresponding segment or grouped segment, those clusters are merged. In contrast, if a set of particle filters no longer shares corresponding segments with the remaining particle filters in the same cluster, the set of particle filters is split as a new cluster. (see Section 4.4 for details)
4. Each cluster is updated independently. (see Sections 4.2 and 4.3 for details)
5. Classification and estimation of the number of pedestrians are performed for each cluster. (see Section 4.1 for details)
6. New clusters are initialized for segments that are not associated with existing clusters.

3 Extraction of Candidate Moving Objects and Initial Velocity Estimation

Before moving objects can be tracked, candidate moving objects must be extracted from LRF scans. Three extraction methods are implemented in the proposed method.

1. **Occupancy grid map with a polar coordinate system** for fixed LRFs
2. **Occupancy grid map with a Cartesian coordinate system** for indoor robots
3. **Extended Trajectories** for outdoor robots

Extraction of moving object candidates using occupancy grid map is a common method. Grid maps are generated in advance before starting tracking. Each occupancy grid cell has one of three states: "free," "occupied," or "unknown." "Occupied" means that there are static objects in the grid cell, whereas "free" means that there are no objects in the grid cell. If LRF scans appear in free grid cells, those scans might be associated with moving objects.

If the LRFs are fixed in the ground, grid maps with polar coordinate systems are selected. The origins of the polar coordinate systems are same as the measurement origins of the LRFs. This method has the lowest computational cost. If the LRFs are mounted on robots that move in indoor environments (i.e., they can estimate their coordinates precisely), grid maps with a Cartesian coordinate systems are selected.

If, on the other hand, the LRFs are mounted on robots that move outdoor environments, the "extended trajectories" method described below is used. This method has two advantages. The first is that it can estimate velocities. Since there are fast-moving objects in outdoor environments, tracking often fails if the initial velocity vectors follow a zero-centered Gaussian distribution. The second advantage is that the proposed method can perform initial grouping of LRF segments.

The "Extended Trajectories" extraction method also uses a grid map in which each grid cell has one of three states: "Unknown", "Foreground", "Background". LRF scans are transformed to global coordinates using the odometry of the robot. Candidate moving objects at time k are extracted according to the following procedure.

1. Grids within L_1 mm of LRF scan points from time $k-t_1$ to time k are set as "Foreground."
2. Grids within L_2 mm of LRF scan points from time $k-t_2$ to time $k-t_1$ are set as "Background." Grids that are already set as Foreground are overwritten.
3. Foreground regions adjacent to Background regions are enumerated. Isolated foreground regions derive from false positives.
4. If an LRF scan segment at time k is on a Foreground region enumerated in step 3, it is extracted as a candidate moving object.

Fig. 1 shows the results of moving object initialization. This procedure builds "extended" trajectories of moving objects. That is, if there are moving objects, foreground regions growing on background regions appear.

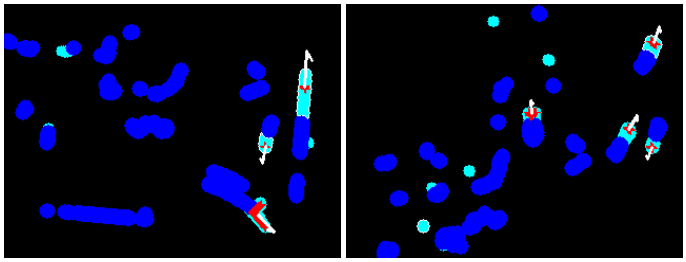

Fig. 1 Results of moving object initialization. Blue and aqua regions respectively indicate "Background" and "Foreground" regions. Red points and white arrows respectively indicate extracted candidate moving objects and their velocities.

The velocity vector of the jth candidate is estimated using the following expressions.

$$|\overline{v_j}| = \frac{\sqrt{(\overline{x_{o_j}} - \overline{x_{b_j}})^2 + (\overline{y_{o_j}} - \overline{y_{b_j}})^2} + L_2}{t_1} \tag{5}$$

$$\overline{\theta_j} = \arctan((\overline{y_{o_j}} - \overline{y_{b_j}}), (\overline{x_{o_j}} - \overline{x_{b_j}})) \tag{6}$$

$\overline{\boldsymbol{x}_{o_j}} = (\overline{x_{o_j}}, \overline{y_{o_j}})$ denotes the centroid of LRF scan points in the corresponding foreground region, and $\overline{\boldsymbol{x}_{b_j}} = (\overline{x_{b_j}}, \overline{y_{b_j}})$ denotes the centroid of border grids between the foreground and background regions. $|\overline{v_j}|$ and $\overline{\theta_j}$ respectively denote the estimated values of the velocity and angle of the jth candidate. In Eq.(5), the moving distances are approximated by sums of L_2 (the size of expansion) and the distances between centroids of border grids and the centroids of the current LRF scan points.

These estimated values are used for initialization of particles in SJPDAFs. If there are two background regions adjacent to the corresponding foreground region, two estimated values of velocity vectors are generated. In this case, particles are divided into two groups and each group is given a separate velocity vector.

Initial grouping of LRF segments is performed using extended trajectories. Segments in the same foreground region belong to the same group. In this process, separated LRF segments associated with a car belong to the same group.

4 Moving Object Classification and Tracking Classification using Cluster-Based SJPDAFs

The proposed method divides LRF segments and particle filters into several clusters, and classification and tracking are performed independently for each cluster.

After classification of each cluster is performed, corresponding particle filters are applied. The proposed method assumes that LRF scans corresponding to pedestrians form cylindrical shapes and those corresponding to cars form rectangular shapes. It also uses different methods to build hypotheses for pedestrians and cars.

Table 2 Class definition of SVM

c_0	c_1	c_2	c_3	$\cdots$
false positive	one car	one pedestrian	two pedestrians	$\cdots$

4.1 Classification and Estimation of Number of Moving Objects

The proposed method performs classification and estimation of the number of pedestrian using a support vector machine (SVM). As the shapes of LRF scan segments are not stable, the method uses a time-series estimation.

The class definition of moving objects is shown in Table 2, where c_k indicates the label of each class.

We define the feature vector of LRF scans in a cluster at time t as $\boldsymbol{z}_f(t)$ and define a set of feature vectors from time 0 to t as $Z_f^t = \{\boldsymbol{z}_f(0) \cdots \boldsymbol{z}_f(t)\}$. The value we want to estimate is $P(c_n|Z_f^t)$, and using the same assumption as in Eq.(4) we obtain

$$P(c_k(t)|Z_f^t) = \alpha \cdot P(z_f(t)|c_k(t)) \cdot P(c_k(t)|Z_f^{t-1}) \tag{7}$$

$$P(c_k(t)|Z_f^{t-1}) = \sum_n [P(c_k(t)|c_k(t-1) = n) \cdot P(c_k(t-1) = n|Z_f^{t-1})] \tag{8}$$

And from Bayes' theorem we obtain

$$P(\boldsymbol{z}_f(t)|c_k) = \alpha \frac{P(c_k|\boldsymbol{z}_f(t))}{P(c_k)} \tag{9}$$

$P(c_k|\boldsymbol{z}_f(t))$ can be estimated using an SVM, and $P(c_k)$ can be estimated using the SVM training sets. The features for the SVM are defined as followed:

z_{f0} : Number of LRF segments
z_{f1} : Sum of lengths of LRF segments
z_{f2} : Average speed
z_{f3} : Difference between angle of directed bounding box and angle of average velocity vector
z_{f4} : Length of long side of directed bounding box
z_{f5} : Length of short side of directed bounding box
z_{f6} : Residual error between directed bounding box and LRF scan points

Once a cluster is classified as a "False Positive," all corresponding particle filters are removed and tracking finishes. If the cluster class changes from "car" to "pedestrian(s)" or from "pedestrian(s)" to "car", new particle filters are deployed. In these cases, old particle filters are not removed immediately and continue to be tracked in preparation for false classifications. The average velocity vector of old particle filters is carried on as the initial velocity vector of new particle filters.

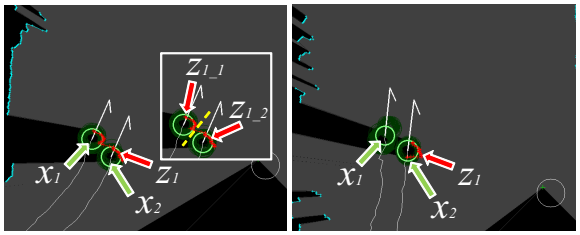

Fig. 2 Results of tracking a group of pedestrians. Dark green circles denote individual particles. Light green circles and white arrows denote results of particle filters (position, radius and velocity). White circles denote the robot, and black regions denote occluded areas.

Table 3 Hypothesis likelihood results for Fig. 2 left

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
$\boldsymbol{x}_1^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_{1-1}(k)$	$\boldsymbol{z}_{1-2}(k)$	$\boldsymbol{z}_{1-1}(k)$	$\boldsymbol{z}_{1-2}(k)$
$\boldsymbol{x}_2^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_{1-1}(k)$	$\boldsymbol{z}_{1-2}(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_{1-2}(k)$	$\boldsymbol{z}_{1-1}(k)$
$P(\theta\|Z^k)$	0.0054	8.7e-7	1.1e-9	0.027	3.2e-15	0.16	6.3e-21	0.80	1.2e-27

Table 4 Hypothesis likelihood results for Fig. 2 right

	θ_1	θ_2	θ_3
$\boldsymbol{x}_1^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$
$\boldsymbol{x}_2^k$	$\boldsymbol{z}_0(k)$	$\boldsymbol{z}_1(k)$	$\boldsymbol{z}_0(k)$
$P(\theta\|Z^k)$	2.8e-9	0.93	0.068

4.2 Tracking and Hypothesis Building for Pedestrians

The proposed method assumes that pedestrians have a cylindrical shape. The nth particle for the ith pedestrian at time k is defined as follows:

$$\boldsymbol{x}_{i,n}^{\boldsymbol{k}} = (x_{i,n}^k, y_{i,n}^k, v_{i,n}^k, \theta_{i,n}^k, r_{i,n}^k)^T \tag{10}$$

where $v_{i,n}^k$, $\theta_{i,n}^k$ and $r_{i,n}^k$ respectively denote the velocity, direction, and radius of the cylinder. The likelihood of a particle is calculated using sum of the distances from LRF scan points in a corresponding segment to the circumference of the particle.

$$p(z_j(k)|x_{i,n}^k) = \frac{1}{\sqrt{2\pi\sigma_p}} \exp\left(-\frac{\sum_{l=0}^{m_j^k}(d_{i,n,l}^k)^2}{2m_j^k\sigma_p^2}\right) \tag{11}$$

$$d_{i,n,l}^k = |\hat{x}_{i,n}^k - z_{j,l}(k)| - r_{i,n}^k \tag{12}$$

$z_{j,l}(k)$ denotes the lth LRF scan point in the jth LRF segment at time k, m_k denotes the number of LRF scans in the corresponding segment, and $\hat{x}_{i,n}^k = (x_{i,n}^k, y_{i,n}^k)$ denotes the position of the nth particle. σ_p is a constant.

When several pedestrians close together are moving, segmentation of a LRF scan often fails and large LRF segments are obtained. To cope with this problem, the method generates hypotheses in which large segments are divided into several small segments. How to divide the segments is decided on the basis of a single-step result

of the SVM $P(\boldsymbol{z}_f(t)|c_k)$ described in Sections 4.1. For example, Hypothesis generation results corresponding to the left side of Fig. 2 are listed in Table 3. In this case, although only one segment is detected, the class that has the largest $P(\boldsymbol{z}_f(t)|c_k)$ is c_3 (two pedestrians). Thus, hypotheses in which the segment is divided into two segments are built. $\boldsymbol{z}_{1-1}(k)$ and $\boldsymbol{z}_{1-2}(k)$ denote divided segments. The most reliable hypothesis is θ_8, in which each tracker corresponds to divided segments.

Hypothesis generation results corresponding to the right side of Fig. 2 are listed in Table 4. The class that has the largest $P(\boldsymbol{z}_f(t)|c_k)$ is c_2 (one pedestrian) and dividing of segments does not occur. In this case the left pedestrian is hidden by the right pedestrian, and the most reliable hypothesisθ_2 supports this status.

4.3 Tracking and Hypothesis Building for Cars

There are two types of LRF scans corresponding to cars. One is associated with the car body, and the other is associated with the window frames. Both, however, are approximately rectangular.

Thus the *n*th particle for the *i*th car at time *k* is defined as follows:

$$\boldsymbol{x}_{i,n}^{\boldsymbol{k}} = (x_{i,n}^k, y_{i,n}^k, v_{i,n}^k, \theta_{i,n}^k, Ll_{i,n}^k, Lw_{i,n}^k)^T \tag{13}$$

where $Ll_{i,n}^k$ and $Lw_{i,n}^k$ respectively denote the lengths of the long and short sides of the rectangle.

$$p(z_j(k)|x_{i,n}^k) = \frac{1}{\sqrt{2\pi\sigma_c}} \exp(-\frac{\sum_{l=0}^{m_j^k}(d_{i,n,l}^k)^2}{2m_j^k\sigma_c^2}) \tag{14}$$

$$d_{i,n,l}^k = \min_{x_l \in L_{i,n}^k} |z_{j,l}(k) - x_l| \tag{15}$$

$L_{i,n}^k$ denotes the rectangle constructed using $x_{i,n}^k$. σ_c is a constant.

Unlike LRF scan segments corresponding to pedestrians, those corresponding to cars are often separated. The proposed method therefore generates hypotheses in which several segments correspond to one tracking target. Fig. 3 shows a result of car tracking. Although the shapes of individual LRF segments change dynamically, the overall shape remains rectangular. Hypothesis generation results corresponding to Fig. 3(1) and Fig. 3(2) are listed in Table 5 and Table 6. (where hypotheses that have small likelihoods are omitted). θ_{17} in Table 5, for instance, means that $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, and $\boldsymbol{z}_4$ correspond to the car, and $\boldsymbol{z}_3$ and $\boldsymbol{z}_5$ are false positives. The most reliable hypothesis Table 5 is θ_{28}, in which $\boldsymbol{z}_3$(the scan segment corresponding to the driver of the car) is a false positive. The most reliable hypothesis in Table 6, on the other hand, is θ_{14}, in which all segments are obtained from the car.

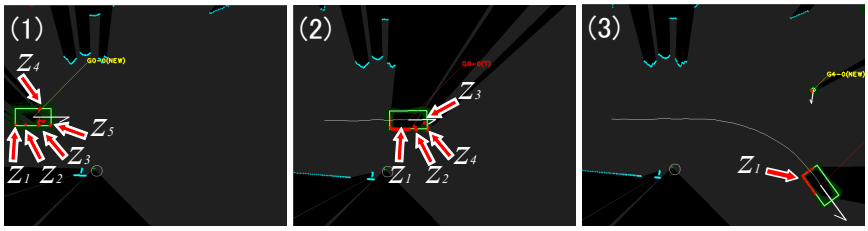

Fig. 3 Result of tracking a car. The car was climbing up a ramp, and LRF firstly detected the window frames, and then detected the body. Dark green squares denote individual particles. Light green squares and white arrows denote results of particle filters (position, angle, size of rectangle, and velocity)

.

Table 5 Example of hypotheses for a car (Fig. 3(1))

	θ_{17}	θ_{18}	θ_{21}	θ_{24}	θ_{28}	θ_{29}	θ_{31}
$\boldsymbol{x}_1^k$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_4$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_5$	$\boldsymbol{z}_1,\boldsymbol{z}_4,\boldsymbol{z}_5$	$\boldsymbol{z}_2,\boldsymbol{z}_4,\boldsymbol{z}_5$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_4,\boldsymbol{z}_5$	$\boldsymbol{z}_1,\boldsymbol{z}_3,\boldsymbol{z}_4,\boldsymbol{z}_5$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3,\boldsymbol{z}_4,\boldsymbol{z}_5$
$P(\theta\|Z^k)$	0.011	0.019	0.022	0.0071	0.63	0.048	0.24

Table 6 Example of hypotheses for a car (Fig. 3(2))

	θ_7	θ_9	θ_{10}	θ_{12}	θ_{13}	θ_{14}	θ_{15}
$\boldsymbol{x}_1^k$	$\boldsymbol{z}_1,\boldsymbol{z}_4$	$\boldsymbol{z}_2,\boldsymbol{z}_4$	$\boldsymbol{z}_3,\boldsymbol{z}_4$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_4$	$\boldsymbol{z}_1,\boldsymbol{z}_3,\boldsymbol{z}_4$	$\boldsymbol{z}_2,\boldsymbol{z}_3,\boldsymbol{z}_4$	$\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3,\boldsymbol{z}_4$
$P(\theta\|Z^k)$	3.7e-4	3.1e-4	2.7e-4	0.016	0.018	1.5e-4	0.96

4.4 Merging and Splitting of Clusters

If a set of particle filters no longer shares corresponding segments with the remaining particle filters in the same cluster, it is split as a new cluster. In contrast, if two or more particle filters belonging to different clusters share a corresponding segment or grouped segment, these clusters are merged. The proposed method uses two types of merging: temporary merging and permanent merging.

If one or both of the clusters is classified as a car or they have different velocity vectors, temporary merging is used (Fig. 4). In this case, the hypotheses for SJPDAFs are created using all LRF segments and particle filters that belong to both clusters. In the current implementation, the $P(c_n|Z_f^t)$ values for clusters that are temporarily merged are updated individually. Corresponding LRF scan segments are selected based on the most reliable SJPDAF hypothesis. If one cluster departs from the other again and the "Foreground region" is divided into two, temporary merging is terminated. Using this framework, the proposed methods can treat clusters that contain both cars and pedestrians.

If both approaching clusters are classified as pedestrian groups and the differences between their average speeds and the directions of their velocity vectors are below the thresholds, permanent merging is used (Fig. 5). In this case, the $P(c_n|Z_f^t)$ values of both clusters are integrated.

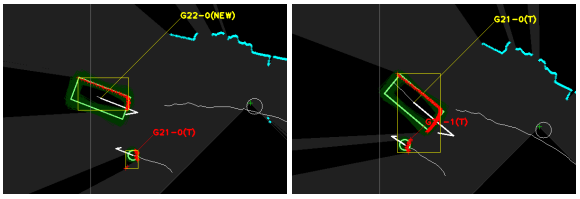

Fig. 4 Example of temporary merging: G21 (a pedestrian) and G22 (a car) were temporarily merged.

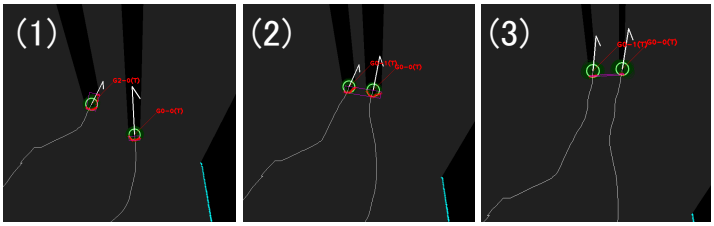

Fig. 5 Example of permanent merging: G0 (a pedestrian) and G2 (a pedestrian) were merged permanently because they had almost same velocity vectors.

Table 7 Numbers and frames of training data set

class	c_0	c_1	c_2	c_3	c_4	c_5	c_6
	false positive	car	1 person	2 persons	3 persons	4 persons	5 or more persons
numbers	212	18	318	54	9	5	3
frames	868	899	20462	3631	352	267	58

Table 8 Results of tracking in an outdoor environment: Per-object accuracies

Target	Succeeded	Disrupted in occ.	Failed	Mismatch	FP
Pedestrian	89	16	2	4	41
Car	7	0	0	0	2

5 Classification and Tracking Experiments and Evaluations

5.1 Classification and Tracking Experiments in an Outdoor Environment

This section describes experimental results obtained using the proposed method in an outdoor environment. An outdoor mobile robot equipped with a LRF(Hokuyo Top-URG) mounted about 800 mm above the ground was allowed to move over a distance of about 250 m in the University of Tokyo Hongo campus. It traveled the course back and forth four times, and LRF scans were obtained every 100 ms. The LRF scans obtained in first three runs were used to train the SVM, and the LRF scans obtained in the last run are used to evaluate accuracy. The total numbers of each class and the total numbers of frames detected each class are listed in Table 7. No groups that consist of more than five pedestrians are appeared in the training data, thus the the maximum number in a group is limited five.

Table 9 Results of classification in an outdoor environment: Per-object accuracies

Target	TP	FP	FN	Precision	Recall
Pedestrian	111	41	11	73.0%	91.0%
Car	4(3)	2	1	77.8%	87.5%

Table 10 Results of classification and tracking in an outdoor environment: Per-frame accuracies

Target	TP	FP	FN	Precision	Recall
Pedestrian	11116	1609	456	87.4%	96.1%
Car	344	44	42	88.6%	89.1%

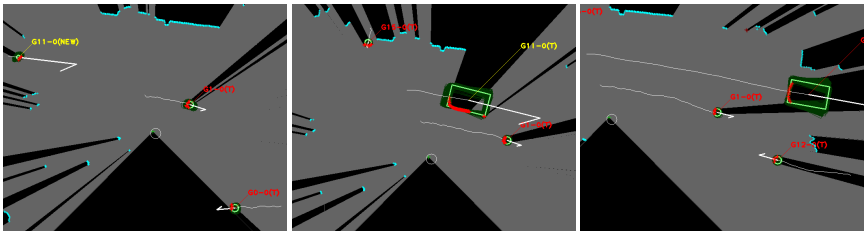

Fig. 6 Result of car tracking. G11 was initially classified as a pedestrian when it was far away from the robot because it initially produced a small LRF segment (left). When it approached the robot, however, the LRF segments became larger and it was correctly classified as a car (center). Tracking was terminated successfully (right).

The parameters defined in Section 3 were set to $t_1 = 1$ sec, $t_2 = 2$ sec, $L_1 = 300$ mm, and $L_2 = 500$ mm. These parameters are suitable for detecting objects moving at a velocity between 500 and 3000 mm/s. The number of particles per object was 500, and average processing time for one frame, including the processing time for the "Extended Trajectory" method described Section 3) was about 42.7 ms (Intel Core i7-940XM). Within the range of the LRF were 8 cars and 113 pedestrians.

The per-object accuracies of the tracking are listed in Table 8. The meaning of the states are the following.

1. **Succeeded**: Tracking completed successfully.
2. **Disrupted in occ.**: Tracking terminated one in an occluded area, and the same object was detected again
3. **Failed** Tracking failed in non-occluded area.
4. **Mismatch** The particle filter migrated another object.

Table 9 and Table 10 shows The per-objects accuracies of the classification and the per-frame accuracies of the tracking are listed in Table 9 and Table 10, where 'TP", "FP," and "FN" mean "True positive." "False positive," and "False Negative.". The number in parentheses in Table 9 is the number of cars first mistakenly classified as a group of pedestrians but later reclassified correctly. Fig. 6 shows an example of such a case. Initially erroneous classification is also the reason that the car Recall value in Table 10 is lower than the pedestrian Recall value there. In these experiments, many false positives were the result of a care being classified as a pedestrian by the "Extended Trajectory" initializing method.

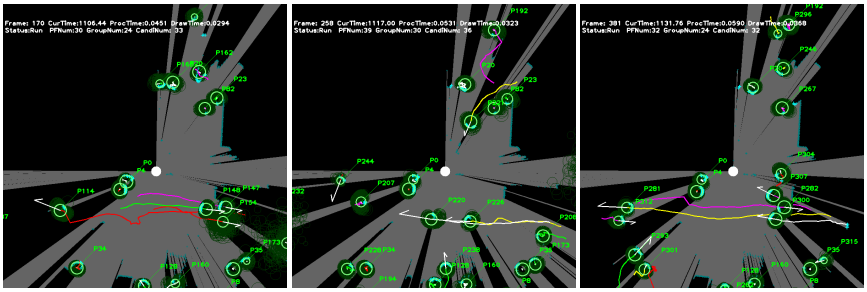

Fig. 7 Scenes of tracking experiment in a crowded environment. White circles indicate the position of the LRF.

Table 11 Tracking results: Per-object accuracies in a crowded environment

Target	Succeeded	Disrupted in occ.	Failed	Mismatch	FP
Pedestrian	625	114	12	37	2

5.2 Classification and Tracking Experiments in a Crowded Indoor Environment

This section describes experimental results obtained using the proposed method in a more crowded environment. One SICK LMS200 was placed on a booth at an exhibition and the real-time performance of the proposed system was demonstrated. The exhibition lasted three days, and we extracted a subset of the whole data for the evaluation (about 10 minutes at the most crowded time). As the LRF was fixed to the ground, the initialize method was the occupancy grid with a polar coordinate.

The SVM data for estimation of numbers of pedestrians was the same as that in the outdoor experiment described in the preceding section. As this experiment was performed in an indoor environment without cars, the class “Car” was removed.

The average numbers of groups and individual pedestrians were 18.3 and 23.8 respectively. The number of particle per object was 500, and average processing time for one frame was about 53.4 ms (Intel Core i7-940XM). Although the number of particles was much greater than that in the experiment described in the preceding section, the initialization method using a grid has a much smaller computation cost than the “Extended Trajectory” method does. As a result, processing time did not become worse. The LRF scans were obtained every 120 ms, so the proposed method could run in real time.

Fig. 7 shows the scenes of the tracking, and the per-object accuracies are listed in Table 11. As the grid map did not update during the evaluation, many stopping persons were detected. Therefore only the numbers of pedestrians that moved more than 1 m are listed in Table 11. As the LRF was fixed, there were far fewer false positives than there were in the outdoor results. The number of pedestrians “Disrupted in occ.” was large, however, because there were several persons were standing near the LRF and they caused large occluded areas.

6 Conclusion

This paper described a method for classifying and tracking multiple moving objects with a laser range finder. The experimental results obtained when we implemented the proposed method using a personal mobility robot demonstrated its effectiveness in the real world. The number of false positives in outdoor environments is a major problem to overcome. We are now implementing initializing test algorithms that use several time-series LRF scans to determine whether or not tracking should start.

A major theoretical shortcoming of this method is that it performs classification and estimation of the number of pedestrians independent of the results of SJPDAFs. In future work, we are planning to integrate classification and tracking using MHT frameworks.

References

1. Y. Bar-Shalom. Extension of the probabilistic data association filter to multi-target tracking. In *Proc. of the 5th Symposium on Nonlinear Estimation*, pp. 16–21, 1974.
2. D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 6, pp. 843–854, 1979.
3. B. Lau, K. O. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, Vol. 2, No. 1, pp. 19–30, 2010.
4. Ata ur Rehman, S. M. Naqvi, L. Mihaylovay, and J. A. Chambers. Clustering and a joint probabilistic data association filter for dealing with occlusions in multi-target tracking. In *Proc. of 16th International Conference on Information Fusion (FUSION 2013)*, 2013.
5. X. Song, J. Cui, H. Zhao, H. Zha, and R. Shibasaki. Laser-based tracking of multiple interacting pedestrians via on-line learning. pp. 92–105, 2013.
6. J. Vermaak and A. Doucet. Maintaining multi-modality through mixture. In *Proc. of 9th IEEE International Conference on Computer Vision(ICCV'03)*, pp. 1110–1116, 2003.
7. R. Kurazume, H. Yamada, K. Murakami, Y. Iwashita, and T. Hasegawa. Target tracking using SIR and MCMC particle filters by multiple cameras and laser range finders. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*, pp. 3838 – 3844, 2008.
8. D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research*, Vol. 22, pp. 99–117, 2003.
9. K. Dietmayer, J. Sparbert, and D. Streller. Model based object classification and object tracking in traffic scenes from range images. In *Proc, of the IEEE Intelligent VehicleSymposium IV*, 2001.
10. H. Zhao, Q. Zhan, M. Chiba, R. Shibasaki, J. Cu, and H. Zha. Moving object classification using horizontal laser scan data. In *Proc. of the IEEE International Conference on Robotics and Automation(ICRA'09)*, pp. 2424–2430, 2009.
11. T. Mori, T. Sato, H. Noguchi, M. Shimosaka, R. Fukui, and T. Sato. Moving objects detection and classification based on trajectories of LRF scan data on a grid map. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pp. 2606–2611, 2010.